

ANSYS LS-DYNA®

Theory Manual

R16@e545952c7 (03/21/25)
LS-DYNA R16

ANSYS

Websites

<https://lsdyna.ansys.com>
<https://support.ansys.com>

Address

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
<http://www.ansys.com>
(T) 724-746-3304
(F) 724-514-9494

Copyright and Trademark Information

© 2025 ANSYS, Inc. Unauthorized use, distribution or duplication is prohibited.

ANSYS, Ansys Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners. FLEXlm and FLEXnet are trademarks of Flexera Software LLC.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001: 2015 companies.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, contact ANSYS, Inc.

Published in the U.S.A.

AES

AES. Copyright © 2001, Dr. Brian Gladman <brg@gladman.uk.net>, Worcester, UK. All rights reserved.

LICENSE TERMS

The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:

1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holder's name is not used to endorse products built using this software without specific written permission.

DISCLAIMER

This software is provided 'as is' with no explicit or implied warranties in respect of any properties, including, but not limited to, correctness and fitness for purpose.

Issue Date: 21/01/2002

This file contains the code for implementing the key schedule for AES (Rijndael) for block and key sizes of 16, 24, and 32 bytes.

Table of Contents

Table of Contents.....	0-1
1 Abstract.....	1-1
2 History of LS-DYNA	2-1
2.1 Features add in 1989-1990.....	2-3
2.2 Options added in 1991-1992	2-4
2.3 Options added in 1993-1994	2-5
2.4 Version 936	2-6
2.5 Version 940	2-8
2.6 Version 950	2-10
2.7 Version 960	2-13
2.8 Version 970	2-18
3 Preliminaries.....	3-1
3.1 Governing Equations	3-1
4 Solid Elements	4-1
4.1 Volume Integration	4-3
4.2 Solid Element 2	4-5
4.2.1 Brief summary of solid element 2	4-5
4.2.2 Transverse shear locking example	4-6
4.2.3 Solid element -2.....	4-7
4.2.4 Transverse shear locking example revisited.....	4-8
4.2.5 Solid element -1	4-8
4.2.6 Example	4-9
4.3 Hourglass Control	4-9
4.4 Puso Hourglass Control.....	4-15
4.5 Fully Integrated Brick Elements and Mid-Step Strain Evaluation.....	4-20
4.6 Four Node Tetrahedron Element	4-22
4.7 Nodal Pressure Tetrahedron.....	4-23
4.8 Six Node Pentahedron Element.....	4-24
4.9 Fully Integrated Brick Element With 48 Degrees-of-Freedom.....	4-25

4.10 Fully Integrated Tetrahedron Element With 24 Degrees-of-Freedom	4-28
4.11 The Cosserat Point Elements in LS-DYNA	4-31
4.11.1 Introduction	4-31
4.11.2 Geometry	4-32
4.11.3 Deformation and strain	4-32
4.11.4 Stress and Force	4-33
4.11.5 CPE3D10 modification	4-34
4.11.6 Hourglass	4-35
4.11.7 Comparison to Jabareen & Rubin	4-36
4.11.8 Nodal formulation	4-36
4.11.9 Cantilever Beam Example	4-37
4.11.10 Compression Test for Rubber Example	4-38
4.11.11 References	4-38
4.12 Integral Difference Scheme as Basis For 2D Solids	4-39
4.12.1 Rezoning With 2D Solid Elements	4-44
4.13 Solid Elements 18 and -18 – Incompatible Modes	4-48
4.13.1 Introduction	4-48
4.13.2 Theory	4-50
5 Connectors	5-1
5.1 Self Piercing Rivet (SPR)	5-1
6 Cohesive Elements	6-1
6.1 Kinematics	6-2
6.2 Constitutive law	6-5
6.3 Nodal forces	6-5
6.4 Drilling constraint in shell ± 29	6-7
6.5 Rotational masses in shell ± 29	6-8
6.6 Stiffness matrix in shell ± 29	6-9
7 Belytschko Beam	7-1
7.1 Co-rotational Technique	7-1
7.2 Belytschko Beam Element Formulation	7-5
7.2.1 Calculation of Deformations	7-5
7.2.2 Calculation of Internal Forces	7-6
7.2.3 Updating the Body Coordinate Unit Vectors	7-8
8 Hughes-Liu Beam	8-1
8.1 Geometry	8-1

8.2 Fiber Coordinate System	8-5
8.2.1 Local Coordinate System	8-6
8.3 Strains and Stress Update.....	8-7
8.3.1 Incremental Strain and Spin Tensors	8-7
8.3.2 Stress Update	8-7
8.3.3 Incremental Strain-Displacement Relations	8-8
8.3.4 Spatial Integration	8-9
9 Warped Beam Elements	9-1
9.1 Resultant Warped Beam.....	9-1
9.1.1 Green-Lagrange Strains in Terms of Deformational Displacements.....	9-1
9.1.2 Deformational Displacements After Large Rotations	9-1
9.1.3 Green-Lagrange Strains in terms of Centroidal Displacements and Angular Rotations	9-2
9.1.4 Strain Energy.....	9-3
10 Belytschko-Lin-Tsay Shell	10-1
10.1 Co-rotational Coordinates	10-1
10.2 Velocity-Strain Displacement Relations.....	10-3
10.3 Stress Resultants and Nodal Forces	10-5
10.4 Hourglass Control (Belytschko-Lin-Tsay)	10-6
10.5 Hourglass Control (Englemann and Whirley)	10-7
10.6 Belytschko-Wong-Chiang Improvements	10-9
11 Triangular Shells.....	11-1
11.1 C^0 Triangular Shell	11-1
11.1.1 Co-rotational Coordinates	11-1
11.1.2 Velocity-Strain Relations.....	11-2
11.1.3 Stress Resultants and Nodal Forces	11-6
11.2 Marchertas-Belytschko Triangular Shell.....	11-7
11.2.1 Element Coordinates	11-7
11.2.2 Displacement Interpolation	11-10
11.2.3 Strain-Displacement Relations	11-11
11.2.4 Nodal Force Calculations	11-12
12 Fully Integrated Shell (Type 16).....	12-1
12.1 Introduction.....	12-1
12.2 Hu-Washizu Three Field Principle.....	12-1
12.3 In-plane Assumed Strain Field.....	12-2

12.4 Transverse Shear Assumed Strain Field.....	12-3
12.5 Rigid Body Motion	12-4
12.6 Belytschko-Leviathan Projection	12-5
13 Shells with Thickness Stretch	13-1
13.1 Introduction.....	13-1
13.2 Shell Type 25	13-2
13.2.1 Formulation	13-2
13.2.2 Hourglass modes.....	13-3
13.3 Shell Type 26	13-4
13.3.1 Modifying shell type 16.....	13-4
13.4 Shell Type 27	13-5
13.4.1 Modifying shell type 4.....	13-5
13.5 Related Features.....	13-5
13.5.1 Continuous vs decoupled thickness field.....	13-5
13.5.2 Nodal masses	13-6
13.5.3 Transfer of contact forces.....	13-6
13.6 Contact Pressure Treatment in Shells 2, 4 and 16.....	13-7
14 Hughes-Liu Shell.....	14-1
14.1 Geometry	14-2
14.2 Kinematics	14-4
14.2.1 Fiber Coordinate System.....	14-5
14.2.2 Lamina Coordinate System	14-7
14.3 Strains and Stress Update.....	14-8
14.3.1 Incremental Strain and Spin Tensors	14-8
14.3.2 Stress Update	14-9
14.3.3 Incremental Strain-Displacement Relations	14-9
14.4 Element Mass Matrix	14-10
14.5 Accounting for Thickness Changes.....	14-11
14.6 Fully Integrated Hughes-Liu Shells.....	14-12
15 Transverse Shear Treatment For Layered Shell	15-1
16 Eight-Node Solid Shell Element.....	16-1
17 Eight-Node Solid Element for Thick Shell Simulations.....	17-1
17.1 Abstract.....	17-1
17.2 Introduction.....	17-1
17.3 Element Formulations	17-2

17.3.1 Strain Operator	17-2
17.3.2 Co-rotational Coordinate System	17-7
17.3.3 Stress and Strain Measures	17-8
17.3.4 Co-rotational Stress and Strain Updates	17-9
Tangent Stiffness Matrix and Nodal Force Vectors	17-12
17.4 Numerical Examples	17-15
17.4.1 Timoshenko Cantilever Beam	17-15
17.4.2 Pinched Cylinder	17-18
17.4.3 Scordelis-Lo Roof	17-18
17.4.4 Circular Sheet Stretched with a Tight Die	17-20
17.5 Conclusions	17-23
18 Truss Element	18-1
19 Membrane Element	19-1
19.1 Co-rotational Coordinates	19-1
19.2 Velocity-Strain Displacement Relations	19-2
19.3 Stress Resultants and Nodal Forces	19-2
19.4 Membrane Hourglass Control	19-3
20 Discrete Elements and Masses	20-1
20.1 Orientation Vectors	20-2
20.2 Dynamic Magnification “Strain Rate” Effects	20-4
20.3 Deflection Limits in Tension and Compression	20-4
20.4 Linear Elastic or Linear Viscous	20-5
20.5 Nonlinear Elastic or Nonlinear Viscous	20-5
20.6 Elasto-Plastic with Isotropic Hardening	20-6
20.7 General Nonlinear	20-6
20.8 Linear Visco-Elastic	20-7
20.9 Muscle Model	20-9
20.10 Seat Belt Material	20-12
20.11 Seat Belt Elements	20-13
20.12 Sliprings	20-14
20.13 Retractors	20-15
20.14 Sensors	20-18
20.15 Pretensioners	20-19
20.16 Accelerometers	20-19
21 Simplified Arbitrary Lagrangian-Eulerian	21-1

21.1 Mesh Smoothing Algorithms	21-3
21.1.1 Equipotential Smoothing of Interior Nodes	21-3
21.1.2 Simple Averaging	21-11
21.1.3 Kikuchi's Algorithm	21-11
21.1.4 Surface Smoothing	21-11
21.1.5 Combining Smoothing Algorithms.....	21-11
21.2 Advection Algorithms.....	21-12
21.2.1 Advection Methods in One Dimension	21-12
21.2.2 Advection Methods in Three Dimensions.....	21-16
21.3 The Manual Rezone	21-24
22 Stress Update Overview	22-1
22.1 Jaumann Stress Rate.....	22-1
22.2 Jaumann Stress Rate Used With Equations of State	22-2
22.3 Green-Naghdi Stress Rate.....	22-3
22.4 Elastoplastic Materials	22-5
22.5 Hyperelastic Materials	22-8
22.6 Layered Composites.....	22-10
22.7 Constraints on Orthotropic Elastic Constants	22-12
22.8 Local Material Coordinate Systems in Solid Elements.....	22-13
22.9 General Erosion Criteria for Solid Elements	22-14
22.10 Strain Output to the LS-DYNA Database	22-15
22.12 Strain Rate Effects in Material Models	22-16
22.12.1 Running N-average option.....	22-16
22.12.2 Last N-average option	22-17
22.12.3 Averaging over a fixed amount of time T	22-17
22.13 Algorithmic Consistent Tangent Modulus for Plasticity	22-18
23 Material Models	23-23
23.1 Material Model 1: Elastic.....	23-37
23.2 Material Model 2: Orthotropic Elastic	23-38
23.3 Material Model 3: Elastic Plastic with Kinematic Hardening	23-40
23.4 Material Model 4: Thermo-Elastic-Plastic.....	23-44
23.5 Material Model 5: Soil and Crushable Foam	23-46
23.6 Material Model 6: Viscoelastic	23-48
23.7 Material Model 7: Continuum Rubber	23-49
23.8 Material Model 8: Explosive Burn	23-50
23.9 Material Model 9: Null Material	23-53

23.10 Material Model 10: Elastic-Plastic-Hydrodynamic	23-54
23.11 Material Model 11: Elastic-Plastic With Thermal Softening	23-57
23.12 Material Model 12: Isotropic Elastic-Plastic	23-59
23.13 Material Model 13: Isotropic Elastic-Plastic with Failure.....	23-60
23.14 Material Model 14: Soil and Crushable Foam With Failure	23-61
23.15 Material Model 15: Johnson and Cook Plasticity Model.....	23-62
23.16 Material Model 16: Pseudo Tensor.....	23-64
23.17 Material Model 17: Isotropic Elastic-Plastic With Oriented Cracks	23-68
23.18 Material Model 18: Power Law Isotropic Plasticity	23-70
23.19 Material Model 19: Strain Rate Dependent Isotropic Plasticity.....	23-71
23.20 Material Model 20: Rigid	23-73
23.21 Material Model 21: Thermal Orthotropic Elastic.....	23-74
23.22 Material Model 22: Chang-Chang Composite Failure Model.....	23-76
23.23 Material Model 23: Thermal Orthotropic Elastic with 12 Curves.....	23-78
23.24 Material Model 24: Piecewise Linear Isotropic Plasticity	23-80
23.25 Material Model 25: Kinematic Hardening Cap Model	23-82
23.26 Material Model 26: Crushable Foam	23-87
23.27 Material Model 27: Incompressible Mooney-Rivlin Rubber	23-91
23.27.1 Stress Update for Shell Elements.....	23-93
23.27.2 Derivation of the Continuum Tangent Stiffness.....	23-94
23.27.3 The Algorithmic Tangent Stiffness for Shell Elements	23-96
23.28 Material Model 28: Resultant Plasticity	23-98
23.29 Material Model 29: FORCE LIMITED Resultant Formulation.....	23-100
23.29.1 Internal Forces	23-102
23.29.2 Tangent Stiffness	23-105
23.30 Material Model 30: Closed-Form Update Shell Plasticity	23-107
23.30.1 Mathematical Description of the Material Model.....	23-107
23.30.2 Algorithmic Stress Update.....	23-108
23.30.3 Tangent Stiffness Matrix	23-109
23.31 Material Model 31: Slightly Compressible Rubber Model.....	23-112
23.32 Material Model 32: Laminated Glass Model.....	23-113
23.33 Material Model 33: Barlat's Anisotropic Plasticity Model.....	23-114
23.34 Material Model 34: Fabric	23-115
23.35 Material Model 35: Kinematic/Isotropic Plastic Green-Naghdi Rate	23-117
23.36 Material Model 36: Barlat's 3-Parameter Plasticity Model.....	23-118
23.36.1 Material Tangent Stiffness	23-119
23.36.2 Load curves in different directions	23-121

23.36.2.1 An introductory remark.....	23-121
23.36.2.2 The model	23-122
23.36.3 Variable Lankford parameters	23-123
23.37 Material Model 37: Transversely Anisotropic Elastic-Plastic	23-125
23.38 Material Model 38: Blatz-Ko Compressible Foam.....	23-127
23.39 Material Model 39: Transversely Anisotropic Elastic-Plastic With FLD.....	23-128
23.42 Material Model 42: Planar Anisotropic Plasticity Model.....	23-129
23.51 Material Model 51: Temperature and Rate Dependent Plasticity	23-130
23.52 Material Model 52: Sandia's Damage Model	23-132
23.53 Material Model 53: Low Density Closed Cell Polyurethane Foam	23-133
23.54 Material Models 54 and 55: Enhanced Composite Damage Model.....	23-135
23.57 Material Model 57: Low Density Urethane Foam	23-138
23.58 Material Model 58: Laminated Composite Fabric	23-141
23.60 Material Model 60: Elastic With Viscosity	23-143
23.61 Material Model 61: Maxwell/Kelvin Viscoelastic with Maximum Strain	23-144
23.62 Material Model 62: Viscous Foam	23-145
23.63 Material Model 63: Crushable Foam	23-146
23.64 Material Model 64: Strain Rate Sensitive Power-Law Plasticity.....	23-148
23.65 Material Model 65: Modified Zerilli/Armstrong	23-149
23.66 Material Model 66: Linear Stiffness/Linear Viscous 3D Discrete Beam	23-150
23.67 Material Model 67: Nonlinear Stiffness/Viscous 3D Discrete Beam	23-151
23.68 Material Model 68: Nonlinear Plastic/Linear Viscous 3D Discrete Beam.....	23-152
23.69 Material Model 69: Side Impact Dummy Damper (SID Damper).....	23-153
23.70 Material Model 70: Hydraulic/Gas Damper	23-155
23.71 Material Model 71: Cable.....	23-156
23.73 Material Model 73: Low Density Viscoelastic Foam.....	23-157
23.74 Material Model 74: Elastic Spring for the Discrete Beam.....	23-158
23.75 Material Model 75: Bilkhu/Dubois Foam Model	23-159
23.76 Material Model 76: General Viscoelastic	23-160
23.76.1 Introduction.....	23-160
23.76.2 Constitutive Model.....	23-160
23.76.3 Tangent Modulus	23-160
23.76.4 Using the Relaxation Curve	23-162
23.77 Material Model 77: Hyperviscoelastic Rubber.....	23-163
23.77.1 General Hyperelastic Rubber Model	23-164
23.77.2 Ogden Rubber Model	23-165
23.77.3 The Viscoelastic Contribution.....	23-166

23.77.4 Stress Update for Shell Elements.....	23-166
23.78 Material Model 78: Soil/Concrete	23-168
23.79 Material Model 79: Hysteretic Soil.....	23-170
23.80 Material Model 80: Ramberg-Osgood Plasticity	23-171
23.81 Material Models 81 and 82: Plasticity with Damage and Orthotropic Option.....	23-172
23.81.1 Material Model 82: Isotropic Elastic-Plastic with Anisotropic Damage	23-174
23.83 Material Model 83: Fu-Chang's Foam With Rate Effects	23-177
23.84 Material Model 84 and 85: Winfrith Concrete.....	23-180
23.87 Material Model 87: Cellular Rubber	23-181
23.88 Material Model 88: MTS Model	23-183
23.89 Material Model 89: Plasticity Polymer	23-185
23.90 Material Model 90: Acoustic	23-186
23.91 Material Model 91: Soft Tissue	23-187
23.94 Material Model 94: Inelastic Spring Discrete Beam	23-188
23.96 Material Model 96: Brittle Damage Model	23-189
23.97 Material Model 97: General Joint Discrete Beam.....	23-191
23.98 Material Model 98: Simplified Johnson Cook.....	23-192
23.100 Material Model 100: Spot Weld	23-193
23.101 Material Model 101: GE Thermoplastics	23-196
23.102 Material Model 102: Hyperbolic Sine.....	23-197
23.103 Material Model 103: Anisotropic Viscoplastic	23-198
23.104 Material Model 104: Continuum Damage Mechanics Model	23-199
23.106 Material Model 106: Elastic Viscoplastic Thermal	23-202
23.110 Material Model 110: Johnson-Holmquist Ceramic Model.....	23-203
23.111 Material Model 111: Johnson-Holmquist Concrete Model	23-205
23.115 Material Model 115: Elastic Creep Model	23-206
23.116 Material Model 116: Composite Layup	23-207
23.117 Material Model 117-118: Composite Matrix.....	23-208
23.119 Material Model 119: General Nonlinear 6DOF Discrete Beam.....	23-209
23.120 Material Model 120: Gurson	23-210
23.120 Material Model 120: Gurson RCDC	23-211
23.124 Material Model 124: Tension-Compression Plasticity	23-212
23.126 Material Model 126: Metallic Honeycomb	23-213
23.126.1 Stress Update	23-215
23.126.2 Support for Independent Shear and Hydrostatic Yield Stress Limit.....	23-216
23.127 Material Model 127: Arruda-Boyce Hyperviscoelastic Rubber	23-219
23.128 Material Model 128: Heart Tissue.....	23-221

23.129 Material Model 129: Isotropic Lung Tissue	23-222
23.130 Material Model 130: Special Orthotropic.....	23-223
23.131 Material Model 131: Isotropic Smeared Crack.....	23-224
23.133 Material Model 133: Barlat_YLD2000.....	23-226
23.133.1 Closest point projection algorithm	23-229
23.134 Material Model 134: Viscoelastic Fabric.....	23-231
23.139 Material Model 139: Modified Force Limited.....	23-232
23.141 Material Model 141: Rate Sensitive Polymer.....	23-235
23.142 Material Model 142: Transversely Isotropic Crushable Foam.....	23-236
23.143 Material Model 143: Wood Model	23-238
23.144 Material Model 144: Pitzer Crushable Foam	23-245
23.147 Material Model 147: FHWA Soil Model.....	23-246
23.154 Material Model 154: Deshpande-Fleck Foam	23-253
23.156 Material Model 156: Muscle	23-254
23.158 Material Model 158: Rate Sensitive Composite Fabric	23-255
23.159 Material Model 159: Continuous Surface Cap Model	23-256
23.161 Material Models 161 and 162: Composite MSC	23-262
23.163 Material Model 163: Modified Crushable Foam	23-268
23.164 Material Model 164: Brain Linear Viscoelastic.....	23-269
23.166 Material Model 166: Moment Curvature Beam	23-270
23.169 Material Model 169: Arup Adhesive.....	23-272
23.170 Material Model 170: Resultant Anisotropic	23-273
23.175 Material Model 175: Viscoelastic Maxwell.....	23-274
23.176 Material Model 176: Quasilinear Viscoelastic.....	23-275
23.177 Material Models 177 and 178: Hill Foam and Viscoelastic Hill Foam	23-276
23.177.1 Hyperelasticity Using the Principal Stretch Ratios.....	23-276
23.177.2 Hill's Strain Energy Function.....	23-277
23.177.3 Viscous Stress	23-279
23.177.4 Viscoelastic Stress Contribution.....	23-279
23.177.5 Material Tangent Modulus for the Fully Integrated Brick	23-280
23.177.6 Viscous damping	23-281
23.179 Material Models 179 and 180: Low Density Synthetic Foam.....	23-282
23.179.1 Hyperelasticity Using the Principal Stretch Ratios.....	23-282
23.179.2 Strain Energy Function	23-283
23.179.3 Modeling of the Hysteresis.....	23-284
23.179.4 Viscous Stress	23-284
23.179.5 Viscoelastic Stress Contribution.....	23-285

23.179.6 Stress Corresponding to First Load Cycle	23-285
23.181 Material Model 181: Simplified Rubber/Foam.....	23-287
23.181.1 Hyperelasticity Using the Principal Stretch Ratios.....	23-287
23.181.1.1 Determination of f, rubber option	23-288
23.181.1.2 Determination of f, foam option.....	23-289
23.181.2 Some Remarks	23-289
23.181.2.1 Strain rates.....	23-289
23.181.2.2 Modeling of the Frequency Independent Damping.....	23-289
23.187 Material Model 187: Semi-Analytical Model for the Simulation of Polymers	23-290
23.187.1 Material law formulation.....	23-290
23.187.2 Hardening formulation	23-294
23.187.3 Rate effects.....	23-295
23.187.4 Damage and failure.....	23-295
23.196 Material Model 196: General Spring Discrete Beam	23-297
24 Equation of State Models	24-1
24.1 Equation of State Form 1: Linear Polynomial	24-1
24.2 Equation of State Form 2: JWL High Explosive.....	24-2
24.3 Equation of State Form 3: Sack “Tuesday” High Explosives	24-2
24.4 Equation of State Form 4: Gruneisen	24-3
24.5 Equation of State Form 5: Ratio of Polynomials.....	24-3
24.6 Equation of State Form 6: Linear With Energy Deposition	24-4
24.7 Equation of State Form 7: Ignition and Growth Model.....	24-4
24.8 Equation of State Form 8: Tabulated Compaction.....	24-5
24.9 Equation of State Form 9: Tabulated.....	24-5
24.10 Equation of State Form 10: Propellant-Deflagration	24-6
25 Artificial Bulk Viscosity	25-1
25.1 Shock Waves	25-1
25.2 Bulk Viscosity.....	25-3
26 Time Step Control	26-1
26.1 Time Step Calculations for Solid Elements	26-1
26.2 Time Step Calculations for Beam and Truss Elements	26-2
26.3 Time Step Calculations for Shell Elements.....	26-3
26.4 Time Step Calculations for Solid Shell Elements	26-4
26.5 Time Step Calculations for Discrete Elements	26-4
26.6 Time Step Calculations for Cohesive Elements.....	26-6

27 Boundary and Loading Conditions	27-1
27.1 Pressure Boundary Conditions	27-1
27.2 Transmitting Boundaries	27-3
27.3 Kinematic Boundary Conditions	27-3
27.4 Displacement Constraints	27-3
27.5 Prescribed Displacements, Velocities, and Accelerations	27-4
27.6 Body Force Loads	27-4
28 Time Integration	28-1
28.1 Background	28-1
28.2 The Central Difference Method	28-2
28.3 Stability of Central Difference Method	28-3
28.4 Subcycling (Mixed Time Integration)	28-6
29 Rigid Body Dynamics	29-1
29.1.1 Equations of Motion for the Body	29-1
29.1.2 Rigid Body Node Constraints	29-1
29.1.3 Equations of Motion for the Nodes	29-2
29.1.4 The Generalized Equations of Motion	29-3
29.1.5 Constraints	29-4
29.1.6 Algorithmic Update	29-5
29.2 Rigid Body Joints	29-6
29.2.1 Penalty types	29-6
29.2.2 Lagrange multiplier types	29-9
29.3 Deformable to Rigid Material Switching	29-12
29.4 Rigid Body Welds	29-13
30 Contact-Impact Algorithm	30-1
30.1 Introduction	30-1
30.2 Kinematic Constraint Method	30-1
30.3 Penalty Method	30-2
30.4 Distributed Parameter Method	30-3
30.5 Preliminaries	30-4
30.6 Sliding with Closure and Separation	30-5
30.6.1 Standard Penalty Formulation	30-5
30.6.2 Soft Constraint Penalty Formulation	30-6
30.6.3 Segment-to-Segment (SOFT = 2) Penalty Formulation	30-7
30.7 Accounting for Shell Thickness	30-8

30.8 Initial Interpenetrations	30-9
30.9 Finding Penetration by Tracked Nodes and Segments.....	30-11
30.9.1 Overview of Bucket Sorting.....	30-11
30.9.2 Bucket Sorting in Non-Automatic Single Surface Contact (*CONTACT_SIN- GLE_SURFACE)	30-13
30.9.3 Bucket Sorting in Surface-to-Surface and Automatic Single Surface Contact (*CONTACT_AUTOMATIC_SINGLE_SURFACE)	30-15
30.9.4 Incremental Tracked Node Search.....	30-16
30.10 Contact Energy Calculation	30-20
30.11 Contact Damping.....	30-21
30.12 Friction	30-23
30.13 Tied Interfaces	30-24
30.13.1 Outline of Available Tied Contacts	30-27
30.13.2 Implementation of Constraint-Based Tied Contacts.....	30-29
30.13.3 Tiebreak Contacts.....	30-30
30.13.4 Alternatives to Tied Contacts	30-32
30.14 Strongly Objective Tied Contacts.....	30-32
30.14.1 Kinematics	30-33
30.14.2 Translational Constraint.....	30-35
30.14.3 Bending and Torsion Constraint.....	30-35
30.15 Sliding-Only Interfaces.....	30-36
30.16 Single Surface Contact Algorithms in LS-DYNA	30-37
30.17 Surface-to-Surface Constraint Algorithm	30-40
30.18 Rigid Boundaries.....	30-43
30.18.1 Planar Rigid Boundaries	30-43
30.18.2 Geometric Rigid Boundaries.....	30-45
30.19 VDA/IGES Contact.....	30-45
30.20 Simulated Draw Beads	30-49
30.21 Edge-to-Edge Contact.....	30-52
30.22 Beam-to-Beam Contact	30-53
30.23 Mortar Contact.....	30-55
30.23.1 Kinematics	30-55
30.23.2 Constitutive Relation.....	30-56
30.23.3 Contact Nodal Forces.....	30-57
30.23.4 Treatment of beams, sharp solid edges, and shell edges	30-57
30.23.5 Characteristic Length and Contact Release.....	30-60
30.23.6 Output for Debugging Implicit Models	30-61

30.23.7 Initial Penetrations	30-61
30.23.8 2D Mortar contact.....	30-64
30.23.9 Tied and tiebreak option	30-67
31 Geometric Contact Entities.....	31-1
32 Nodal Constraints	32-1
32.1 Nodal Constraint Sets.....	32-1
32.2 Linear Constraint Equations	32-1
33 Vectorization and Parallelization	33-1
33.1 Vectorization	33-1
33.2 Parallelization	33-4
34 Airbags	34-1
34.1 Control Volume Modeling	34-1
34.2 Equation of State Model	34-3
34.3 Airbag Inflation Model.....	34-5
34.4 Wang's Hybrid Inflation Model	34-7
34.5 Constant Volume Tank Test	34-10
35 Dynamic Relaxation and System Damping	35-1
35.1 Dynamic Relaxation For Initialization.....	35-1
35.2 Mass Weighted Damping.....	35-4
35.3 Dynamic Relaxation—How Fast Does it Converge?	35-5
36 Heat Transfer	36-1
36.1 Conduction of Heat in an Orthotropic Solid	36-1
36.2 Thermal Boundary Conditions	36-2
36.3 Thermal Energy Balances	36-4
36.4 Heat Generation.....	36-4
36.5 Initial Conditions	36-4
36.6 Material Properties.....	36-4
36.7 Nonlinear Analysis.....	36-5
36.8 Units	36-5
37 Adaptivity for Shell Elements	37-1
38 Implicit.....	38-1
38.1 Introduction.....	38-1
38.2 Equations	38-2
38.2.1 Discretization	38-2

38.2.2 Constraints.....	38-3
38.2.3 Reaction forces due to constraints.....	38-5
38.2.4 Some common constraints.....	38-6
38.2.5 Inequality constraints.....	38-7
38.3 Implicit Statics.....	38-9
38.3.1 Linearization.....	38-9
38.3.2 Linear theory	38-11
38.3.3 Quasi-Newton iterations	38-13
38.3.4 Tangent stiffness reformations	38-17
38.3.5 Line search.....	38-18
38.3.6 Convergence check.....	38-19
38.3.7 Automatic time stepping.....	38-21
38.4 Implicit Dynamics.....	38-22
38.4.1 Newmark time integration.....	38-22
38.4.2 Energy balance	38-23
38.4.3 Practical considerations	38-24
38.4.4 Linear theory	38-25
39 Arc-length	39-1
39.1 Overview	39-1
39.2 Nonlinear equations	39-4
39.3 Newton iterations.....	39-5
39.4 Arc-length constraint – predictor step.....	39-7
39.5 Arc-length constraint – corrector steps.....	39-8
40 Sparse Direct Linear Equation Solvers	40-1
40.1 Sparsity Preserving Orderings.....	40-1
40.2 Multifrontal Algorithm.....	40-2
40.3 The Five Solver Options.....	40-3
40.4 Treating Matrix Singularities	40-4
41 Sparse Eigensolver	41-1
41.1 The Eigenvalue Problem for Rotating Systems	41-2
42 Boundary Element Method	42-1
42.1 Governing Equations	42-1
42.2 Surface Representation	42-2
42.3 The Neighbor Array	42-3
42.4 Wakes	42-5

42.5 Execution Time Control	42-6
42.6 Free-Stream Flow	42-7
43 SPH	43-1
43.1 SPH Formulation.....	43-1
43.1.1 Definitions	43-1
43.1.2 Discrete Form of Conservation Equations.....	43-3
43.1.3 Applications to Conservation Equations	43-4
43.1.4 Formulation Available in LS-DYNA	43-4
43.2 Sorting.....	43-5
43.3 Artificial Viscosity	43-5
43.4 Time Integration	43-6
43.5 Initial Setup	43-6
44 Element-Free Galerkin	44-1
44.1 Moving least-squares	44-1
44.2 Integration constraint and strain smoothing.....	44-3
44.3 Lagrangian strain smoothing for path-dependent problems.....	44-4
44.4 Galerkin approximation for explicit dynamic computation.....	44-5
44.5 Imposition of essential boundary condition.....	44-6
44.6 Mesh-free Shell	44-8
44.6.1 Mesh-free Shell Surface Representation.....	44-8
44.6.2 Updated Lagrangian Formulation and Co-rotational Procedure.....	44-10
45 Linear shells	45-1
45.1 Shells for Linear Analysis.....	45-1
45.2 Wilson's Shell (element #20).....	45-1
45.2.1 Plate Element.....	45-1
45.2.2 Membrane Element.....	45-4
45.3 Assumed Strain/Membrane with Drilling Degree-of-freedom (element #18)	45-7
45.3.1 Membrane Element.....	45-7
45.3.2 Plate Element.....	45-7
45.4 Differences between Element Types 18 and 20.....	45-8
46 Random Geometrical Imperfections.....	46-1
46.1 Introduction to Random Geometrical Imperfections Using Karhunen-Loève Expansions	46-1
46.2 Methodology	46-1
46.2.1 Generation of random fields using Karhunen-Loève expansion	46-1

46.2.2 Solution of Fredholm integral of the second kind for analytical covariance functions	46-2
46.2.3 Generating eigenfunctions from experimentally measured fields	46-4
47 Frequency Domain.....	47-1
47.1 Frequency Response Functions	47-1
47.1.1 FRF Computations.....	47-1
47.1.2 About the damping	47-3
47.2 ACOUSTIC FEM.....	47-4
47.2.1 Theory basis.....	47-4
48 Rotor Dynamics	48-1
48.1 Introduction.....	48-1
48.2 Two Coordinate systems	48-1
48.3 Forces in the Rotating Coordinate System.....	48-2
48.4 Transformation between Coordinate Systems	48-3
48.5 Equation of Motion in Rotating Coordinate System	48-4
48.6 Equation of Motion in Fixed Coordinate System.....	48-6
48.7 Arbitrary Rotation Axis	48-6
49 References	49-1

1

Abstract

LS-DYNA is a general purpose finite element code for analyzing the large deformation static and dynamic response of structures including structures coupled to fluids. The main solution methodology is based on explicit time integration. An implicit solver is currently available with somewhat limited capabilities including structural analysis and heat transfer. A contact-impact algorithm allows difficult contact problems to be easily treated with heat transfer included across the contact interfaces. By a specialization of this algorithm, such interfaces can be rigidly tied to admit variable zoning without the need of mesh transition regions. Other specializations, allow draw beads in metal stamping applications to be easily modeled simply by defining a line of nodes along the draw bead. Spatial discretization is achieved by the use of four node tetrahedron and eight node solid elements, two node beam elements, three and four node shell elements, eight node solid shell elements, truss elements, membrane elements, discrete elements, and rigid bodies. A variety of element formulations are available for each element type. Specialized capabilities for airbags, sensors, and seatbelts have tailored LS-DYNA for applications in the automotive industry. Adaptive remeshing is available for shell elements and is widely used in sheet metal stamping applications. LS-DYNA currently contains approximately one-hundred constitutive models and ten equations-of-state to cover a wide range of material behavior. This theoretical manual has been written to provide users and potential users with insight into the mathematical and physical basis of the code.

2

History of LS-DYNA

The origin of LS-DYNA dates back to the public domain software, DYNA3D, which was developed in the mid-seventies at the Lawrence Livermore National Laboratory. The first version of DYNA3D [Hallquist 1976a] was released in 1976 with constant stress 4- or 8-node solid elements, 16- and 20-node solid elements with $2 \times 2 \times 2$ Gaussian quadrature, 3, 4, and 8-node membrane elements, and a 2-node cable element. A nodal constraint contact-impact interface algorithm [Hallquist 1977] was available. On the Control Data CDC-7600, a supercomputer in 1976, the speed of the code varied from 36 minutes per 106 mesh cycles with 4-8 node solids to 180 minutes per 106 mesh cycles with 16 and 20 node solids. Without hourglass control to prevent formation of non-physical zero energy deformation modes, constant stress solids were processed at 12 minutes per 106 mesh cycles. A moderate number of very costly solutions were obtained with this version of DYNA3D using 16- and 20-node solids. Hourglass modes combined with the procedure for computing the time step size prevented us from obtaining solutions with constant stress elements.

In this early development, several things became apparent. Hourglass deformation modes of the constant stress elements were invariably excited by the contact-impact algorithm, showing that a new sliding interface algorithm was needed. Higher order elements seemed to be impractical for shock wave propagation because of numerical noise resulting from the *ad hoc* mass lumping necessary to generate a diagonal mass matrix. Although the lower frequency structural response was accurately computed with these elements, their high computer cost made analysis so expensive as to be impractical. It was obvious that realistic three-dimensional structural calculations were possible, if and only if the under-integrated eight node constant stress solid element could be made to function. This implied a need for a much better sliding interface algorithm, a more cost-effective hourglass control, more optimal programming, and a machine much faster than the CDC-7600. This latter need was fulfilled several years later when LLNL took delivery of its first CRAY-1. At this time, DYNA3D was completely rewritten.

The next version, released in 1979, achieved the aforementioned goals. On the CRAY the vectorized speed was 50 times faster, 0.67 minutes per million mesh cycles. A

symmetric, penalty-based, contact-impact algorithm was considerably faster in execution speed and exceedingly reliable. Due to lack of use, the membrane and cable elements were stripped and all higher order elements were eliminated as well. Wilkins' finite difference equations [Wilkins et al. 1974] were implemented in unvectorized form in an overlay to compare their performance with the finite element method. The finite difference algorithm proved to be nearly two times more expensive than the finite element approach (apart from vectorization) with no compensating increase in accuracy, and was removed in the next code update.

The 1981 version [Hallquist 1981a] evolved from the 1979 version. Nine additional material models were added to allow a much broader range of problems to be modeled including explosive-structure and soil-structure interactions. Body force loads were implemented for angular velocities and base accelerations. A link was also established from the 3D Eulerian code JOY [Couch, et. al., 1983] for studying the structural response to impacts by penetrating projectiles. An option was provided for storing element data on disk thereby doubling the capacity of DYNA3D.

The 1982 version of DYNA3D [Hallquist 1982] accepted DYNA2D [Hallquist 1980] material input directly. The new organization was such that equations of state and constitutive models of any complexity could be easily added. Complete vectorization of the material models had been nearly achieved with about a 10 percent increase in execution speed over the 1981 version.

In the 1986 version of DYNA3D [Hallquist and Benson 1986], many new features were added, including beams, shells, rigid bodies, single surface contact, interface friction, discrete springs and dampers, optional hourglass treatments, optional exact volume integration, and VAX/VMS, IBM, UNIX, COS operating systems compatibility, that greatly expanded its range of applications. DYNA3D thus became the first code to have a general single surface contact algorithm.

In the 1987 version of DYNA3D [Hallquist and Benson 1987] metal forming simulations and composite analysis became a reality. This version included shell thickness changes, the Belytschko-Tsay shell element [Belytschko and Tsay, 1981], and dynamic relaxation. Also included were non-reflecting boundaries, user specified integration rules for shell and beam elements, a layered composite damage model, and single point constraints.

New capabilities added in the 1988 DYNA3D [Hallquist 1988] version included a cost effective resultant beam element, a truss element, a C0 triangular shell, the BCIZ triangular shell [Bazeley et al., 1965], mixing of element formulations in calculations, composite failure modeling for solids, noniterative plane stress plasticity, contact surfaces with spot welds, tiebreak sliding surfaces, beam surface contact, finite stonewalls, stone-wall reaction forces, energy calculations for all elements, a crushable foam constitutive model, comment cards in the input, and one-dimensional slidelines.

In 1988 the Hallquist began working half-time at LLNL to devote more time to the development and support of LS-DYNA for automotive applications. By the end of 1988 it was obvious that a much more concentrated effort would be required in the development of LS-DYNA if problems in crashworthiness were to be properly solved; therefore, at the start of 1989 the Hallquist resigned from LLNL to continue code development full time at Livermore Software Technology Corporation. The 1989 version introduced many enhanced capabilities including a one-way treatment of slide surfaces with voids and friction; cross-sectional forces for structural elements; an optional user specified minimum time step size for shell elements using elastic and elastoplastic material models; nodal accelerations in the time history database; a compressible Mooney-Rivlin material model; a closed-form update shell plasticity model; a general rubber material model; unique penalty specifications for each slide surface; external work tracking; optional time step criterion for 4-node shell elements; and internal element sorting to allow full vectorization of right-hand-side force assembly.

2.1 Features add in 1989-1990

Throughout the past decade, considerable progress has been made as may be seen in the chronology of the developments which follows. During 1989 many extensions and developments were completed, and in 1990 the following capabilities were delivered to users:

- arbitrary node and element numbers,
- fabric model for seat belts and airbags,
- composite glass model,
- vectorized type 3 contact and single surface contact,
- many more I/O options,
- all shell materials available for 8 node brick shell,
- strain rate dependent plasticity for beams,
- fully vectorized iterative plasticity,
- interactive graphics on some computers,
- nodal damping,
- shell thickness taken into account in shell type 3 contact,
- shell thinning accounted for in type 3 and type 4 contact,
- soft stonewalls,
- print suppression option for node and element data,
- massless truss elements, rivets – based on equations of rigid body dynamics,
- massless beam elements, spot welds – based on equations of rigid body dynamics,

- expanded databases with more history variables and integration points,
- force limited resultant beam,
- rotational spring and dampers, local coordinate systems for discrete elements,
- resultant plasticity for C0 triangular element,
- energy dissipation calculations for stonewalls,
- hourglass energy calculations for solid and shell elements,
- viscous and Coulomb friction with arbitrary variation over surface,
- distributed loads on beam elements,
- Cowper and Symonds strain rate model,
- segmented stonewalls,
- stonewall Coulomb friction,
- stonewall energy dissipation,
- airbags (1990),
- nodal rigid bodies,
- automatic sorting of triangular shells into C0 groups,
- mass scaling for quasi static analyses,
- user defined subroutines,
- warpage checks on shell elements,
- thickness consideration in all contact types,
- automatic orientation of contact segments,
- sliding interface energy dissipation calculations,
- nodal force and energy database for applied boundary conditions,
- defined stonewall velocity with input energy calculations,

2.2 Options added in 1991-1992

- rigid/deformable material switching,
- rigid bodies impacting rigid walls,
- strain-rate effects in metallic honeycomb model 26,
- shells and beams interfaces included for subsequent component analyses,
- external work computed for prescribed displacement/velocity/accelerations,
- linear constraint equations,

- MPGS database,
- MOVIE database,
- Slideline interface file,
- automated contact input for all input types,
- automatic single surface contact without element orientation,
- constraint technique for contact,
- cut planes for resultant forces,
- crushable cellular foams,
- urethane foam model with hysteresis,
- subcycling,
- friction in the contact entities,
- strains computed and written for the 8 node thick shells,
- “good” 4 node tetrahedron solid element with nodal rotations,
- 8 node solid element with nodal rotations,
- 2×2 integration for the membrane element,
- Belytschko-Schwer integrated beam,
- thin-walled Belytschko-Schwer integrated beam,
- improved LS-DYNA database control,
- null material for beams to display springs and seatbelts in TAURUS,
- parallel implementation on Cray and SGI computers,
- coupling to rigid body codes,
- seat belt capability.

2.3 Options added in 1993-1994

- Arbitrary Lagrangian Eulerian brick elements,
- Belytschko-Wong-Chiang quadrilateral shell element,
- Warping stiffness in the Belytschko-Tsay shell element,
- Fast Hughes-Liu shell element,
- Fully integrated brick shell element,
- Discrete 3D beam element,
- Generalized dampers,

- Cable modeling,
- Airbag reference geometry,
- Multiple jet model,
- Generalized joint stiffnesses,
- Enhanced rigid body to rigid body contact,
- Orthotropic rigid walls,
- Time zero mass scaling,
- Coupling with USA (Underwater Shock Analysis),
- Layered spot welds with failure based on resultants or plastic strain,
- Fillet welds with failure,
- Butt welds with failure,
- Automatic eroding contact,
- Edge-to-edge contact,
- Automatic mesh generation with contact entities,
- Drawbead modeling,
- Shells constrained inside brick elements,
- NIKE3D coupling for springback,
- Barlat's anisotropic plasticity,
- Superplastic forming option,
- Rigid body stoppers,
- Keyword input,
- Adaptivity,
- First MPP (Massively Parallel) version with limited capabilities.
- Built in least squares fit for rubber model constitutive constants,
- Large hysteresis in hyperelastic foam,
- Bilhku/Dubois foam model,
- Generalized rubber model,

2.4 Version 936

New options added to version 936 in 1995 include:

- Belytschko - Leviathan Shell

- Automatic switching between rigid and deformable bodies.
- Accuracy on SMP machines to give identical answers on one, two or more processors.
- Local coordinate systems for cross-section output can now be specified.
- Null material for shell elements.
- Global body force loads now may be applied to a subset of materials.
- User defined loading subroutine.
- Improved interactive graphics.
- New initial velocity options for specifying rotational velocities.
- Geometry changes after dynamic relaxation can be considered for initial velocities.
- Velocities may also be specified by using material or part ID's.
- Improved speed of brick element hourglass force and energy calculations.
- Pressure outflow boundary conditions have been added for the ALE options.
- More user control for hourglass control constants for shell elements.
- Full vectorization in constitutive models for foam, models 57 and 63.
- Damage mechanics plasticity model, material 81,
- General linear viscoelasticity with 6 term prony series.
- Least squares fit for viscoelastic material constants.
- Table definitions for strain rate effects in material type 24.
- Improved treatment of free flying nodes after element failure.
- Automatic projection of nodes in CONTACT_TIED to eliminate gaps in the surface.
- More user control over contact defaults.
- Improved interpenetration warnings printed in automatic contact.
- Flag for using actual shell thickness in single surface contact logic rather than the default.
- Definition by exempted part ID's.
- Airbag to Airbag venting/segmented airbags are now supported.
- Airbag reference geometry speed improvements by using the reference geometry for the time step size calculation.
- Isotropic airbag material may now be directly for cost efficiency.
- Airbag fabric material damping is now specified as the ratio of critical damping.

- Ability to attach jets to the structure so the airbag, jets, and structure to move together.
- PVM 5.1 Madymo coupling is available.
- Meshes are generated within LS-DYNA3D for all standard contact entities.
- Joint damping for translational motion.
- Angular displacements, rates of displacements, damping forces, etc. in JNTFORC file.
- Link between LS-NIKE3D to LS-DYNA3D via *INITIAL_STRESS keywords.
- Trim curves for metal forming springback.
- Sparse equation solver for springback.
- Improved mesh generation for IGES and VDA provides a mesh that can directly be used to model tooling in metal stamping analyses.

2.5 Version 940

New options added to Version 940 in 1996 and 1997:

- Part/Material ID's may be specified with 8 digits.
- Rigid body motion can be prescribed in a local system fixed to the rigid body.
- Nonlinear least squares fit available for the Ogden rubber model.
- Least squares fit to the relaxation curves for the viscoelasticity in rubber.
- Fu-Chang rate sensitive foam.
- 6 term Prony series expansion for rate effects in model 57-now 73
- Viscoelastic material model 76 implemented for shell elements.
- Mechanical threshold stress (MTS) plasticity model for rate effects.
- Thermoelastic-plastic material model for Hughes-Liu beam element.
- Ramberg-Osgood soil model
- Invariant local coordinate systems for shell elements are optional.
- Second order accurate stress updates.
- Four-noded, linear, tetrahedron element.
- Co-rotational solid element for foam that can invert without stability problems.
- Improved speed in rigid body to rigid body contacts.
- Improved searching for the a_3, a_5 and a10 contact types.
- Invariant results on shared memory parallel machines with the a_n contact types.

- Thickness offsets in type 8 and 9 tie break contact algorithms.
- Bucket sort frequency can be controlled by a load curve for airbag applications.
- In automatic contact each part ID in the definition may have unique:
 - Static coefficient of friction
 - Dynamic coefficient of friction
 - Exponential decay coefficient
 - Viscous friction coefficient
 - Optional contact thickness
 - Optional thickness scale factor
 - Local penalty scale factor
- Automatic beam-to-beam, shell edge-to-beam, shell edge-to-shell edge and single surface contact algorithm.
- Release criteria may be a multiple of the shell thickness in types a_3, a_5, a10, 13, and 26 contact.
- Force transducers to obtain reaction forces in automatic contact definitions. Defined manually via segments, or automatically via part ID's.
- Searching depth can be defined as a function of time.
- Bucket sort frequency can be defined as a function of time.
- Interior contact for solid (foam) elements to prevent "negative volumes."
- Locking joint
- Temperature dependent heat capacity added to Wang-Nefske inflator models.
- Wang Hybrid inflator model [Wang, 1996] with jetting options and bag-to-bag venting.
- Aspiration included in Wang's hybrid model [Nucholtz, Wang, Wylie, 1996].
- Extended Wang's hybrid inflator with a quadratic temperature variation for heat capacities [Nusholtz, 1996].
- Fabric porosity added as part of the airbag constitutive model.
- Blockage of vent holes and fabric in contact with structure or itself considered in venting with leakage of gas.
- Option to delay airbag liner with using the reference geometry until the reference area is reached.
- Birth time for the reference geometry.
- Multi-material Euler/ALE fluids,
 - 2nd order accurate formulations.

- Automatic coupling to shell, brick, or beam elements
 - Coupling using LS-DYNA contact options.
 - Element with fluid + void and void material
 - Element with multi-materials and pressure equilibrium
- Nodal inertia tensors.
- 2D plane stress, plane strain, rigid, and axisymmetric elements
- 2D plane strain shell element
- 2D axisymmetric shell element.
- Full contact support in 2D, tied, sliding only, penalty and constraint techniques.
- Most material types supported for 2D elements.
- Interactive remeshing and graphics options available for 2D.
- Subsystem definitions for energy and momentum output. and many more enhancements not mentioned above.

2.6 Version 950

Capabilities added during 1997-1998 in Version 950 include:

- Adaptive refinement can be based on tooling curvature with FORMING contact.
- The display of draw beads is now possible since the draw bead data is output into the d3plot database.
- An adaptive box option, *DEFINE_BOX_ADAPTIVE, allows control over the refinement level and location of elements to be adapted.
- A root identification file, adapt.rid, gives the parent element ID for adapted elements.
- Draw bead box option, *DEFINE_BOX_DRAWBEAD, simplifies draw bead input.
- The new control option, CONTROL_IMPLICIT, activates an implicit solution scheme.
- 2D Arbitrary-Lagrangian-Eulerian elements.
- 2D automatic contact is defined by listing part ID's.
- 2D r-adaptivity for plane strain and axisymmetric forging simulations is available.
- 2D automatic non-interactive rezoning as in LS-DYNA2D.
- 2D plane strain and axisymmetric element with 2×2 selective-reduced integration are implemented.

- Implicit 2D solid and plane strain elements are available.
- Implicit 2D contact is available.
- The new keyword, *DELETE_CONTACT_2DAUTO, allows the deletion of 2D automatic contact definitions.
- The keyword, *LOAD_BEAM is added for pressure boundary conditions on 2D elements.
- A viscoplastic strain rate option is available for materials:
 - *MAT_PLASTIC_KINEMATIC
 - *MAT_JOHNSON_COOK
 - *MAT_POWER_LAW_PLASTICITY
 - *MAT_STRAIN_RATE_DEPENDENT_PLASTICITY
 - *MAT_PIECEWISE_LINEAR_PLASTICITY
 - *MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY
 - *MAT_ZERILLI-ARMSTRONG
 - *MAT_PLASTICITY_WITH_DAMAGE
 - *MAT_PLASTICITY_COMPRESSION_TENSION
- Material model, *MAT_PLASTICITY_WITH_DAMAGE, has a piecewise linear damage curve given by a load curve ID.
- The Arruda-Boyce hyper-viscoelastic rubber model is available, see *MAT_ARRUDA_BOYCE.
- Transverse-anisotropic-viscoelastic material for heart tissue, see *MAT_HEART_TISSUE.
- Lung hyper-viscoelastic material, see *MAT_LUNG_TISSUE.
- Compression/tension plasticity model, see *MAT_PLASTICITY_COMPRESSION_TENSION.
- The Lund strain rate model, *MAT_STEINBERG_LUND, is added to Steinberg-Guinan plasticity model.
- Rate sensitive foam model, *MAT_FU_CHANG_FOAM, has been extended to include engineering strain rates, etc.
- Model, *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY, is added for modeling the failure of aluminum.
- Material model, *MAT_SPECIAL_ORTHOTROPIC, added for television shadow mask problems.
- Erosion strain is implemented for material type, *MAT_BAMMAN_DAMAGE.

- The equation of state, *EOS_JWL, is available for modeling the expansion of explosive gases.
- The reference geometry option is extended for foam and rubber materials and can be used for stress initialization, see *INITIAL_FOAM_REFERENCE_GEOMETRY.
- A vehicle positioning option is available for setting the initial orientation and velocities, see *INITIAL_VEHICLE_KINEMATICS.
- A boundary element method is available for incompressible fluid dynamics problems.
- The thermal materials work with instantaneous coefficients of thermal expansion:
 - *MAT_ELASTIC_PLASTIC_THERMAL
 - *MAT_ORTHOTROPIC_THERMAL
 - *MAT_TEMPERATURE_DEPENDENT_ORTHOTROPIC
 - *MAT_ELASTIC_WITH_VISCOSITY.
- Airbag interaction flow rate versus pressure differences.
- Contact segment search option, [bricks first optional]
- A through thickness Gauss integration rule with 1-10 points is available for shell elements. Previously, 5 were available.
- Shell element formulations can be changed in a full deck restart.
- The tied interface which is based on constraint equations, TIED_SURFACE_TO_SURFACE, can now fail with FAILURE option.
- A general failure criteria for solid elements is independent of the material type, see *MAT_ADD_EROSION
- Load curve control can be based on thinning and a flow limit diagram, see *DEFINE_CURVE_FEEDBACK.
- An option to filter the spotweld resultant forces prior to checking for failure has been added the option, *CONSTRAINED_SPOTWELD, by appending, FILTERED_FORCE, to the keyword.
- Bulk viscosity is available for shell types 1, 2, 10, and 16.
- When defining the local coordinate system for the rigid body inertia tensor a local coordinate system ID can be used. This simplifies dummy positioning.
- Prescribing displacements, velocities, and accelerations is now possible for rigid body nodes.
- One-way flow is optional for segmented airbag interactions.
- Pressure time history input for airbag type, LINEAR_FLUID, can be used.
- An option is available to independently scale system damping by part ID in each of the global directions.

- An option is available to independently scale global system damping in each of the global directions.
- Added option to constrain global DOF along lines parallel with the global axes. The keyword is *CONSTRAINED_GLOBAL. This option is useful for adaptive remeshing.
- Beam end code releases are available, see *ELEMENT_BEAM.
- An initial force can be directly defined for the cable material, *MAT_CABLE_DISCRETE_BEAM. The specification of slack is not required if this option is used.
- Airbag pop pressure can be activated by accelerometers.
- Termination may now be controlled by contact, via *TERMINATION_CONTACT.
- Modified shell elements types 8, 10 and the warping stiffness option in the Belytschko-Tsay shell to ensure orthogonality with rigid body motions in the event that the shell is badly warped. This is optional in the Belytschko-Tsay shell and the type 10 shell.
- A one point quadrature brick element with an exact hourglass stiffness matrix has been implemented for implicit and explicit calculations.
- Automatic file length determination for d3plot binary database is now implemented. This insures that at least a single state is contained in each d3plot file and eliminates the problem with the states being split between files.
- The dump files, which can be very large, can be placed in another directory by specifying d=/home/user /test/d3dump on the execution line.
- A print flag controls the output of data into the MATSUM and RBDOUT files by part ID's. The option, PRINT, has been added as an option to the *PART keyword.
- Flag has been added to delete material data from the d3thdt file. See *DATABASE_EXTENT_BINARY and column 25 of the 19th control card in the structured input.
- After dynamic relaxation completes, a file is written giving the displaced state which can be used for stress initialization in later runs.

2.7 Version 960

Capabilities added during 1998-2000 in Version 960. Most new capabilities work on both the MPP and SMP versions; however, the capabilities that are implemented for the SMP version only, which were not considered critical for this release, are flagged below. These SMP unique capabilities are being extended for MPP calculations and will be available in the near future. The implicit capabilities for MPP require the development of a scalable eigenvalue solver, which is under development for a later release of LS-DYNA.

- Incompressible flow solver is available. Structural coupling is not yet implemented.
- Adaptive mesh coarsening can be done before the implicit spring back calculation in metal forming applications.
- Two-dimensional adaptivity can be activated in both implicit and explicit calculations. (SMP version only)
- An internally generated smooth load curve for metal forming tool motion can be activated with the keyword: *DEFINE_CURVE_SMOOTH.
- Torsional forces can be carried through the deformable spot welds by using the contact type: *CONTACT_SPOTWELD_WITH_TORSION (SMP version only with a high priority for the MPP version if this option proves to be stable.)
- Tie break automatic contact is now available via the *CONTACT_AUTOMATIC_..._TIEBREAK options. This option can be used for glued panels. (SMP only)
- *CONTACT_RIGID_SURFACE option is now available for modeling road surfaces (SMP version only).
- Fixed rigid walls PLANAR and PLANAR_FINITE are represented in the binary output file by a single shell element.
- Interference fits can be modeled with the INTERFERENCE option in contact.
- A layered shell theory is implemented for several constitutive models including the composite models to more accurately represent the shear stiffness of laminated shells.
- Damage mechanics is available to smooth the post-failure reduction of the resultant forces in the constitutive model *MAT_SPOTWELD_DAMAGE.
- Finite elastic strain isotropic plasticity model is available for solid elements. *MAT_FINITE_ELASTIC_STRAIN_PLASTICITY.
- A shape memory alloy material is available: *MAT_SHAPE_MEMORY.
- Reference geometry for material, *MAT_MODIFIED_HONEYCOMB, can be set at arbitrary relative volumes or when the time step size reaches a limiting value. This option is now available for all element types including the fully integrated solid element.
- Non orthogonal material axes are available in the airbag fabric model. See *MAT_FABRIC.
- Other new constitutive models include for the beam elements:
 - *MAT_MODIFIED_FORCE_LIMITED
 - *MAT_SEISMIC_BEAM
 - *MAT_CONCRETE_BEAM
- for shell and solid elements:

- *MAT_ELASTIC_VISCOPLASTIC_THERMAL
- for the shell elements:
 - *MAT_GURSON
 - *MAT_GEPLASTIC_SRATE2000
 - *MAT_ELASTIC_VISCOPLASTIC_THERMAL
 - *MAT_COMPOSITE_LAYUP
 - *MAT_COMPOSITE_LAYUP
 - *MAT_COMPOSITE_direct
- for the solid elements:
 - *MAT_JOHNSON_HOLMQUIST_CERAMICS
 - *MAT_JOHNSON_HOLMQUIST_CONCRETE
 - *MAT_INV_HYPERBOLIC_SIN
 - *MAT_UNIFIED_CREEP
 - *MAT_SOIL_BRICK
 - *MAT_DRUCKER_PRAGER
 - *MAT_RC_SHEAR_WALL
- and for all element options a very fast and efficient version of the Johnson-Cook plasticity model is available:
 - *MAT_SIMPLIFIED_JOHNSON_COOK
- A fully integrated version of the type 16 shell element is available for the resultant constitutive models.
- A nonlocal failure theory is implemented for predicting failure in metallic materials. The keyword *MAT_NONLOCAL activates this option for a subset of elasto-plastic constitutive models.
- A discrete Kirchhoff triangular shell element (DKT) for explicit analysis with three in plane integration points is flagged as a type 17 shell element. This element has much better bending behavior than the C0 triangular element.
- A discrete Kirchhoff linear triangular and quadrilateral shell element is available as a type 18 shell. This shell is for extracting normal modes and static analysis.
- A C0 linear 4-node quadrilateral shell element is implemented as element type 20 with drilling stiffness for normal modes and static analysis.
- An assumed strain linear brick element is available for normal modes and statics.
- The fully integrated thick shell element has been extended for use in implicit calculations.

- A fully integrated thick shell element based on an assumed strain formulation is now available. This element uses a full 3D constitutive model which includes the normal stress component and, therefore, does not use the plane stress assumption.
- The 4-node constant strain tetrahedron element has been extended for use in implicit calculations.
- Relative damping between parts is available, see *DAMPING_RELATIVE (SMP only).
- Preload forces can be input for the discrete beam elements.
- Objective stress updates are implemented for the fully integrated brick shell element.
- Acceleration time histories can be prescribed for rigid bodies.
- Prescribed motion for nodal rigid bodies is now possible.
- Generalized set definitions, i.e., SET_SHELL_GENERAL etc. provide much flexibility in the set definitions.
- The command "sw4." will write a state into the dynamic relaxation file, D3DRLF, during the dynamic relaxation phase if the d3drf file is requested in the input.
- Added mass by PART ID is written into the matsum file when mass scaling is used to maintain the time step size, (SMP version only).
- Upon termination due to a large mass increase during a mass scaled calculation a print summary of 20 nodes with the maximum added mass is printed.
- Eigenvalue analysis of models containing rigid bodies is now available using BC-SLIB-EXT solvers from Boeing. (SMP version only).
- Second order stress updates can be activated by part ID instead of globally on the *CONTROL_ACCURACY input.
- Interface frictional energy is optionally computed for heat generation and is output into the interface force file (SMP version only).
- The interface force binary database now includes the distance from the contact surface for the FORMING contact options. This distance is given after the nodes are detected as possible contact candidates. (SMP version only).
- Type 14 acoustic brick element is implemented. This element is a fully integrated version of type 8, the acoustic element (SMP version only).
- A flooded surface option for acoustic applications is available (SMP version only).
- Attachment nodes can be defined for rigid bodies. This option is useful for NVH applications.
- CONSTRAINED_POINTS tie any two points together. These points must lie on a shell element.
- Soft constraint is available for edge-to-edge contact in type 26 contact.

- CONSTAINED_INTERPOLATION option for beam to solid interfaces and for spreading the mass and loads. (SMP version only).
- A database option has been added that allows the output of added mass for shell elements instead of the time step size.
- A new contact option allows the inclusion of all internal shell edges in contact type *CONTACT_GENERAL, type 26. This option is activated by adding INTERIOR option.
- A new option allows the use deviatoric strain rates rather than total rates in material model 24 for the Cowper-Symonds rate model.
- The CADFEM option for ASCII databases is now the default. Their option includes more significant figures in the output files.
- When using deformable spot welds, the added mass for spot welds is now printed for the case where global mass scaling is activated. This output is in the log file, d3hsp file, and the messag file.
- Initial penetration warnings for edge-to-edge contact are now written into the MESSAG file and the D3HSP file.
- Each compilation of LS-DYNA is given a unique version number.
- Finite length discrete beams with various local axes options are now available for material types 66, 67, 68, 93, and 95. In this implementation the absolute value of SCOOR must be set to 2 or 3 in the *SECTION_BEAM input.
- New discrete element constitutive models are available:
 - *MAT_ELASTIC_SPRING_DISCRETE_BEAM
 - *MAT_INELASTIC_SPRING_DISCRETE_BEAM
 - *MAT_ELASTIC_6DOF_SPRING_DISCRETE_BEAM
 - *MAT_INELASTIC_6DOF_SPRING_DISCRETE_BEAM

The latter two can be used as finite length beams with local coordinate systems.

- Moving SPC's are optional in that the constraints are applied in a local system that rotates with the 3 defining nodes.
- A moving local coordinate system, CID, can be used to determine orientation of discrete beam elements.
- Modal superposition analysis can be performed after an eigenvalue analysis. Stress recovery is based on type 18 shell and brick (SMP only).
- Rayleigh damping input factor is now input as a fraction of critical damping, i.e. 0.10. The old method required the frequency of interest and could be highly unstable for large input values.
- Airbag option "SIMPLE_PRESSURE_VOLUME" allows for the constant CN to be replaced by a load curve for initialization. Also, another load curve can be defined

which allows CN to vary as a function of time during dynamic relaxation. After dynamic relaxation CN can be used as a fixed constant or load curve.

- Hybrid inflator model utilizing CHEMKIN and NIST databases is now available. Up to ten gases can be mixed.
- Option to track initial penetrations has been added in the automatic SMP contact types rather than moving the nodes back to the surface. This option has been available in the MPP contact for some time. This input can be defined on the fourth card of the *CONTROL_CONTACT input and on each contact definition on the third optional card in the *CONTACT definitions.
- If the average acceleration flag is active, the average acceleration for rigid body nodes is now written into the d3thdt and nodout files. In previous versions of LS-DYNA, the accelerations on rigid nodes were not averaged.
- A capability to initialize the thickness and plastic strain in the crash model is available through the option *INCLUDE_STAMPED_PART, which takes the results from the LS-DYNA stamping simulation and maps the thickness and strain distribution onto the same part with a different mesh pattern.
- A capability to include finite element data from other models is available through the option, *INCLUDE_TRANSFORM. This option will take the model defined in an INCLUDE file: offset all ID's; translate, rotate, and scale the coordinates; and transform the constitutive constants to another set of units.

2.8 Version 970

Many new capabilities were added during 2001-2002 to create version 970 of LS-DYNA. Some of the new features, which are also listed below, were also added to later releases of version 960. Most new explicit capabilities work for both the MPP and SMP versions; however, the implicit capabilities for MPP require the development of a scalable eigenvalue solver and a parallel implementation of the constraint equations into the global matrices. This work is underway. A later release of version 970 is planned that will be scalable for implicit solutions.

- MPP decomposition can be controlled using *CONTROL_MPP_DECOMPOSITION commands in the input deck.
- The MPP arbitrary Lagrangian-Eulerian fluid capability now works for airbag deployment in both SMP and MPP calculations.
- Euler-to-Euler coupling is now available through the keyword *CONSTRAINED_EULER_TO_EULER.
- Up to ten ALE multi-material groups may now be defined. The previous limit was three groups.

- Volume fractions can be automatically assigned during initialization of multi-material cells. See the GEOMETRY option of *INITIAL_VOLUME_FRACTION.
- A new ALE smoothing option is available to accurately predict shock fronts.
- DATABASE_FSI activates output of fluid-structure interaction data to ASCII file DBFSI.
- Point sources for airbag inflators are available. The origin and mass flow vector of these inflators are permitted to vary with time.
- A majority of the material models for solid materials are available for calculations using the SPH (Smooth Particle Hydrodynamics) option.
- The Element Free Galerkin method (EFG or meshfree) is available for two-dimensional and three-dimensional solids. This new capability is not yet implemented for MPP applications.
- A binary option for the ASCII files is now available. This option applies to all ASCII files and results in one binary file that contains all the information normally spread between a large number of separate ASCII files.
- Material models can now be defined by numbers rather than long names in the keyword input. For example the keyword *MAT_PIECEWISE_LINEAR_PLASTICITY can be replaced by the keyword: *MAT_024.
- An embedded NASTRAN reader for direct reading of NASTRAN input files is available. This option allows a typical input file for NASTRAN to be read directly and used without additional input. See the *INCLUDE_NASTRAN keyword.
- Names in the keyword input can represent numbers if the *PARAMETER option is used to relate the names and the corresponding numbers.
- Model documentation for the major ASCII output files is now optional. This option allows descriptors to be included within the ASCII files that document the contents of the file.
- ID's have been added to the following keywords:
 - *BOUNDARY_PRESCRIBED_MOTION
 - *BOUNDARY_PRESCRIBED_SPC
 - *CONSTRAINED_GENERALIZED_WELD
 - *CONSTRAINED_JOINT
 - *CONSTRAINED_NODE_SET
 - *CONSTRAINED_RIVET
 - *CONSTRAINED_SPOTWELD
 - *DATABASE_CROSS_SECTION
 - *ELEMENT_MASS

- The *DATABASE_ADAMS keyword is available to output a modal neutral file d3mnf. This is available upon customer request since it requires linking to an ADAMS library file.
- Penetration warnings for the contact option, “ignore initial penetration,” are added as an option. Previously, no penetration warnings were written when this contact option was activated.
- Penetration warnings for nodes in-plane with shell mid-surface are printed for the AUTOMATIC contact options. Previously, these nodes were ignored since it was assumed that they belonged to a tied interface where an offset was not used; consequently, they should not be treated in contact.
- For the arbitrary spot weld option, the spot welded nodes and their contact segments are optionally written into the d3hsp file. See *CONTROL_CONTACT.
- For the arbitrary spot weld option, if a segment cannot be found for the spot welded node, an option now exists to error terminate. See *CONTROL_CONTACT.
- Spot weld resultant forces are written into the swforc file for solid elements used as spot welds.
- Solid materials have now been added to the failed element report and additional information is written for the “node is deleted” messages.
- A new option for terminating a calculation is available, *TERMINATION_CURVE.
- A 10-noded tetrahedron solid element is available with either a 4 or 5 point integration rule. This element can also be used for implicit solutions.
- A new 4 node linear shell element is available that is based on Wilson’s plate element combined with a Pian-Sumihara membrane element. This is shell type 21.
- A shear panel element has been added for linear applications. This is shell type 22. This element can also be used for implicit solutions.
- A null beam element for visualization is available. The keyword to define this null beam is *ELEMENT_PLOTEL. This element is necessary for compatibility with NASTRAN.
- A scalar node can be defined for spring-mass systems. The keyword to define this node is *NODE_SCALAR. This node can have from 1 to 6 scalar degrees-of-freedom.
- A thermal shell has been added for through-thickness heat conduction. Internally, 8 additional nodes are created, four above and four below the mid-surface of the shell element. A quadratic temperature field is modeled through the shell thickness. Internally, the thermal shell is a 12 node solid element.
- A beam OFFSET option is available for the *ELEMENT_BEAM definition to permit the beam to be offset from its defining nodal points. This has the advantage that all beam formulations can now be used as shell stiffeners.

- A beam ORIENTATION option for orienting the beams by a vector instead of the third node is available in the *ELEMENT_BEAM definition for NASTRAN compatibility.
- Non-structural mass has been added to beam elements for modeling trim mass and for NASTRAN compatibility.
- An optional checking of shell elements to avoid abnormal terminations is available. See *CONTROL_SHELL. If this option is active, every shell is checked each time step to see if the distortion is so large that the element will invert, which will result in an abnormal termination. If a bad shell is detected, either the shell will be deleted or the calculation will terminate. The latter is controlled by the input.
- An offset option is added to the inertia definition. See *ELEMENT_INERTIA_OFFSET keyword. This allows the inertia tensor to be offset from the nodal point.
- Plastic strain and thickness initialization is added to the draw bead contact option. See *CONTACT_DRAWBEAD_INITIALIZE.
- Tied contact with offsets based on both constraint equations and beam elements for solid elements and shell elements that have 3 and 6 degrees-of-freedom per node, respectively. See BEAM_OFFSET and CONSTRAINED_OFFSET contact options. These options will not cause problems for rigid body motions.
- The segment-based (SOFT = 2) contact is implemented for MPP calculations. This enables airbags to be easily deployed on the MPP version.
- Improvements are made to segment-based contact for edge-to-edge and sliding conditions, and for contact conditions involving warped segments.
- An improved interior contact has been implemented to handle large shear deformations in the solid elements. A special interior contact algorithm is available for tetrahedron elements.
- Coupling with MADYMO 6.0 uses an extended coupling that allows users to link most MADYMO geometric entities with LS-DYNA FEM simulations. In this coupling MADYMO contact algorithms are used to calculate interface forces between the two models.
- Release flags for degrees-of-freedom for nodal points within nodal rigid bodies are available. This makes the nodal rigid body option nearly compatible with the RBE2 option in NASTRAN.
- Fast updates of rigid bodies for metal forming applications can now be accomplished by ignoring the rotational degrees-of-freedom in the rigid bodies that are typically inactive during sheet metal stamping simulations. See the keyword: *CONTROL_RIGID.
- Center of mass constraints can be imposed on nodal rigid bodies with the SPC option in either a local or a global coordinate system.

- Joint failure based on resultant forces and moments can now be used to simulate the failure of joints.
- CONSTRAINED_JOINT_STIFFNESS now has a TRANSLATIONAL option for the translational and cylindrical joints.
- Joint friction has been added using table look-up so that the frictional moment can now be a function of the resultant translational force.
- The nodal constraint options *CONSTRAINED_INTERPOLATION and *CONSTRAINED_LINEAR now have a local option to allow these constraints to be applied in a local coordinate system.
- Mesh coarsening can now be applied to automotive crash models at the beginning of an analysis to reduce computation times. See the new keyword: *CONTROL_COARSEN.
- Force versus time seatbelt pretensioner option has been added.
- Both static and dynamic coefficients of friction are available for seat belt slip rings. Previously, only one friction constant could be defined.
- *MAT_SPOTWELD now includes a new failure model with rate effects as well as additional failure options.
- Constitutive models added for the discrete beam elements:
 - *MAT_1DOF_GENERALIZED_SPRING
 - *MAT_GENERAL_NONLINEAR_6dof_DISCRETE_BEAM
 - *MAT_GENERAL_NONLINEAR_1dof_DISCRETE_BEAM
 - *MAT_GENERAL_SPRING_DISCRETE_BEAM
 - *MAT_GENERAL_JOINT_DISCRETE_BEAM
 - *MAT_SEISMIC_ISOLATOR
- for shell and solid elements:
 - *MAT_PLASTICITY_WITH_DAMAGE_ORTHO
 - *MAT_SIMPLIFIED_JOHNSON_COOK_ORTHOTROPIC_DAMAGE
 - *MAT_HILL_3R
 - *MAT_GURSON_RCDC
- for the solid elements:
 - *MAT_SPOTWELD
 - *MAT_HILL_FOAM
 - *MAT_WOOD
 - *MAT_VISCOELASTIC_HILL_FOAM

- *MAT_LOW_DENSITY_SYNTHETIC_FOAM
- *MAT_RATE_SENSITIVE_POLYMER
- *MAT_QUASILINEAR_VISCOELASTIC
- *MAT_TRANSVERSELY_ANISOTROPIC_CRUSHABLE_FOAM
- *MAT_VACUUM
- *MAT_MODIFIED_CRUSHABLE_FOAM
- *MAT_PITZER_CRUSHABLE_FOAM
- *MAT_JOINTED_ROCK
- *MAT_SIMPLIFIED_RUBBER
- *MAT_FHWA_SOIL
- *MAT_SCHWER_MURRAY_CAP_MODEL
- Failure time added to MAT_EROSION for solid elements.
- Damping in the material models *MAT_LOW_DENSITY_FOAM and *MAT_LOW_DENSITY_VISCOUS_FOAM can now be a tabulated function of the smallest stretch ratio.
- The material model *MAT_PLASTICITY_WITH_DAMAGE allows the table definitions for strain rate.
- Improvements in the option *INCLUDE_STAMPED_PART now allow all history data to be mapped to the crash part from the stamped part. Also, symmetry planes can be used to allow the use of a single stamping to initialize symmetric parts.
- Extensive improvements in trimming result in much better elements after the trimming is completed. Also, trimming can be defined in either a local or global coordinate system. This is a new option in *DEFINE_CURVE_TRIM.
- An option to move parts close before solving the contact problem is available, see *CONTACT_AUTO_MOVE.
- An option to add or remove discrete beams during a calculation is available with the new keyword: *PART_SENSOR.
- Multiple jetting is now available for the Hybrid and Chemkin airbag inflator models.
- Nearly all constraint types are now handled for implicit solutions.
- Calculation of constraint and attachment modes can be easily done by using the option: *CONTROL_IMPLICIT_MODES.
- Penalty option, see *CONTROL_CONTACT, now applies to all *RIGIDWALL options and is always used when solving implicit problems.
- Solid elements types 3 and 4, the 4 and 8 node elements with 6 degrees-of-freedom per node, are available for implicit solutions.

- The warping stiffness option for the Belytschko-Tsay shell is implemented for implicit solutions. The Belytschko-Wong-Chang shell element is now available for implicit applications. The full projection method is implemented due to its accuracy over the drill projection.
- Rigid to deformable switching is implemented for implicit solutions.
- Automatic switching can be used to switch between implicit and explicit calculations. See the keyword: *CONTROL_IMPLICIT_GENERAL.
- Implicit dynamics rigid bodies are now implemented. See the keyword *CONTROL_IMPLICIT_DYNAMIC.
- Eigenvalue solutions can be intermittently calculated during a transient analysis.
- A linear buckling option is implemented. See the new control input: *CONTROL_IMPLICIT_BUCKLE
- Implicit initialization can be used instead of dynamic relaxation. See the keyword *CONTROL_DYNAMIC_RELAXATION where the parameter, IDFLG, is set to 5.
- Superelements, i.e., *ELEMENT_DIRECT_MATRIX_INPUT, are now available for implicit applications.
- There is an extension of the option, *BOUNDARY_CYCLIC, to symmetry planes in the global Cartesian system. Also, automatic sorting of nodes on symmetry planes is now done by LS-DYNA.
- Modeling of wheel-rail contact for railway applications is now available, see *RAIL_TRACK and *RAIL_TRAIN.
- A new, reduced CPU, element formulation is available for vibration studies when elements are aligned with the global coordinate system. See *SECTION_SOLID and *SECTION_SHELL formulation 98.
- An option to provide approximately constant damping over a range of frequencies is implemented, see *DAMPING_FREQUENCY_RANGE.

3

Preliminaries

NOTE: Einstein summation convention is used. *For each repeated index, there is an implied summation.*

Consider the body shown in [Figure 3.1](#). We are interested in time-dependent deformation for which a point in b initially at X_α ($\alpha = 1, 2, 3$) in a fixed rectangular Cartesian coordinate system moves to a point x_i ($i = 1, 2, 3$) in the same coordinate system. Since a Lagrangian formulation is considered, the deformation can be expressed in terms of the convected coordinates X_α , and time t

$$x_i = x_i(X_\alpha, t). \quad (3.1)$$

At time $t = 0$, we have the initial conditions

$$\begin{aligned} x_i(\mathbf{X}, 0) &= X_i \\ \dot{x}_i(\mathbf{X}, 0) &= V_i(\mathbf{X}) \end{aligned} \quad (3.2)$$

where \mathbf{V} is the initial velocity.

3.1 Governing Equations

We seek a solution to the momentum equation

$$\sigma_{ij,j} + \rho f_i = \rho \ddot{x}_i \quad (3.3)$$

satisfying the traction boundary conditions,

$$\sigma_{ij} n_j = t_i(t), \quad (3.4)$$

on boundary ∂b_1 , the displacement boundary conditions,

$$x_i(X_\alpha, t) = D_i(t), \quad (3.5)$$

on boundary ∂b_2 , and the contact discontinuity condition,

$$(\sigma_{ij}^+ - \sigma_{ij}^-) n_i = 0, \quad (3.6)$$

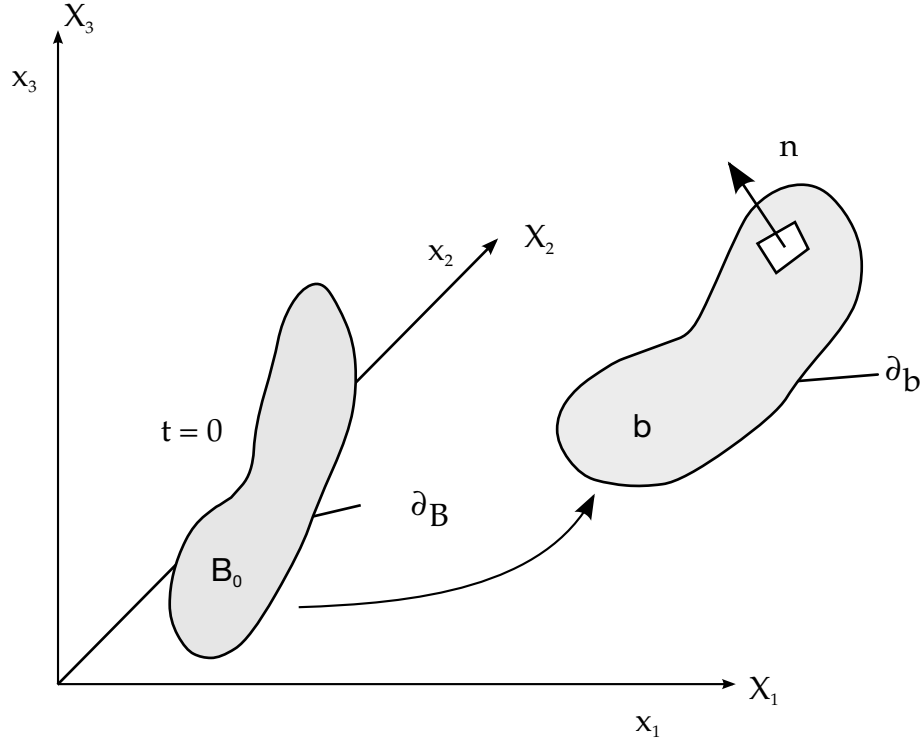


Figure 3.1. Notation.

along an interior boundary ∂b_3 when $x_i^+ = x_i^-$. Here σ is the Cauchy stress, ρ is the current density, \mathbf{f} is the body force density, and $\ddot{\mathbf{x}}$ is acceleration. The comma on $\sigma_{ij,j}$ denotes covariant differentiation, and n_j is a unit outward normal to a boundary element on ∂b .

Mass conservation is trivially stated as

$$\rho V = \rho_0 \quad (3.7)$$

where V is the relative volume, i.e., the determinant of the deformation gradient matrix, F_{ij} ,

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \quad (3.8)$$

and ρ_0 is the reference density. The energy equation

$$\dot{E} = V s_{ij} \dot{\epsilon}_{ij} - (p + q) \dot{V} \quad (3.9)$$

is integrated in time and is used for evaluating equations of state and tracking the global energy balance. In [Equation \(3.9\)](#), s_{ij} and p represent the deviatoric stresses and pressure,

$$s_{ij} = \sigma_{ij} + (p + q) \delta_{ij} \quad (3.10)$$

$$\begin{aligned} p &= -\frac{1}{3} \sigma_{ij} \delta_{ij} - q \\ &= -\frac{1}{3} \sigma_{kk} - q \end{aligned} \quad (3.11)$$

respectively, where q is the bulk viscosity, δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$; otherwise $\delta_{ij} = 0$), and $\dot{\epsilon}_{ij}$ is the strain rate tensor. The strain rates and bulk viscosity are discussed later.

We can write:

$$\int_v (\rho \ddot{x}_i - \sigma_{ij,j} - \rho f_i) \delta x_i dv + \int_{\partial b_1} (\sigma_{ij} n_j - t_i) \delta x_i ds + \int_{\partial b_3} (\sigma_{ij}^+ - \sigma_{ij}^-) n_j \delta x_i ds = 0 \quad (3.12)$$

where δx_i satisfies all boundary conditions on ∂b_2 , and the integrations are over the current geometry. Application of the divergence theorem gives

$$\int_v (\sigma_{ij} \delta x_i)_{,j} dv = \int_{\partial b_1} \sigma_{ij} n_j \delta x_i ds + \int_{\partial b_3} (\sigma_{ij}^+ - \sigma_{ij}^-) n_j \delta x_i ds \quad (3.13)$$

and noting that

$$(\sigma_{ij} \delta x_i)_{,j} - \sigma_{ij,j} \delta x_i = \sigma_{ij} \delta x_{i,j} \quad (3.14)$$

leads to the weak form of the equilibrium equation,

$$\delta \pi = \int_v \rho \ddot{x}_i \delta x_i dv + \int_v \sigma_{ij} \delta x_{i,j} dv - \int_v \rho f_i \delta x_i dv - \int_{\partial b_1} t_i \delta x_i ds = 0, \quad (3.15)$$

which is a statement of the principle of virtual work.

We superimpose a mesh of finite elements interconnected at nodal points on the reference configuration and track particles through time, i.e.,

$$x_i(X_\alpha, t) = x_i(X_\alpha(\xi, \eta, \zeta), t) = \sum_{j=1}^k N_j(\xi, \eta, \zeta) x_i^j(t) \quad (3.16)$$

where N_j are shape (interpolation) functions in the parametric coordinates (ξ, η, ζ) , k is the number of nodal points defining the element, and x_i^j is the nodal coordinate of the j^{th} node in the i^{th} direction. Each shape function has a finite support that is limited to the elements for which its associated node is a member (hence the name *finite element method*). Consequently, within each element the interpolation only depends on the nodal values for the nodes in *that* element, and hence expressions like Equation (3.16) are meaningful.

The condition $\delta \pi = 0$ holds for all variations, δx_i , and, in particular, it holds for variations along the shape functions. In each of the 3 Cartesian directions upon setting the variation to one of the shape functions the weak form reduces to a necessary (but not sufficient) condition that must be satisfied by any solution so that the

$$\text{number of equations} = 3 \times \text{number of nodes}.$$

At this stage, it is useful to introduce a vector space having dimension $\mathbb{R}^{(\text{number of nodes})}$ with a corresponding cartesian basis $\{\mathbf{e}_i'\}_{i=1}^{\text{number of nodes}}$. Since the body is discretized into n disjoint elements, the integral in (3.15) may be separated using the spatial additivity of integration into n terms, one for each element

$$\delta \pi = \sum_{m=1}^n \delta \pi_m = 0. \quad (3.17)$$

The contribution from each element is

$$\delta\pi_m = \int_{v_m} \rho \ddot{x}_i \delta x_i dv + \int_{v_m} \sigma_{ij} \delta x_{i,j} dv - \int_{v_m} \rho f_i \delta x_i dv - \int_{\partial b_1 \cap \partial v_m} t_i \delta x_i ds. \quad (3.18)$$

Assembling the element contributions back into a system of equations leads to

$$\sum_{m=1}^n \left\{ \int_{v_m} \rho \ddot{x}_i (\mathbf{e}_i \otimes \mathbf{v}^m) dv + \int_{v_m} \sigma_{ij}^m (\mathbf{e}_i \otimes \mathbf{v}_{,j}^m) dv - \int_{v_m} \rho f_i (\mathbf{e}_i \otimes \mathbf{v}^m) dv - \int_{\partial b_1 \cap \partial v_m} t_i (\mathbf{e}_i \otimes \mathbf{v}^m) ds \right\} = 0. \quad (3.19)$$

In which

$$\mathbf{v}^m = \sum_{i=1}^k N_i \mathbf{e}'_{n_m(i)} \quad (3.20)$$

where $n_m(i)$ is the global node number.

Applying the approximation scheme of [Equation \(3.16\)](#) to the dependent variables and substituting into [Equation \(3.19\)](#) yields

$$\sum_{m=1}^n \left\{ \int_{v_m} \rho \mathbf{N}_m^T \mathbf{N}_m \mathbf{a} dv + \int_{v_m} \mathbf{B}_m^T \boldsymbol{\sigma} dv - \int_{v_m} \rho \mathbf{N}_m^T \mathbf{b} dv - \int_{\partial b_1} \mathbf{N}_m^T \mathbf{t} ds \right\} = 0 \quad (3.21)$$

where \mathbf{N} is an interpolation matrix; $\boldsymbol{\sigma}$ is the stress vector

$$\boldsymbol{\sigma}^T = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}); \quad (3.22)$$

\mathbf{B} is the strain-displacement matrix; \mathbf{a} is the nodal acceleration vector

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{bmatrix} = \mathbf{N} \begin{bmatrix} a_x^1 \\ a_x^2 \\ \vdots \\ a_y^k \\ a_z^k \end{bmatrix} = \mathbf{N} \mathbf{a}; \quad (3.23)$$

\mathbf{b} is the body force load vector

$$\mathbf{b} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}; \quad (3.24)$$

and \mathbf{t} is the applied traction load

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \quad (3.25)$$

4

Solid Elements

For a mesh of 8-node hexahedron solid elements, Equation ((3.16)) becomes:

$$x_i(X_\alpha, t) = x_i(X_\alpha(\xi, \eta, \zeta), t) = \sum_{j=1}^8 \phi_j(\xi, \eta, \zeta) x_i^j(t). \quad (4.1)$$

The shape function ϕ_j is defined for the 8-node hexahedron as

$$\phi_j = \frac{1}{8} (1 + \xi \xi_j) (1 + \eta \eta_j) (1 + \zeta \zeta_j), \quad (4.2)$$

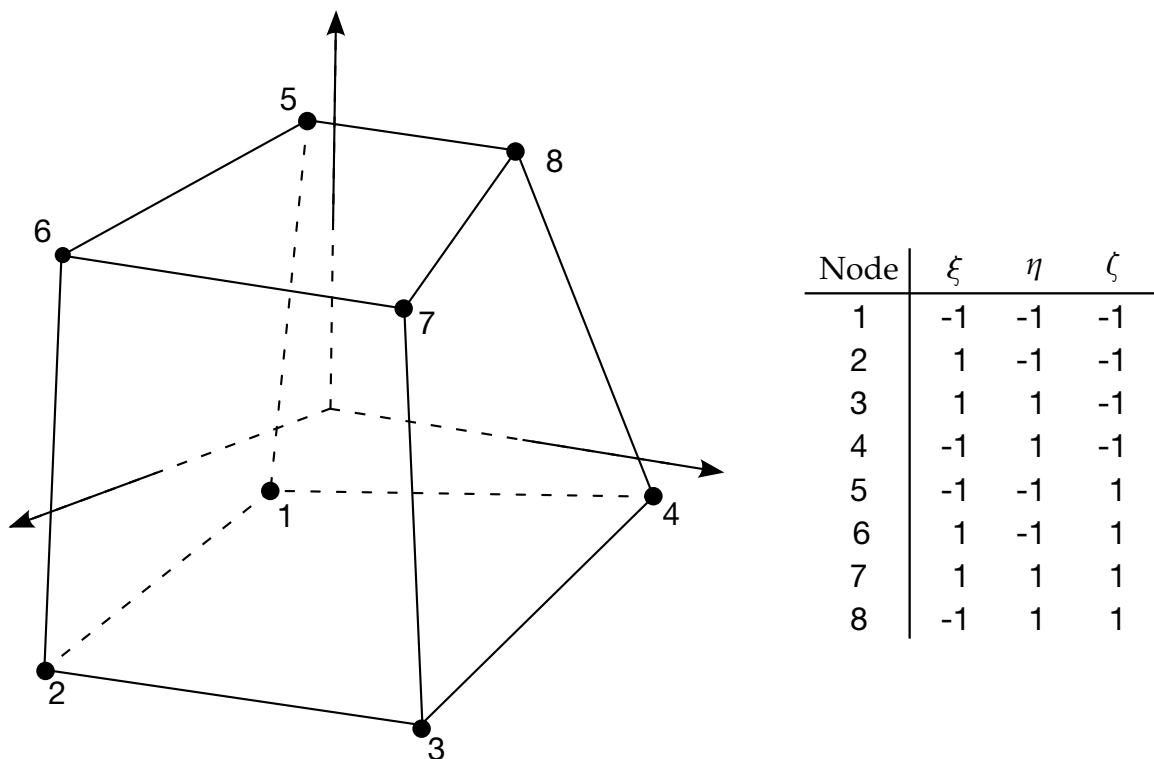


Figure 4.1. Eight node solid hexahedron element.

where ξ_j, η_j, ζ_j take on their nodal values of $(\pm 1, \pm 1, \pm 1)$ and x_i^j is the nodal coordinate of the j^{th} node in the i^{th} direction (see [Figure 4.1](#)).

For a solid element, \mathbf{N} is the 3×24 rectangular interpolation matrix given by

$$\mathbf{N}(\xi, \eta, \zeta) = \begin{bmatrix} \phi_1 & 0 & 0 & \phi_2 & 0 & \dots & 0 & 0 \\ 0 & \phi_1 & 0 & 0 & \phi_2 & \dots & \phi_8 & 0 \\ 0 & 0 & \phi_1 & 0 & 0 & \dots & 0 & \phi_8 \end{bmatrix}, \quad (4.3)$$

σ is the stress vector

$$\sigma^T = (\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sigma_{xy} \quad \sigma_{yz} \quad \sigma_{zx}). \quad (4.4)$$

\mathbf{B} is the 6×24 strain-displacement matrix

$$\mathbf{B} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \mathbf{N}. \quad (4.5)$$

To achieve a diagonal mass matrix, the rows are summed, giving the k^{th} diagonal term as:

$$m_{kk} = \int_v \rho \phi_k \sum_{i=1}^8 \phi_i dv = \int_v \rho \phi_k dv, \quad (4.6)$$

because the basis functions sum to unity.

Terms in the strain-displacement matrix are readily calculated. Note that

$$\begin{aligned} \frac{\partial \phi_i}{\partial \xi} &= \frac{\partial \phi_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial \phi_i}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial \phi_i}{\partial z} \frac{\partial z}{\partial \xi}, \\ \frac{\partial \phi_i}{\partial \eta} &= \frac{\partial \phi_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial \phi_i}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial \phi_i}{\partial z} \frac{\partial z}{\partial \eta}, \\ \frac{\partial \phi_i}{\partial \zeta} &= \frac{\partial \phi_i}{\partial x} \frac{\partial x}{\partial \zeta} + \frac{\partial \phi_i}{\partial y} \frac{\partial y}{\partial \zeta} + \frac{\partial \phi_i}{\partial z} \frac{\partial z}{\partial \zeta}, \end{aligned} \quad (4.7)$$

which can be rewritten as

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix}. \quad (4.8)$$

Inverting the Jacobian matrix, \mathbf{J} , we can solve for the desired terms

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial x} \\ \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} \\ \frac{\partial \phi_i}{\partial \eta} \\ \frac{\partial \phi_i}{\partial \zeta} \end{bmatrix}. \quad (4.9)$$

4.1 Volume Integration

Volume integration is carried out with Gaussian quadrature. If g is some function defined over the volume, and n is the number of integration points, then

$$\int_v g dv = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 g |\mathbf{J}| d\xi d\eta d\zeta, \quad (4.10)$$

is approximated by

$$\sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n g_{jkl} |\mathbf{J}_{jkl}| w_j w_k w_l, \quad (4.11)$$

where w_j, w_k, w_l are the weighting factors,

$$g_{jkl} = g(\xi_j, \eta_k, \zeta_l), \quad (4.12)$$

and $|\mathbf{J}|$ is the determinant of the Jacobian matrix. For one-point quadrature

$$\begin{aligned} n &= 1, \\ w_i &= w_j = w_k = 2, \\ \xi_1 &= \eta_1 = \zeta_1 = 0, \end{aligned} \quad (4.13)$$

and we can write

$$\int g dv = 8g(0,0,0) |\mathbf{J}(0,0,0)|. \quad (4.14)$$

Note that $8|\mathbf{J}(0,0,0)|$ approximates the element volume.

	DYNA3D	Flanagan Belytschko	Wilkins FDM
Strain displacement matrix	94	357	843
Strain rates	87	156	
Force	117	195	270
Subtotal	298	708	1113
Hourglass control	130	620	680
<i>Total</i>	428	1328	1793

Table 4.1. Operation counts for a constant stress hexahedron (includes adds, subtracts, multiplies, and divides in major subroutines, and is independent of vectorization). Material subroutines will add as little as 60 operations for the bilinear elastic-plastic routine to ten times as much for multi-surface plasticity and reactive flow models. Unvectorized material models will increase that share of the cost by a factor of four or more.

Perhaps the biggest advantage of one-point integration is the substantial savings in computer time. An anti-symmetry property of the strain matrix

$$\begin{aligned} \frac{\partial \phi_1}{\partial x_i} &= -\frac{\partial \phi_7}{\partial x_i}, & \frac{\partial \phi_3}{\partial x_i} &= -\frac{\partial \phi_5}{\partial x_i}, \\ \frac{\partial \phi_2}{\partial x_i} &= -\frac{\partial \phi_8}{\partial x_i}, & \frac{\partial \phi_4}{\partial x_i} &= -\frac{\partial \phi_6}{\partial x_i}, \end{aligned} \quad (4.15)$$

at $\xi = \eta = \zeta = 0$ reduces the amount of effort required to compute this matrix by more than 25 times over an 8-point integration. This cost savings extends to strain and element nodal force calculations where the number of multiplies is reduced by a factor of 16. Because only one constitutive evaluation is needed, the time spent determining stresses is reduced by a factor of 8. Operation counts for the constant stress hexahedron are given in Table 4.1. Included are counts for the Flanagan and Belytschko [1981] hexahedron and the hexahedron used by Wilkins [1974] in his integral finite difference method, which was also implemented [Hallquist 1979].

8-point integration has another disadvantage in addition to cost. Fully integrated elements used in the solution of plasticity problems and other problems where the Poisson's ratio approaches 0.5 lock up in the constant volume bending modes. To preclude locking, an average pressure must be used over the elements; consequently, the zero energy modes are resisted by the deviatoric stresses. If the deviatoric stresses are insignificant relative to the pressure, or, even worse if material failure causes loss of this stress state component, hourglassing will still occur, but with no means of resisting it. Sometimes, however, the cost of the fully integrated element may be justified by increased reliability and, if used sparingly, may actually increase the overall speed.

4.2 Solid Element 2

Solid element 2 is a selective reduced (S/R) integrated element that in general is regarded as too stiff. In particular this is the case when the elements exhibit poor aspect ratio, i.e., when one element dimension is significantly smaller than the other(s). This occurs for instance when modelling thin walled structures and the time for solving the problem prevents using a sufficient number of elements for maintaining close to cubic elements throughout the structure. The reason for the locking phenomenon is that the element is not able to represent pure bending modes without introducing transverse shear strains, and this may be bad enough to lock the element to a great extent. In an attempt to solve this transverse shear locking problem, two new fully integrated solid elements are introduced and documented herein that may become of practical use for these types of applications.

4.2.1 Brief summary of solid element 2

Let x_{Ii} represent the nodal coordinate of dimension i and node I , and likewise v_{Ii} its velocity. Furthermore denote

$$N_I(\tilde{\zeta}_1, \tilde{\zeta}_2, \tilde{\zeta}_3) = \frac{1}{8} (1 + \tilde{\zeta}_1^I \tilde{\zeta}_1 + \tilde{\zeta}_2^I \tilde{\zeta}_2 + \tilde{\zeta}_3^I \tilde{\zeta}_3 + \tilde{\zeta}_{12}^I \tilde{\zeta}_1 \tilde{\zeta}_2 + \tilde{\zeta}_{13}^I \tilde{\zeta}_1 \tilde{\zeta}_3 + \tilde{\zeta}_{23}^I \tilde{\zeta}_2 \tilde{\zeta}_3 + \tilde{\zeta}_{123}^I \tilde{\zeta}_1 \tilde{\zeta}_2 \tilde{\zeta}_3), \quad (4.2.16)$$

the shape functions for the standard isoparametric domain where

$$\begin{aligned} \tilde{\zeta}_1^* &= [-1 & 1 & 1 & -1 & -1 & 1 & 1 & -1], \\ \tilde{\zeta}_2^* &= [-1 & -1 & 1 & 1 & -1 & -1 & 1 & 1], \\ \tilde{\zeta}_3^* &= [-1 & -1 & -1 & -1 & 1 & 1 & 1 & 1], \\ \tilde{\zeta}_{12}^* &= [1 & -1 & 1 & -1 & 1 & -1 & 1 & -1], \\ \tilde{\zeta}_{13}^* &= [1 & -1 & -1 & 1 & 1 & -1 & 1 & -1], \\ \tilde{\zeta}_{23}^* &= [1 & 1 & -1 & -1 & -1 & -1 & 1 & 1], \\ \tilde{\zeta}_{123}^* &= [-1 & 1 & -1 & 1 & 1 & -1 & 1 & -1], \end{aligned} \quad (4.2.17)$$

and let furthermore

$$\begin{aligned} \tilde{\zeta}_{21}^I &= \tilde{\zeta}_{12}^I, \\ \tilde{\zeta}_{32}^I &= \tilde{\zeta}_{23}^I, \\ \tilde{\zeta}_{31}^I &= \tilde{\zeta}_{13}^I. \end{aligned} \quad (4.2.18)$$

The isoparametric representation of the coordinates of a material point in the element is then given as (where the dependence on $\tilde{\zeta}_1, \tilde{\zeta}_2, \tilde{\zeta}_3$ is suppressed for brevity)

$$x_i = x_{Ii} N_I, \quad (4.2.19)$$

and its associated jacobian matrix is

$$J_{ij} = \frac{\partial x_i}{\partial \tilde{\zeta}_j} = x_{Ii} \frac{1}{8} (\tilde{\zeta}_j^I + \tilde{\zeta}_{jk}^I \tilde{\zeta}_k + \tilde{\zeta}_{jl}^I \tilde{\zeta}_l + \tilde{\zeta}_{123}^I \tilde{\zeta}_k \tilde{\zeta}_l), \quad (4.2.20)$$

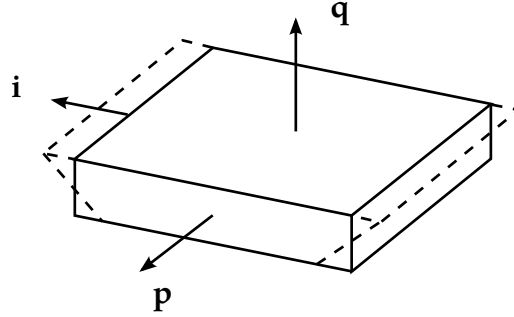


Figure 4.2.2. Bending mode for a fully integrated brick.

where $k = 1 + \text{mod}(j, 3)$ and $l = 1 + \text{mod}(j + 1, 3)$. For future reference let

$$J_{ij}^0 = x_{Ii}(0) \frac{1}{8} \tilde{\zeta}_j^I, \quad (4.2.21)$$

be the jacobian evaluated in the element center and in the beginning of the simulation (i.e., at time zero). The velocity gradient computed directly from the shape functions and velocity components is

$$L_{ij} = \frac{\partial v_i}{\partial x_j} = \dot{J}_{ik} J_{kj}^{-1} = B_{ijlk} v_{lk}, \quad (4.2.22)$$

where

$$B_{ijlk} = \frac{\partial N_I}{\partial \tilde{\zeta}_l} J_{lj}^{-1} \delta_{ik}, \quad (4.2.23)$$

is the gradient-displacement matrix and represents the element except for the alleviation of volumetric locking. To do just that, let B_{ijlk}^0 be defined by

$$\dot{\bar{J}}_{ik} \bar{J}_{kj}^{-1} = B_{ijlk}^0 v_{lk}, \quad (4.2.24)$$

with \bar{J}_{ij} being the element averaged jacobian matrix, and construct the gradient-displacement matrix used for the element as

$$\bar{B}_{ijlk} = B_{ijlk} + \frac{1}{3} (B_{lllk}^0 - B_{lllk}) \delta_{ij}. \quad (4.2.25)$$

This is what is often called the B-bar method.

4.2.2 Transverse shear locking example

To get the idea of the modifications needed to alleviate transverse shear locking let's look at the parallelepiped of dimensions $l_1 \times l_2 \times l_3$ in [Figure 4.2.2](#). For this simple geometry the jacobian matrix is diagonal and the velocity gradient is expressed as

$$L_{ij} = \frac{2}{l_j} \dot{J}_{ij} = \frac{1}{4l_j} v_{Ii} (\tilde{\zeta}_j^I + \tilde{\zeta}_{jk}^I \tilde{\zeta}_k + \tilde{\zeta}_{jl}^I \tilde{\zeta}_l + \tilde{\zeta}_{123}^I \tilde{\zeta}_k \tilde{\zeta}_l), \quad (4.2.26)$$

where, again, $k = 1 + \text{mod}(j, 3)$ and $l = 1 + \text{mod}(j + 1, 3)$. Now let $i \neq p \neq q \neq i$, then a pure bending mode in the plane with normal in direction q and about axis p is represented by

$$\begin{aligned} v_{Li} &= \tilde{\zeta}_{iq}^I, \\ v_{Ip} &= 0, \\ v_{Iq} &= 0, \end{aligned} \quad (4.2.27)$$

and thus, the velocity gradient is given as

$$\begin{aligned} L_{ij} &= \frac{1}{4l_j} (\tilde{\zeta}_{iq}^I \tilde{\zeta}_{jk}^I \tilde{\zeta}_k + \tilde{\zeta}_{iq}^I \tilde{\zeta}_{jl}^I \tilde{\zeta}_l), \\ L_{pj} &= 0, \\ L_{qj} &= 0, \end{aligned} \quad (4.2.28)$$

for $j = 1, 2, 3$. The nonzero expression above amounts to

$$\begin{aligned} L_{ii} &= \frac{1}{4l_i} \tilde{\zeta}_q, \\ L_{ip} &= 0, \\ L_{iq} &= \frac{1}{4l_q} \tilde{\zeta}_i. \end{aligned} \quad (4.2.29)$$

Notable here is that a pure bending mode gives arise to a transverse shear strain represented by the last expression in the above. Assuming that l_q is small compared to l_i this may actually lock the element.

4.2.3 Solid element -2

Given this insight the modifications in the expression of the jacobian matrix are as follows. Let

$$\kappa_{mn} = \min \left(1, \frac{\sqrt{J_{1m}^0 J_{1m}^0 + J_{2m}^0 J_{2m}^0 + J_{3m}^0 J_{3m}^0}}{\sqrt{J_{1n}^0 J_{1n}^0 + J_{2n}^0 J_{2n}^0 + J_{3n}^0 J_{3n}^0}} \right), \quad (4.2.30)$$

be the aspect ratio between dimensions m and n at time zero. The modified jacobian is written

$$\tilde{J}_{ij} = x_{Li} \frac{1}{8} (\tilde{\zeta}_j^I + \tilde{\zeta}_{jk}^I \tilde{\zeta}_{ki} + \tilde{\zeta}_{jl}^I \tilde{\zeta}_{li} + \tilde{\zeta}_{123}^I \tilde{\zeta}_{ki} \tilde{\zeta}_{li}), \quad (4.2.31)$$

where

$$\tilde{\zeta}_{ki} = \begin{cases} \tilde{\zeta}_k \kappa_{jk} & i \neq j \\ \tilde{\zeta}_k & \text{otherwise} \end{cases} \quad (4.2.32)$$

and

$$\tilde{\zeta}_{li} = \begin{cases} \tilde{\zeta}_l \kappa_{jl} & i \neq j \\ \tilde{\zeta}_l & \text{otherwise} \end{cases} \quad (4.2.33)$$

The velocity gradient is now given as

$$L_{ij} = \dot{\tilde{J}}_{ik} \tilde{J}_{kj}^{-1} = \tilde{B}_{ijlk} v_{Ik}, \quad (4.2.34)$$

where \tilde{B}_{ijlk} is the gradient-displacement matrix used for solid element type -2 in LS-DYNA. The B-bar method is used to prevent volumetric locking.

4.2.4 Transverse shear locking example revisited

Once again let's look at the parallelepiped of dimensions $l_1 \times l_2 \times l_3$. The jacobian matrix is still diagonal and the velocity gradient is with the new element formulation expressed as

$$L_{ij} = \frac{2}{l_j} \dot{\tilde{J}}_{ij} = \frac{1}{4l_j} v_{li} (\tilde{\zeta}_j^I + \tilde{\zeta}_{jk}^I \tilde{\zeta}_{ki} + \tilde{\zeta}_{jl}^I \tilde{\zeta}_{li} + \tilde{\zeta}_{123}^I \tilde{\zeta}_{ki} \tilde{\zeta}_{li}), \quad (4.2.35)$$

where, again, $k = 1 + \text{mod}(j, 3)$ and $l = 1 + \text{mod}(j + 1, 3)$. The velocity gradient for a pure bending mode is now given as

$$L_{ij} = \frac{1}{4l_j} (\tilde{\zeta}_{iq}^I \tilde{\zeta}_{jk}^I \tilde{\zeta}_{ki} + \tilde{\zeta}_{iq}^I \tilde{\zeta}_{jl}^I \tilde{\zeta}_{li}), \quad (4.2.36)$$

which amounts to (for the potential nonzero elements)

$$\begin{aligned} L_{ii} &= \frac{1}{4l_i} \tilde{\zeta}_q, \\ L_{ip} &= 0, \\ L_{iq} &= \frac{1}{4l_q} \tilde{\zeta}_i \kappa_{qi}. \end{aligned} \quad (4.2.37)$$

If we assume that this is the geometry in the beginning of the simulation and that l_q is smaller than l_i the transverse shear strain can be expressed as

$$L_{iq} = \frac{1}{4l_i} \tilde{\zeta}_i, \quad (4.2.38)$$

meaning that the transverse shear energy is not affected by poor aspect ratios, i.e., the transverse shear strain does not grow with decreasing l_q .

4.2.5 Solid element -1

Working out the details in the expression of the gradient-displacement matrix for solid element type -2 reveals that this matrix is dense, i.e., there are 216 nonzero elements in this matrix that needs to be processed compared to 72 for the standard solid element type 2. A slight modification of the jacobian matrix will substantially reduce the computational expense for this element. Simply substitute the expressions for $\tilde{\zeta}_{ki}$ and $\tilde{\zeta}_{li}$ by

$$\tilde{\zeta}_{ki} = \tilde{\zeta}_k \kappa_{jk}, \quad (4.2.39)$$

and

$$\tilde{\zeta}_{li} = \tilde{\zeta}_l \kappa_{jl}. \quad (4.2.40)$$

This will lead to a stiffness reduction for certain modes, in particular the out-of-plane hourglass mode as can be seen by once again looking at the transverse shear locking example. The velocity gradient for pure bending is now

$$\begin{aligned} L_{ii} &= \frac{1}{4l_i} \tilde{\xi}_q \kappa_{iq}, \\ L_{ip} &= 0, \\ L_{iq} &= \frac{1}{4l_q} \tilde{\xi}_i \kappa_{qi}, \end{aligned} \quad (4.2.41)$$

and if it turns out that l_i is smaller than l_q , then this results in

$$L_{ii} = \frac{1}{4l_q} \tilde{\xi}_q. \quad (4.2.42)$$

That is, if i represents the direction of the thinnest dimension, its corresponding bending strain is inadequately reduced.

4.2.6 Example

A plate of dimensions $10 \times 5 \times 1 \text{ mm}^3$ is clamped at one end and subjected to a 1 Nm torque at the other end. The Young's modulus is 210 GPa and the analytical solution for the end tip deflection is 0.57143 mm. In order to study the mesh convergence for the three fully integrated bricks the plate is discretized into $2 \times 1 \times 1$, $4 \times 2 \times 2$, $8 \times 4 \times 4$, $16 \times 8 \times 8$ and finally $32 \times 16 \times 16$ elements, all elements having the same aspect ratio of 5×1 . The table below shows the results for the different fully integrated elements, and indicates an accuracy improvement for solid elements -1 and -2.

Discretization	Solid element type 2	Solid element type -2	Solid element type -1
2x1x1	0.0564 (90.1%)	0.6711 (17.4%)	0.6751 (18.1%)
4x2x2	0.1699 (70.3%)	0.5466 (4.3%)	0.5522 (3.4%)
8x4x4	0.3469 (39.3%)	0.5472 (4.2%)	0.5500 (3.8%)
16x8x8	0.4820 (15.7%)	0.5516 (3.5%)	0.5527 (3.3%)
32x16x16	0.5340 (6.6%)	0.5535 (3.1%)	0.5540 (3.1%)

4.3 Hourglass Control

The biggest disadvantage to one-point integration is the need to control the zero energy modes, which arise, called hourglassing modes. Undesirable hourglass modes tend to have periods that are typically much shorter than the periods of the structural response, and they are often observed to be oscillatory. However, hourglass modes that have periods that are comparable to the structural response periods may be a stable kinematic component of the global deformation modes and must be admissible. One way of resisting undesirable hourglassing is with a viscous damping or small elastic stiffness capable of stopping the formation of the anomalous modes but having a negligible affect on the stable global modes. Two of the early three-dimensional algorithms for controlling

the hourglass modes were developed by Kosloff and Frazier [1974] and Wilkins et al. [1974].

Since the hourglass deformation modes are orthogonal to the strain calculations, work done by the hourglass resistance is neglected in the energy equation. This may lead to a slight loss of energy; however, hourglass control is always recommended for the under integrated solid elements. The energy dissipated by the hourglass forces reacting against the formations of the hourglass modes is tracked and reported in the output files `matsum` and `glistat`.

It is easy to understand the reasons for the formation of the hourglass modes. Consider the following strain rate calculations for the 8-node solid element

$$\dot{\epsilon}_{ij} = \frac{1}{2} \left(\sum_{k=1}^8 \frac{\partial \phi_k}{\partial x_i} \dot{x}_j^k + \frac{\partial \phi_k}{\partial x_j} \dot{x}_i^k \right). \quad (4.43)$$

Whenever diagonally opposite nodes have identical velocities, i.e.,

$$\dot{x}_i^1 = \dot{x}_i^7, \quad \dot{x}_i^2 = \dot{x}_i^8, \quad \dot{x}_i^3 = \dot{x}_i^5, \quad \dot{x}_i^4 = \dot{x}_i^6, \quad (4.44)$$

the strain rates are identically zero:

$$\dot{\epsilon}_{ij} = 0, \quad (4.45)$$

due to the asymmetries in [Equations \(4.15\)](#). It is easy to prove the orthogonality of the hourglass shape vectors, which are listed in [Table 4](#) and shown in [Figure 4.3](#) with the derivatives of the shape functions:

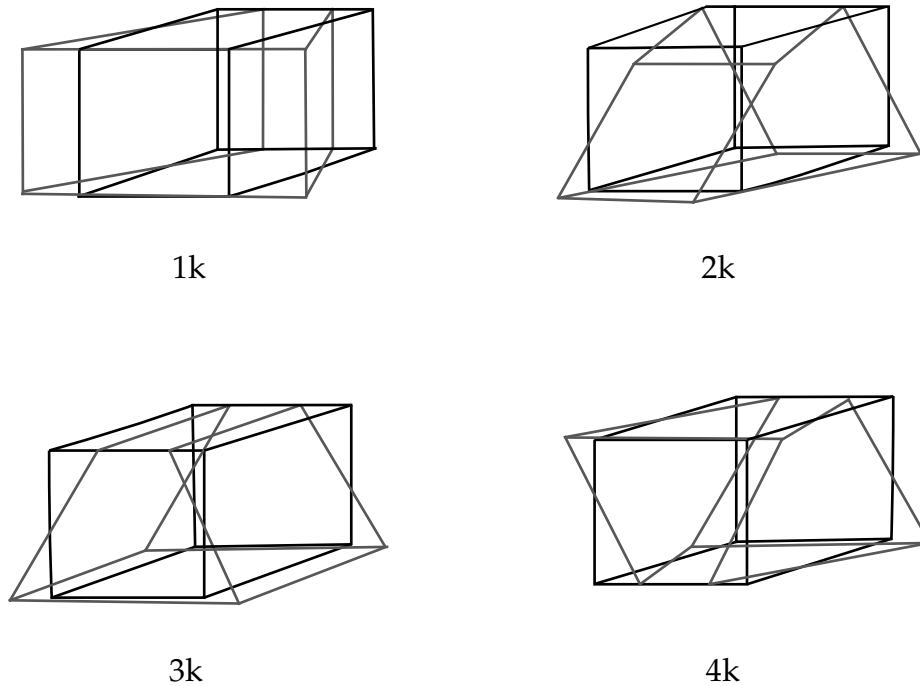


Figure 4.3. The hourglass modes of an eight-node element with one integration point are shown [Flanagan and Belytschko 1981]. A total of twelve modes exist.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
Γ_{j1}	1	1	1	1
Γ_{j2}	1	-1	-1	-1
Γ_{j3}	-1	-1	1	1
Γ_{j4}	-1	1	-1	-1
Γ_{j5}	-1	-1	1	-1
Γ_{j6}	-1	1	-1	1
Γ_{j7}	1	1	1	-1
Γ_{j8}	1	-1	-1	1

Table 4. Hourglass base vectors.

$$\sum_{k=1}^8 \frac{\partial \phi_k}{\partial x_i} \Gamma_{\alpha k} = 0, \quad i = 1, 2, 3, \quad \alpha = 1, 2, 3, 4. \quad (4.46)$$

The product of the base vectors with the nodal velocities is zero when the element velocity field has no hourglass component,

$$h_{i\alpha} = \sum_{k=1}^8 \dot{x}_i^k \Gamma_{\alpha k} = 0. \quad (4.47)$$

are nonzero if hourglass modes are present. The 12 hourglass-resisting force vectors, $f_{i\alpha}^k$ are

$$f_{i\alpha}^k = a_h h_{i\alpha} \Gamma_{\alpha k}, \quad (4.48)$$

where

$$a_h = Q_{\text{HG}} \rho v_e^{2/3} \frac{c}{4}, \quad (4.49)$$

in which v_e is the element volume, c is the material sound speed, and Q_{HG} is a user-defined constant usually set to a value between .05 and .15. Equation (1.21) is hourglass control type 1 in the LS-DYNA User's Manual.

A shortcoming of hourglass control type 1 is that the hourglass resisting forces of Equation (1.21) are not orthogonal to linear velocity field when elements are not in the shape of parallelepipeds. As a consequence, such elements can generate hourglass energy with a constant strain field or rigid body rotation. Flanagan and Belytschko [1981] developed an hourglass control that is orthogonal to all modes except the zero energy hourglass modes. Instead of resisting components of the bilinear velocity field that are orthogonal to the strain calculation, Flanagan and Belytschko resist components of the velocity field that are not part of a fully linear field. They call this field, defined below, the hourglass velocity field

$$\dot{x}_i^{k\text{HG}} = \dot{x}_i^k - \dot{x}_i^{k\text{LIN}}, \quad (4.50)$$

where

$$\dot{x}_i^{k\text{LIN}} = \dot{x}_i + \dot{x}_{i,j}(x_j^k - \bar{x}_j), \quad (4.51)$$

and

$$\begin{aligned} \bar{x}_i &= \frac{1}{8} \sum_{k=1}^8 x_i^k, \\ \dot{\bar{x}}_i &= \frac{1}{8} \sum_{k=1}^8 \dot{x}_i^k. \end{aligned} \quad (4.52)$$

Flanagan and Belytschko construct geometry-dependent hourglass shape vectors that are orthogonal to the fully linear velocity field and the rigid body field. With these vectors they resist the hourglass velocity deformations. Defining hourglass shape vectors in terms of the base vectors as

$$\gamma_{\alpha k} = \Gamma_{\alpha k} - \phi_{k,i} \sum_{n=1}^8 x_i^n \Gamma_{\alpha n}, \quad (4.53)$$

the analogue for (4.47) is,

$$g_{i\alpha} = \sum_{k=1}^8 \dot{x}_i^k \gamma_{\alpha k} = 0, \quad (4.54)$$

with the 12 resisting force vectors being

$$f_{i\alpha}^k = a_h g_{i\alpha} \gamma_{\alpha k}, \quad (4.55)$$

where a_h is a constant given in Equation (4.48). Equation (1.28) corresponds to hourglass control type 2 in the LS-DYNA User's Manual. The γ terms used of equation of Equation (1.26) are used not only type hourglass control type 2, but are the basis for all solid element hourglass control except for form 1.

A cost comparison in Table 4.1 shows that the default type 1 hourglass viscosity requires approximately 130 adds or multiplies per hexahedron, compared to 620 and 680 for the algorithms of Flanagan-Belytschko and Wilkins. Therefore, for a very regular mesh, type 1 hourglass control may provide a faster, sufficiently accurate solution, but in general, any of the other hourglass options which are all based on the $\gamma_{\alpha k}$ terms of Equation (1.26) will be a better choice.

Type 3 hourglass control is identical to type 2, except that the shape function derivatives in Eq. (1.26) are evaluated at the centroid of the element rather than at the origin of the referential coordinate system. With this method, Equation (1.14) produces the exact element volume. However, the anti-symmetry property of Equation (1.15) is not true, so there is some increased number of computations.

The remaining hourglass control types calculated hourglass forces proportional to total hourglass deformation rather than hourglass viscosity. A stiffness form of hourglass control allows elements to spring back and will absorb less energy than the viscous forms.

Types 4 and 5 hourglass control are similar to types 2 and 3, except that they evaluate hourglass stiffness rather than viscosity. The hourglass rates of equation (1.27) are multiplied by the solution time step to produce increments of hourglass deformation. The hourglass stiffness is scaled by the element's maximum frequency so that stability can be maintained as long as the hourglass scale factor, a_h , is sufficiently small.

Type 6 hourglass control improves on type 5 by scaling the stiffness such that the hourglass forces match those generated by a fully integrated element control by doing closed form integration over the element volume scaling the hourglass stiffness by matching the stabilization for the 3D hexahedral element is available for both implicit and explicit solutions. Based on material properties and element geometry, this stiffness type stabilization is developed by an assumed strain method [Belytschko and Bindeman 1993] such that the element does not lock with nearly incompressible material. When the user defined hourglass constant a_h is set to 1.0, accurate coarse mesh bending stiffness is obtained for elastic material. For nonlinear material, a smaller value of a_h is suggested and the default value is set to 0.1. In the implicit form, the assumed strain stabilization matrix is:

$$\mathbf{K}^{\text{stab}} = 2\mu a_h \begin{bmatrix} \mathbf{k}_{11} & \mathbf{k}_{12} & \mathbf{k}_{13} \\ \mathbf{k}_{21} & \mathbf{k}_{22} & \mathbf{k}_{23} \\ \mathbf{k}_{31} & \mathbf{k}_{32} & \mathbf{k}_{33} \end{bmatrix}, \quad (4.56)$$

where the 8×8 submatrices are calculated by:

$$\begin{aligned} \mathbf{k}_{ii} &\equiv H_{ii} \left[\left(\frac{1}{1-v} \right) (\gamma_j \gamma_j^T + \gamma_k \gamma_k^T) + \left(\frac{1+v}{3} \right) \gamma_4 \gamma_4^T \right] + \frac{1}{2} (H_{jj} + H_{kk}) \gamma_i \gamma_i^T, \\ \mathbf{k}_{ij} &\equiv H_{ij} \left[\left(\frac{v}{1-v} \right) \gamma_j \gamma_i^T + \frac{1}{2} \gamma_i \gamma_j^T \right] \end{aligned} \quad (4.57)$$

with,

$$\begin{aligned} H_{ii} &\equiv \int_v (h_{j,i})^2 dv = \int_v (h_{k,i})^2 dv = 3 \int_v (h_{4,i})^2 dv, \\ H_{ij} &\equiv \int_v h_{i,j} h_{j,i} dv, \end{aligned} \quad (4.58)$$

where,

$$h_1 = \zeta \eta \quad h_2 = \eta \zeta \quad h_3 = \zeta \xi \quad h_4 = \xi \eta \zeta, \quad (4.59)$$

Subscripts i, j , and k are permuted as in [Table 44.2](#). A comma indicates a derivative with respect to the spatial variable that follows. The hourglass vectors, γ_α are defined by [equation \(4.53\)](#). The stiffness matrix is evaluated in a corotational coordinate system that is aligned with the referential axis of the element. The use of a corotational system allows direct evaluation of integrals in [equations \(4.58\)](#) by simplified equations that produce a more accurate element than full integration.

$$H_{ii} = \frac{1}{3} \frac{(\Lambda_j^T \mathbf{x}_j)(\Lambda_k^T \mathbf{x}_k)}{(\Lambda_i^T \mathbf{x}_i)} \quad i \neq j \neq k, \quad (4.60)$$

<i>i</i>	<i>j</i>	<i>k</i>
1	2	3
1	3	2
2	3	1
2	1	3
3	1	2
3	2	1

Table 44.2. Permutations of *i*, *j*, and *k*.

$$H_{ij} = \frac{1}{3} (\Lambda_k^T \mathbf{x}_k) \quad i \neq j \neq k, \quad (4.61)$$

Λ_i are 8×1 matrices of the referential coordinates of the nodes as given in [Figure 4.1](#), and \mathbf{x}_i are 8×1 matrices of the nodal coordinates in the corotational system. For each material type, a Poisson's ratio, ν , and an effective shear modulus, μ , is needed.

In the explicit form, the 12 hourglass force stabilization vectors are

$$\mathbf{f}_i^{\text{stab}} = \sum_{\alpha=1}^4 a_h g_{i\alpha} \gamma_{\alpha}^T, \quad (4.62)$$

where the 12 generalized stresses are calculated incrementally by

$$g_{i\alpha}^n = g_{i\alpha}^{n-1} + \Delta t \dot{g}_{i\alpha}^{n-\frac{1}{2}}, \quad (4.63)$$

and

$$\begin{aligned} \dot{g}_{ii} &= \mu [(H_{jj} + H_{kk}) \dot{q}_{ii} + H_{ij} \dot{q}_{jj} + H_{ik} \dot{q}_{kk}], \\ \dot{g}_{ij} &= 2\mu \left[\frac{1}{1-\nu} H_{ii} \dot{q}_{ij} + \nu H_{kk} \dot{q}_{ii} \right] \\ \dot{g}_{i4} &= 2\mu \left(\frac{1+\nu}{3} \right) H_{ii} \dot{q}_{i4} \end{aligned} \quad (4.64)$$

where,

$$\dot{q}_{i\alpha} = (\gamma_{\alpha}^T \dot{\mathbf{x}}_i). \quad (4.65)$$

Subscripts *i*, *j*, and *k* are permuted as per [Table 44.2](#). As with the implicit form, calculations are done in a corotational coordinate system in order to use the simplified [equations \(4.60\) and \(4.61\)](#).

Type 7 hourglass control is very similar to type 6 hourglass control but with one significant difference. As seen in Equation (1.36), type 6 obtains the current value of the generalized stress from the previous value and the current increment. The incremental method is nearly always sufficiently accurate, but it is possible for hourglass modes to fail to spring back to the initial element geometry since the hourglass stiffness varies as the *H* terms given by Equations (1.33) and (1.34) are recalculated in the deformed configuration each cycle. Type 7 hourglass control eliminates this possible error by calculating the total

hourglass deformation in each cycle. For type 7 hourglass control, Equations (1.37) are rewritten using g and q in place of \dot{g} and \dot{q} , and Equation (1.38) is replaced by (1.39).

$$q_{i\alpha} = (\gamma_{\alpha}^T \mathbf{x}_i) - (\gamma_{0\alpha}^T \mathbf{x}_{0i}). \quad (4.39)$$

In Equation (1.39), \mathbf{x}_{0i} and $\gamma_{0\alpha}$ are evaluated using the initial, undeformed nodal coordinate values. Type 7 hourglass control is considerably slower than type 6, so it is not generally recommended, but may be useful when the solution involves at least several cycles of loading and unloading that involve large element deformation of elastic or hyperelastic material.

Both type 6 and 7 hourglass control are stiffness type methods, but may have viscosity added through the VDC parameter on the *HOURLASS card. The VDC parameter scales the added viscosity, and VDC = 1.0 corresponds approximately to critical damping. The primary motivation for damping is to reduce high frequency oscillations. A small percentage of critical damping should be sufficient for this, but it is also possible to add supercritical damping along with a small value of QM to simulate a very viscous material that springs back slowly.

4.4 Puso Hourglass Control

Regarding the solid elements in LS-DYNA, the fully integrated brick uses selective-reduced integration, which is known to alleviate volumetric locking but not shear locking for elements with poor aspect ratio. The enhanced assumed strain methods have been the most successful at providing coarse mesh accuracy for general non-linear material models. In short, these elements tend to sacrifice computational efficiency for accuracy and are hence of little interest in explicit analysis. Puso [2000] developed an enhanced assumed strain element that combines coarse mesh accuracy with computational efficiency. It is formulated as a single point integrated brick with an enhanced assumed strain physical stabilization. In this project, we have implemented this element in LS-DYNA and made comparisons with the assumed strain element developed by Belytschko and Bindeman [1993] to see whether it brings anything new to the existing LS-DYNA element library. This is hourglass control type 9.

The element formulation is that of Puso [2000], and is essentially the mean strain hexahedral element by Flanagan and Belytschko [1981] in which the perturbation hourglass control is substituted for an enhanced assumed strain stabilization force. Given the matrices

$$\mathbf{S} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{\Xi} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix}, \quad (4.66)$$

we can define the vector of shape functions as

$$\mathbf{N}(\xi) = \frac{1}{8} [\mathbf{s} + \mathbf{\Xi}\xi + \mathbf{H}\mathbf{h}(\xi)], \quad (4.67)$$

where

$$\xi = \begin{bmatrix} \tilde{\xi} \\ \eta \\ \zeta \end{bmatrix}, \quad \mathbf{h}(\xi) = \begin{bmatrix} \eta\zeta \\ \tilde{\xi}\zeta \\ \tilde{\xi}\eta \\ \tilde{\xi}\eta\zeta \end{bmatrix}. \quad (4.68)$$

The position vector

$$\mathbf{x}(\xi) = \begin{bmatrix} x(\xi) \\ y(\xi) \\ z(\xi) \end{bmatrix}, \quad (4.69)$$

is for isoparametric finite elements given as

$$\mathbf{x}(\xi) = \mathbf{X}^T \mathbf{N}(\xi), \quad (4.70)$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_5 & y_5 & z_5 \\ x_6 & y_6 & z_6 \\ x_7 & y_7 & z_7 \\ x_8 & y_8 & z_8 \end{bmatrix}, \quad (4.71)$$

is the matrix of nodal coordinates. The Jacobian matrix maps the isoparametric domain to the physical domain as

$$\mathbf{J}(\xi) = \frac{\partial \mathbf{x}(\xi)}{\partial \xi}, \quad (4.72)$$

and we find the Jacobian matrix at the element centroid to be

$$\mathbf{J}_0 = \mathbf{J}(0) = \frac{1}{8} \mathbf{X}^T \mathbf{\Xi}. \quad (4.73)$$

We may use this to rewrite the vector of shape functions partially in terms of the position vector as

$$\mathbf{N}(\xi) = \mathbf{b}_0 + \mathbf{B}_0 \mathbf{x} + \Gamma \mathbf{h}(\xi), \quad (4.74)$$

Where

$$\mathbf{b}_0 = \frac{1}{8} \{ \mathbf{I} - \mathbf{B}_0 \mathbf{X}^T \} \mathbf{s}, \quad (4.75a)$$

$$\Gamma = \frac{1}{8} \{ \mathbf{I} - \mathbf{B}_0 \mathbf{X}^T \} \mathbf{H}, \quad (4.75b)$$

$$\mathbf{B}_0 = \frac{1}{8} \Xi \mathbf{J}_0^{-1}. \quad (4.75c)$$

The gradient-displacement matrix from this expression is given as

$$\mathbf{B}(\xi) = \mathbf{B}_0 + \mathbf{B}_s(\xi), \quad (4.76)$$

where

$$\mathbf{B}_s(\xi) = \Gamma \frac{\partial \mathbf{h}(\xi)}{\partial \xi} \mathbf{J}(\xi)^{-1}. \quad (4.77)$$

We have

$$\frac{\partial \mathbf{h}(\xi)}{\partial \xi} = \begin{bmatrix} 0 & \varsigma & \eta \\ \varsigma & 0 & \tilde{\varsigma} \\ \eta & \tilde{\varsigma} & 0 \\ \eta\varsigma & \tilde{\varsigma}\varsigma & \tilde{\varsigma}\eta \end{bmatrix}. \quad (4.78)$$

At this point we substitute the gradient-displacement matrix at the centroid of the element \mathbf{B}_0 with the mean gradient-displacement matrix $\bar{\mathbf{B}}$ defined as

$$\bar{\mathbf{B}} = \frac{1}{V_e} \int_e \mathbf{B}(\xi) dV_e, \quad (4.79)$$

where e refers to the element domain and V_e is the volume of the element, in all of the expressions above. That is

$$\bar{\Gamma} = \frac{1}{8} \{ \mathbf{I} - \bar{\mathbf{B}} \mathbf{X}^T \} \mathbf{H}, \quad (4.80)$$

and

$$\mathbf{B}(\xi) = \bar{\mathbf{B}} + \bar{\mathbf{B}}_s(\xi), \quad (4.81)$$

where

$$\bar{\mathbf{B}}_s(\xi) = \bar{\Gamma} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} \mathbf{J}(\xi). \quad (4.82)$$

Proceeding, we write the expression for the rate-of-deformation as

$$(4.83)$$

$$= \frac{1}{2} (\dot{\mathbf{X}}^T \bar{\mathbf{B}} + \bar{\mathbf{B}}^T \dot{\mathbf{X}}) + \frac{1}{2} \mathbf{J}(\xi)^{-T} \left\{ \mathbf{J}(\xi)^T \dot{\mathbf{X}}^T \bar{\mathbf{\Gamma}} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} + \left[\frac{\partial \mathbf{h}(\xi)}{\partial \xi} \right]^T \bar{\mathbf{\Gamma}}^T \dot{\mathbf{X}} \mathbf{J}(\xi) \right\} \mathbf{J}(\xi)^{-1},$$

where we substitute the occurrences of the jacobian matrix $\mathbf{J}(\xi)$ with the following expressions

$$\dot{\mathbf{e}} \approx \frac{1}{2} (\dot{\mathbf{X}}^T \bar{\mathbf{B}} + \bar{\mathbf{B}}^T \dot{\mathbf{X}}) + \frac{1}{2} \hat{\mathbf{J}}_0^{-T} \left\{ \mathbf{J}_0^T \dot{\mathbf{X}}^T \bar{\mathbf{\Gamma}} \frac{\partial \mathbf{h}(\xi)}{\partial \xi} + \left[\frac{\partial \mathbf{h}(\xi)}{\partial \xi} \right]^T \bar{\mathbf{\Gamma}}^T \dot{\mathbf{X}} \mathbf{J}_0 \right\} \hat{\mathbf{J}}_0^{-1}, \quad (4.84)$$

where

$$\hat{\mathbf{J}}_0 = \begin{bmatrix} \|\mathbf{j}_1\| & & \\ & \|\mathbf{j}_2\| & \\ & & \|\mathbf{j}_3\| \end{bmatrix}, \quad (4.85)$$

and \mathbf{j}_i is the i :th column in the matrix \mathbf{J}_0 . This last approximation is the key to the mesh distortion insensitivity that characterizes the element.

Changing to Voigt notation, we define the stabilization portion of the strain rate as

$$\dot{\mathbf{e}}_s = \hat{\mathbf{J}}_0^{-1} \tilde{\mathbf{B}}_s(\xi) \tilde{\mathbf{u}}, \quad (4.86)$$

where now

$$\mathbf{J}_0^{-1} = \begin{bmatrix} \|\mathbf{j}_1\|^{-2} & & & & & \\ & \|\mathbf{j}_2\|^{-2} & & & & \\ & & \|\mathbf{j}_3\|^{-2} & & & \\ & & & \|\mathbf{j}_1\|^{-1} \|\mathbf{j}_2\|^{-1} & & \\ & & & & \|\mathbf{j}_3\|^{-1} \|\mathbf{j}_2\|^{-1} & \\ & & & & & \|\mathbf{j}_1\|^{-1} \|\mathbf{j}_3\|^{-1} \end{bmatrix}, \quad (4.87)$$

$$\tilde{\mathbf{B}}_s(\xi) = \begin{bmatrix} \bar{\gamma}_2 \zeta + \bar{\gamma}_3 \eta + \bar{\gamma}_4 \zeta \eta & 0 & 0 \\ 0 & \bar{\gamma}_1 \zeta + \bar{\gamma}_3 \zeta + \bar{\gamma}_4 \zeta \zeta & 0 \\ 0 & 0 & \bar{\gamma}_1 \eta + \bar{\gamma}_2 \zeta + \bar{\gamma}_4 \zeta \eta \\ \bar{\gamma}_1 \zeta & \bar{\gamma}_2 \zeta & 0 \\ 0 & \bar{\gamma}_2 \zeta & \bar{\gamma}_3 \zeta \\ \bar{\gamma}_1 \eta & 0 & \bar{\gamma}_3 \eta \end{bmatrix}, \quad (4.88)$$

and $\tilde{\mathbf{u}}$ is the vector of nodal velocities transformed to the isoparametric system according to

$$\tilde{\mathbf{u}} = \mathfrak{J} \dot{\mathbf{u}}, \quad (4.89)$$

where \mathfrak{J} is the 24×24 matrix that transforms the 8 nodal velocity vectors to the isoparametric domain given by

$$\mathfrak{J} = \text{perm} \begin{bmatrix} \mathbf{J}_0^T & & \\ & \ddots & \\ & & \mathbf{J}_0^T \end{bmatrix}. \quad (4.90)$$

Moreover, $\bar{\gamma}_i$ is the i^{th} row of $\bar{\Gamma}^T$. We have deliberately neglected terms that cause parallelepiped finite elements to lock in shear.

To eliminate Poisson type locking in bending and volumetric locking, an enhanced isoparametric rate-of-strain field is introduced as

$$\dot{\epsilon}_e = \hat{\mathbf{J}}_0^{-1} \tilde{\mathbf{G}}(\xi) \dot{\alpha}, \quad (4.91)$$

with

$$\tilde{\mathbf{G}}(\xi) = \begin{bmatrix} \xi & 0 & 0 & \xi\eta & \eta\zeta & \zeta\xi \\ 0 & \eta & 0 & \xi\eta & \eta\zeta & \zeta\xi \\ 0 & 0 & \zeta & \xi\eta & \eta\zeta & \zeta\xi \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.92)$$

Hence, the stabilized strain field becomes

$$\dot{\epsilon}_s = \hat{\mathbf{J}}_0^{-1} [\tilde{\mathbf{B}}_s(\xi) \tilde{\mathbf{u}} + \tilde{\mathbf{G}}(\xi) \dot{\alpha}] = \hat{\mathbf{J}}_0^{-1} \tilde{\epsilon}_s, \quad (4.93)$$

where $\dot{\alpha}$ is the enhanced strain vector that must be determined from an equilibrium condition.

The virtual work equation can be written

$$\delta W_{\text{int}} = \int_e \delta \epsilon^T \sigma dV_e, \quad (4.94a)$$

$$= \int_e \delta \epsilon^T J^{-1} \mathbf{J}^T \mathbf{S} dV_e, \quad (4.94b)$$

$$= \int_e \delta \epsilon^T J^{-1} \mathbf{J}^T \left(\int_0^t \mathbf{C}^{SE} \dot{\mathbf{E}} d\tau \right) dV_e, \quad (4.94c)$$

$$= \int_e \delta \epsilon^T J^{-1} \mathbf{J}^T \left(\int_0^t J J^{-T} \mathbf{C}^\sigma \dot{\epsilon} d\tau \right) dV_e, \quad (4.94d)$$

$$= \int_p \delta \epsilon^T \bar{j}_0 \mathbf{J}_0^T \left(\int_0^t J J_0^{-T} \mathbf{C}^\sigma \dot{\epsilon} d\tau \right) dV_p, \quad (4.94e)$$

$$\approx \int_p \delta \tilde{\epsilon}^T \left(\int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^\sigma \hat{\mathbf{J}}_0^{-1} \tilde{\epsilon} d\tau \right) dV_p, \quad (4.94f)$$

where J is the determinant of the deformation gradient, \mathbf{J} is the push-forward operator of a symmetric 2nd order tensor, j_0 is the determinant of the jacobian matrix, \bar{j}_0 is the determinant of the jacobian matrix at time 0, σ is the true stress tensor, \mathbf{S} is the 2nd Piola-Kirchhoff stress tensor, \mathbf{C}^{SE} is the material tangent modulus, \mathbf{C}^σ is the spatial tangent modulus and V_e is the volume of the element. In the above, we have used various transformation

formulae between different stress and constitutive tensors. At this point we are only interested in how to handle the stabilization portion of the strain rate field, the constant part is only used to update the midpoint stress as usual. Because of orthogonality properties of the involved matrices, it turns out that we may just insert the expression for the stabilization strain rate field to get

$$\delta W_{\text{int}}^s \approx \int_p \delta \tilde{\mathbf{e}}_s^T \left(\int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^\sigma \hat{\mathbf{J}}_0^{-1} \tilde{\mathbf{e}}_s d\tau \right) dV_p, \quad (4.95a)$$

$$= [\delta \tilde{\mathbf{u}}^T \quad \delta \boldsymbol{\alpha}^T] \int_p \begin{bmatrix} \tilde{\mathbf{B}}_s(\boldsymbol{\xi})^T \\ \tilde{\mathbf{G}}(\boldsymbol{\xi})^T \end{bmatrix} \left(\int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^\sigma \hat{\mathbf{J}}_0^{-1} [\tilde{\mathbf{B}}_s(\boldsymbol{\xi}) \quad \tilde{\mathbf{G}}(\boldsymbol{\xi})] \begin{bmatrix} \tilde{\mathbf{u}} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix} d\tau \right) dV_p. \quad (4.95b)$$

The stabilization contribution to the internal force vector is given by

$$\begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{J}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \int_p \begin{bmatrix} \tilde{\mathbf{B}}_s(\boldsymbol{\xi})^T \\ \tilde{\mathbf{G}}(\boldsymbol{\xi})^T \end{bmatrix} \left(\int_0^t \frac{V_e}{8} \hat{\mathbf{J}}_0^{-T} \mathbf{C}^\sigma \hat{\mathbf{J}}_0^{-1} [\tilde{\mathbf{B}}_s(\boldsymbol{\xi}) \quad \tilde{\mathbf{G}}(\boldsymbol{\xi})] \begin{bmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\boldsymbol{\alpha}} \end{bmatrix} d\tau \right) dV_p. \quad (4.96)$$

In a discretization, the condition $\mathbf{f}_\alpha = \mathbf{0}$ is used to determine $\Delta \boldsymbol{\alpha}$, the increment of the enhanced strain variables, from $\Delta \mathbf{u}$, the increment in displacements. This is inserted back into the expression for the internal force vectors to determine \mathbf{f}_u , the stabilization contribution to the internal force vector.

The implementation of the element is very similar to the implementation of the one point integrated mean strain hexahedral by Flanagan and Belytschko [1981]. The hourglass forces are calculated in a different manner.

From the midpoint stress update we get a bulk and shear modulus characterizing the material at this specific point in time. From this we form the isotropic spatial tangent modulus \mathbf{C}^σ to be used for computing the stabilization force from [Equation \(4.96\)](#).

4.5 Fully Integrated Brick Elements and Mid-Step Strain Evaluation

To avoid locking in the fully integrated brick elements strain increments at a point in a constant pressure, solid element are defined by [see Nagtegaal, Parks, anmmmmmd Rice 1974]

(4.97)

$$\Delta \varepsilon_{zz} = \frac{\partial \Delta w}{\partial z^{n+1/2}} + \phi, \quad \Delta \varepsilon_{zx} = \frac{\frac{\partial \Delta u}{\partial z^{n+1/2}} + \frac{\partial \Delta w}{\partial x^{n+1/2}}}{2},$$

where ϕ modifies the normal strains to ensure that the total volumetric strain increment at each integration point is identical

$$\phi = \Delta \varepsilon_v^{n+1/2} - \frac{\frac{\partial \Delta u}{\partial x^{n+1/2}} + \frac{\partial \Delta v}{\partial y^{n+1/2}} + \frac{\partial \Delta w}{\partial z^{n+1/2}}}{3}, \quad (4.98)$$

and $\Delta \varepsilon_v^{n+1/2}$ is the average volumetric strain increment in the midpoint geometry

$$\frac{\frac{1}{3} \int_{v^{n+1/2}} \left(\frac{\partial \Delta u}{\partial x^{n+1/2}} + \frac{\partial \Delta v}{\partial y^{n+1/2}} + \frac{\partial \Delta w}{\partial z^{n+1/2}} \right) dv^{n+1/2}}{\int_{v^{n+1/2}} dv^{n+1/2}}, \quad (4.99)$$

Δu , Δv , and Δw are displacement increments in the x , y , and z directions, respectively, and

$$x^{n+1/2} = \frac{(x^n + x^{n+1})}{2}, \quad (4.100a)$$

$$y^{n+1/2} = \frac{(y^n + y^{n+1})}{2}, \quad (4.100b)$$

$$z^{n+1/2} = \frac{(z^n + z^{n+1})}{2}. \quad (4.100c)$$

To satisfy the condition that rigid body rotations cause zero straining, it is necessary to use the geometry at the mid-step in the evaluation of the strain increments. As the default, LS-DYNA currently uses the geometry at step $n + 1$ to save operations; however, for implicit calculations the mid-step strain calculation is always recommended, and, for explicit calculations, which involve rotating parts, the mid-step geometry should be used especially if the number of revolutions is large. The mid-step geometry can be activated either globally or for a subset of parts in the model by using the options on the control card, *CONTROL_ACCURACY.

Since the bulk modulus is constant in the plastic and viscoelastic material models, constant pressure solid elements result. In the thermoelastoplastic material, a constant temperature is assumed over the element. In the soil and crushable foam material, an average relative volume is computed for the element at time step $n + 1$, and the pressure and bulk modulus associated with this relative volume is used at each integration point. For equations of state, one pressure evaluation is done per element and is added to the deviatoric stress tensor at each integration point.

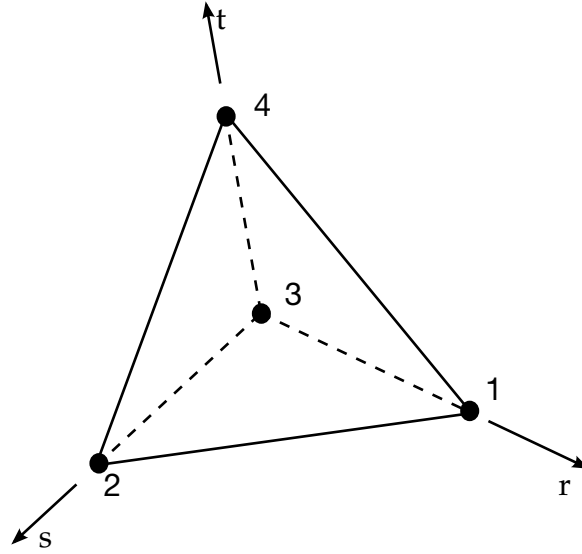


Figure 4.4. Four node tetrahedron.

The foregoing procedure requires that the strain-displacement matrix corresponding to [Equations \(4.66\)](#) and consistent with a constant volumetric strain, $\bar{\mathbf{B}}$, be used in the nodal force calculations [Hughes 1980]. It is easy to show that:

$$\mathbf{F} = \int_{v^{n+1}} (\bar{\mathbf{B}}^{n+1})^T \boldsymbol{\sigma}^{n+1} dv = \int_{v^{n+1}} (\mathbf{B}^{n+1})^T \boldsymbol{\sigma}^{n+1} dv, \quad (4.101)$$

and avoid the needless complexities of computing $\bar{\mathbf{B}}$.

4.6 Four Node Tetrahedron Element

The four node tetrahedron element with one point integration, shown in [Figure 4.4](#), is a simple, fast, solid element that has proven to be very useful in modeling low density foams that have high compressibility. For most applications, however, this element is too stiff to give reliable results and is primarily used for transitions in meshes. The formulation follows the formulation for the one point solid element with the difference that there are no kinematic modes, so hourglass control is not needed. The basis functions are given by:

$$N_1(r, s, t) = r, \quad (4.102a)$$

$$N_2(r, s, t) = s, \quad (4.102b)$$

$$N_3(r, s, t) = 1 - r - s - t, \quad (4.102c)$$

$$N_4(r, s, t) = t. \quad (4.102d)$$

If a tetrahedron element is needed, this element should be used instead of the collapsed solid element since it is, in general, considerably more stable in addition to being much faster. Automatic sorting can be used, see `*CONTROL_SOLID` keyword, to segregate these elements in a mesh of 8 node solids for treatment as tetrahedrons.

4.7 Nodal Pressure Tetrahedron

For applications requiring a tetrahedron mesh, the volume averaged tetrahedron type 13 is an interesting alternative to the standard single point tetrahedron, also known as the 1 point nodal pressure tetrahedron. The idea is to average the volumetric strain over adjacent elements to smooth the pressure response. To this end, we assume that E is the set of all such elements in a model, and likewise N is the set of all nodes belonging to these elements. We introduce the indicator

$$\chi_e^n = \begin{cases} 1/4 & \text{if node } n \text{ is connecting to element } e \\ 0 & \text{otherwise} \end{cases} \quad (4.103)$$

The volume of a node n is defined as

$$V_n = \sum_{e \in E} \chi_e^n V_e \quad (4.104)$$

where V_e is the exact volume of element e . This allows us to define an average jacobian \bar{J}_e for element e as

$$\bar{J}_e = \sum_{n \in N} \chi_e^n \left\{ \frac{V_n}{V_n^0} \right\} \quad (4.105)$$

which is essentially the relative volume. Here V_n^0 is the nodal volume in the reference configuration. The element is completely defined by the assumed deformation gradient given as

$$\bar{F}_e = \left(\frac{\bar{J}_e}{J_e} \right)^{1/3} F_e \quad (4.106)$$

where F_e is the deformation gradient from the isoparametric shape functions and $J_e = \det F_e$, also given as

$$J_e = \frac{V_e}{V_e^0}. \quad (4.107)$$

The assumed rate of deformation, derived from \bar{F}_e , is given as

$$\delta \bar{\epsilon}_e = \frac{1}{3} \left(\frac{\delta \bar{J}_e}{\bar{J}_e} - \frac{\delta J_e}{J_e} \right) \mathbf{I} + \delta F_e F_e^{-1} \quad (4.108)$$

meaning that the volumetric strain is replaced by that of the averaged one. Continuing, the virtual work equation is given as

$$\sum_{n \in N} \mathbf{f}_n^T \delta \mathbf{x}_n = \sum_{e \in E} \sigma_e : \delta \bar{\epsilon}_e J_e V_e^0 \quad (4.109)$$

which provides the equation for the nodal forces, \mathbf{f}_n . An expression for the strain-displacement matrix, \mathbf{B}_e^n , is deduced from combining (4.108) and the relation $\delta \bar{\epsilon}_e = \sum_{n \in N} \mathbf{B}_e^n \delta \mathbf{x}_n$. It turns out to be given as

$$\mathbf{B}_e^n = \sum_{f \in E} \sum_{m \in N} \chi_f^m \frac{V_f \chi_e^m}{\bar{J}_e V_m^0} \bar{\mathbf{B}}_f^n + \tilde{\mathbf{B}}_e^n \quad (4.110)$$

where $\bar{\mathbf{B}}_e^n$ and $\tilde{\mathbf{B}}_e^n$ are the volumetric and deviatoric parts of the standard (derived from isoparametric shape functions) strain displacement matrix. Noticable is that the support for the assumed strain displacement is not restricted to the nodal connectivity of a single

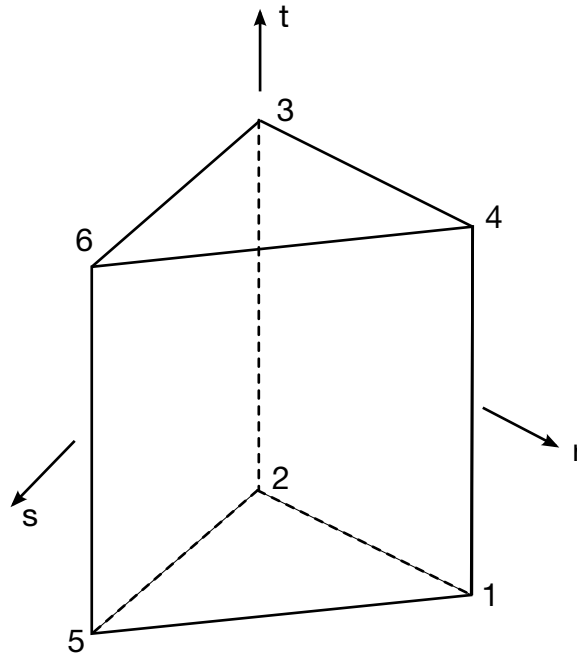


Figure 4.5. Six node Pentahedron.

element but is spread over a region of adjacent elements. This renders a less sparse stiffness matrix, but also explains the smoothening effect of the pressure.

4.8 Six Node Pentahedron Element

The pentahedron element with two point Gauss integration along its length, shown in [Figure 4.5](#), is a solid element that has proven to be very useful in modeling axisymmetric structures where wedge shaped elements are used along the axis-of-revolution. The formulation follows the formulation for the one point solid element with the difference that, like the tetrahedron element, there are no kinematic modes, so hourglass control is not needed. The basis functions are given by:

$$N_1(r, s, t) = \frac{1}{2}(1 - t)r, \quad (4.111a)$$

$$N_2(r, s, t) = \frac{1}{2}(1 - t)(1 - r - s), \quad (4.111b)$$

$$N_3(r, s, t) = \frac{1}{2}(1 + t)(1 - r - s), \quad (4.111c)$$

$$N_4(r, s, t) = \frac{1}{2}(1 + t)r, \quad (4.111d)$$

$$N_5(r, s, t) = \frac{1}{2}(1 - t)s, \quad (4.111e)$$

$$N_6(r, s, t) = \frac{1}{2}(1 + t)s. \quad (4.111f)$$

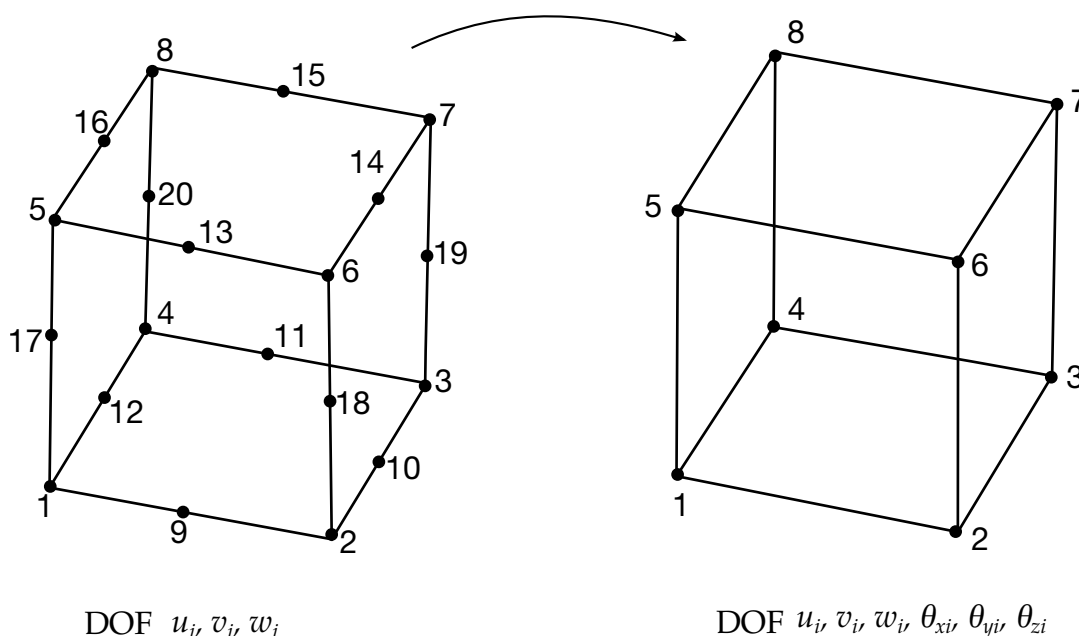


Figure 4.6. The 20-node solid element is transformed to an 8-node solid with 6 degrees-of-freedom per node.

If a pentahedron element is needed, this element should be used instead of the collapsed solid element since it is, in general, more stable and significantly faster. Automatic sorting can be used, see *CONTROL_SOLID keyword, to segregate these elements in a mesh of 8 node solids for treatment as pentahedrons. Selective-reduced integration is used to prevent volumetric locking, i.e., a constant pressure over the domain of the element is assumed.

4.9 Fully Integrated Brick Element With 48 Degrees-of-Freedom

The forty-eight degree of freedom brick element is derived from the twenty node solid element; see [Figure 4.6](#), through a transformation of the nodal displacements and rotations of the mid-side nodes [Yunus, Pawlak, and Cook, 1989]. This element has the advantage that shell nodes can be shared with brick nodes and that the faces have just four nodes — a real advantage for the contact-impact logic. The accuracy of this element is relatively good for problems in linear elasticity but degrades as Poisson's ratio approaches the incompressible limit. This can be remedied by using incompatible modes in the element formulation, but such an approach seems impractical for explicit computations.

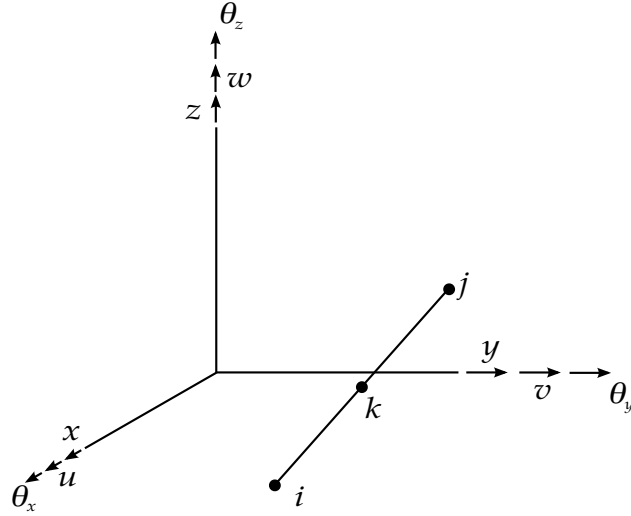


Figure 4.7. A typical element edge is shown from [Yunus, Pawlak, and Cook, 1989].

The instantaneous velocity for a midside node k is given as a function of the corner node velocities as (See [Figure 4.7](#)),

$$\begin{aligned}\dot{u}_k &= \frac{1}{2}(\dot{u}_i + \dot{u}_j) + \frac{y_j - y_i}{8}(\dot{\theta}_{zj} - \dot{\theta}_{zi}) + \frac{z_j - z_i}{8}(\dot{\theta}_{yi} - \dot{\theta}_{yj}), \\ \dot{v}_k &= \frac{1}{2}(\dot{v}_i + \dot{v}_j) + \frac{z_j - z_i}{8}(\dot{\theta}_{xj} - \dot{\theta}_{xi}) + \frac{x_j - x_i}{8}(\dot{\theta}_{zi} - \dot{\theta}_{zj}), \\ \dot{w}_k &= \frac{1}{2}(\dot{w}_i + \dot{w}_j) + \frac{x_j - x_i}{8}(\dot{\theta}_{yj} - \dot{\theta}_{yi}) + \frac{y_j - y_i}{8}(\dot{\theta}_{xi} - \dot{\theta}_{xj}),\end{aligned}\quad (4.112)$$

where $u, v, w, \theta_x, \theta_y$, and θ_z are the translational and rotational displacements in the global x, y , and z directions. The velocity field for the twenty-node hexahedron element in terms of the nodal velocities is:

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_{20} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \phi_1 & \phi_2 & \dots & \phi_{20} & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \phi_1 & \phi_2 & \dots & \phi_{20} \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \vdots \\ \dot{u}_{20} \\ \dot{v}_1 \\ \vdots \\ \dot{v}_{20} \\ \dot{w}_1 \\ \vdots \\ \dot{w}_{20} \end{Bmatrix}, \quad (4.113)$$

where ϕ_i are given by [Bathe and Wilson 1976] as,

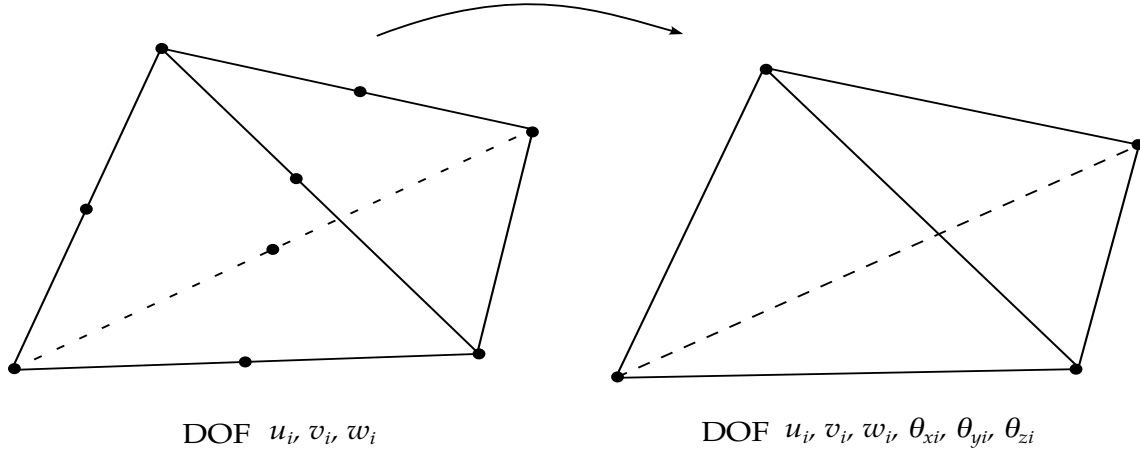


Figure 4.8. Twenty-four degrees of freedom tetrahedron element [Yunus, Pawlak, and Cook, 1989].

$$\begin{aligned}
 \phi_1 &= g_1 - \frac{(g_9 + g_{12} + g_{17})}{2}, & \phi_5 &= g_5 - \frac{(g_{13} + g_{16} + g_{17})}{2}, \\
 \phi_2 &= g_2 - \frac{(g_9 + g_{10} + g_{18})}{2}, & \phi_6 &= g_6 - \frac{(g_{13} + g_{14} + g_{18})}{2}, \\
 \phi_3 &= g_3 - \frac{(g_{10} + g_{11} + g_{19})}{2}, & \phi_7 &= g_7 - \frac{(g_{14} + g_{15} + g_{19})}{2}, \\
 \phi_4 &= g_4 - \frac{(g_{11} + g_{12} + g_{20})}{2}, & \phi_8 &= g_8 - \frac{(g_{15} + g_{16} + g_{20})}{2}, \\
 \phi_i &= g_i \text{ for } i = 9, \dots, 20 \\
 g_i &= G(\xi, \xi_i)G(\eta, \eta_i)G(\zeta, \zeta_i), \\
 G(\beta, \beta_i) &= \begin{cases} \frac{1}{2}(1 + \beta\beta_i) & \text{for } \beta_i = \pm 1; \quad \beta = \xi, \eta, \zeta \\ 1 - \beta^2 & \text{for } \beta_i = 0 \end{cases}
 \end{aligned} \tag{4.114}$$

The standard formulation for the twenty node solid element is used with the above transformations. The element is integrated with a fourteen point integration rule [Cook 1974]:

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta = B_6 [f(-b, 0, 0) + f(b, 0, 0) + f(0, -b, 0) + \dots + (6 \text{ terms})] + C_8 [f(-c, -c, -c) + f(c, -c, -c) + f(c, c, -c) + \dots + (8 \text{ terms})], \tag{4.115}$$

where

$$\begin{aligned}
 B_6 &= 0.8864265927977938, & b &= 0.7958224257542215, \\
 C_8 &= 0.3351800554016621, & c &= 0.7587869106393281.
 \end{aligned} \tag{4.116}$$

Cook reports that this rule has nearly the same accuracy as the twenty-seven point Gauss rule, which is very costly. The difference in cost between eight point and fourteen point integration, though significant, is necessary to eliminate the zero energy modes.

4.10 Fully Integrated Tetrahedron Element With 24 Degrees-of-Freedom

The twenty-four degree of freedom tetrahedron element is derived from the ten-node tetrahedron element; see [Figure 4.8](#), following the same procedure used above for the forty-eight degree of freedom brick element [Yunus, Pawlak, and Cook, 1989]. This element has the advantage that shell nodes can be shared with its nodes and it is compatible with the brick element discussed above. The accuracy of this element is relatively good—at least when compared to the constant strain tetrahedron element. This is illustrated by the bar impact example in [Figure 4.9](#) which compares the 12 and 24 degree of freedom tetrahedron elements. The 12 degree-of-freedom tetrahedron displays severe volumetric locking.

In our implementation we have not strictly followed the reference. In order to prevent locking in applications that involve incompressible behavior, selective reduced integration is used with a total of 5 integration points. Although this is rather expensive, no zero energy modes exist. We use the same approach in determining the rotary mass that is used in the implementation of the shell elements.

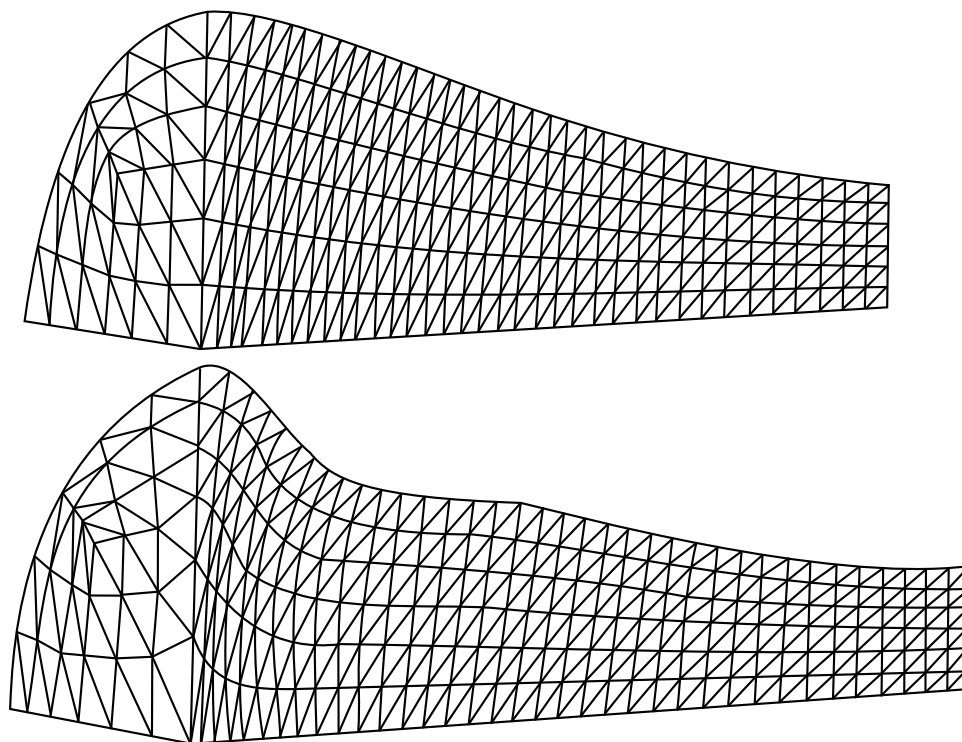


Figure 4.9. A comparison of the 12 and 24 degree-of-freedom tetrahedron elements is shown. The 12 degree-of-freedom tetrahedron element on the top displays severe volumetric locking.

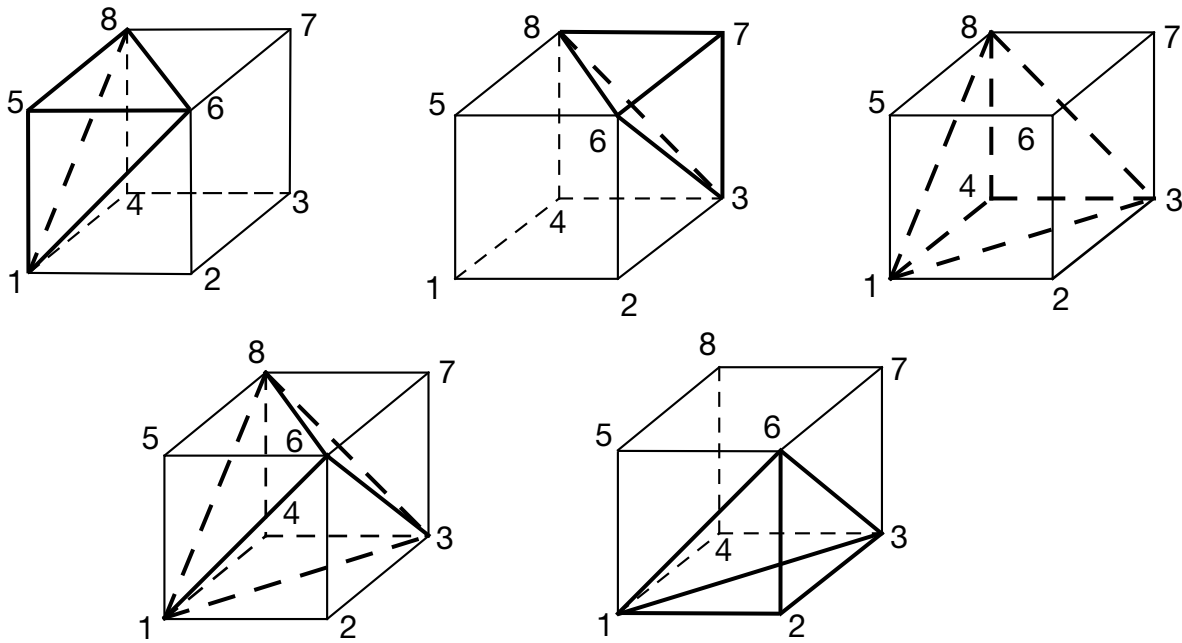


Figure 4.10a. Construction of a hexahedron element with five tetrahedrons.

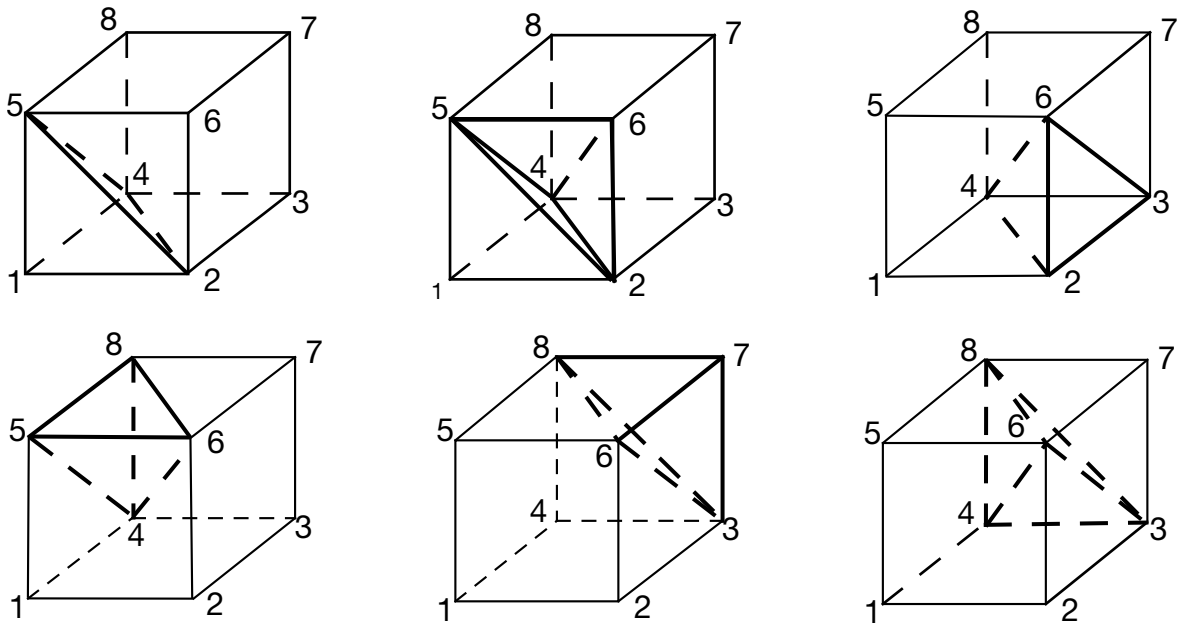


Figure 4. 10b. Construction of a hexahedron element with six tetrahedrons

Figures 4.10a and 4. 10b show the construction of a hexahedron element from five and six tetrahedron elements, respectively. When two sides of the adjacent bricks made from five tetrahedrons are together, it is likely that four unique triangular segments exist. This creates a problem in LS-PREPOST, which uses the numbering as a basis for eliminating interior polygons prior to display. Consequently, the graphics in the post-processing phase can be considerably slower with the degeneration in Figure 4.10a. However, marginally better results may be obtained with five tetrahedrons per hexahedron due to a better constraint count.

4.11 The Cosserat Point Elements in LS-DYNA

4.11.1 Introduction

The Cosserat Point Elements (CPE) are based on the works by Jabareen et.al.[1,2]. In contrast with a conventional approach, the CPE is treated as a structure rather than a continuum. The kinematic variables of the CPE are characterized by a volume averaged deformation gradient and other measures of inhomogeneous deformations. The CPE models the response of a simple continuum (not a generalized Cosserat media) and the additional kinematic degrees of freedom model physical modes of deformation of the structure. Moreover, the CPE uses a hyperelastic constitutive equation for elastic response with the strain energy of the CPE being separated additively into a part dependent on the strain energy of the three-dimensional material and another strain energy associated with inhomogeneous deformations. Also, for the tetrahedral CPE use is made of a new measure of dilatation that stabilizes hourglass type modes in large deformations. This formulation is valid for large deformations and the coefficients in the inhomogeneous strain energy are ingeniously determined by comparison with exact linear solutions. This ensures that CPE yields accurate results for elementary deformation modes in linear elasticity. Moreover, using the average deformation gradient for the response to homogeneous deformations the CPE formulation can be coupled with arbitrary material models in LS-DYNA. Still the element is, due to its complexity and slight loss of generality, first and foremost recommended for hyperelasticity in implicit analysis.

Section 4.11.2 through 4.11.8 describes the theory for the hexahedral CPE element, the tetrahedron is based on the same concepts except for the volumetric correction presented in Section 4.11.5. For more details we refer to [1] and [2]. We end with two examples in Sections 4.11.9 and 4.11.10.

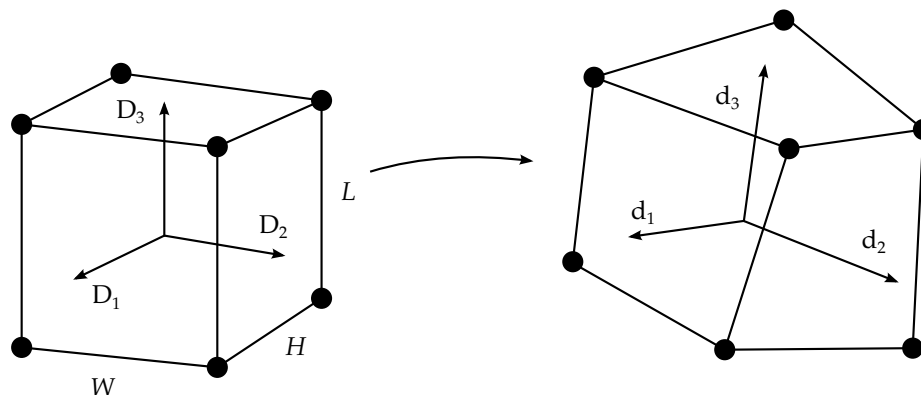


Figure 4.11. The CPE hexahedron

4.11.2 Geometry

The geometry of the hexahedral CPE element is characterized by the three-dimensional directors \mathbf{D}_i and \mathbf{d}_i , $i = 0, 1, \dots, 7$, where the formers are associated with the reference configuration and the latters with the current configuration. The reciprocal vectors \mathbf{D}^i and \mathbf{d}^i , $i = 1, 2, 3$, are such that

$$\mathbf{d}_i \cdot \mathbf{d}^j = \mathbf{D}_i \cdot \mathbf{D}^j = \delta_{ij}, \quad i, j = 1, 2, 3, \quad (4.11.117)$$

and we also have that

$$|\mathbf{D}_i| = 1, \quad i = 1, 2, 3, \quad (4.11.118)$$

The coordinates θ^i , $i = 1, 2, 3$, ranges between

$$-H/2 \leq \theta^1 \leq H/2, \quad -W/2 \leq \theta^2 \leq W/2, \quad -L/2 \leq \theta^3 \leq L/2, \quad (4.11.119)$$

and the \mathbf{A} matrix is given such that

$$\mathbf{D}_i = \sum_{j=0}^7 A_i^j \mathbf{X}_j, \quad \mathbf{d}_i = \sum_{j=0}^7 A_i^j \mathbf{x}_j, \quad i = 0, 1, \dots, 7, \quad (4.11.120)$$

where \mathbf{X}_i and \mathbf{x}_i are the nodal coordinates in the reference and current configuration, respectively. Hence the \mathbf{A} matrix represents the mapping between the nodal coordinates and the Cosserat point directors. The reference position vector \mathbf{X} is expressed as

$$\mathbf{X} = \mathbf{X}(\theta^1, \theta^2, \theta^3) = \sum_{j=0}^7 N^j(\theta^1, \theta^2, \theta^3) \mathbf{D}_j, \quad (4.11.121)$$

and likewise the current position \mathbf{x} as

$$\mathbf{x} = \mathbf{x}(\theta^1, \theta^2, \theta^3) = \sum_{j=0}^7 N^j(\theta^1, \theta^2, \theta^3) \mathbf{d}_j, \quad (4.11.122)$$

The shape functions are given as

$$\begin{aligned} N^0 &= 1, & N^1 &= \theta^1, & N^2 &= \theta^2, & N^3 &= \theta^3, \\ N^4 &= \theta^1 \theta^2, & N^5 &= \theta^1 \theta^3, & N^6 &= \theta^2 \theta^3, & N^7 &= \theta^1 \theta^2 \theta^3. \end{aligned} \quad (4.11.123)$$

4.11.3 Deformation and strain

The deformation measures used are

$$\mathbf{F} = \sum_{i=1}^3 \mathbf{d}_i \otimes \mathbf{D}^i, \quad \beta_i = \mathbf{F}^{-1} \mathbf{d}_{i+3} - \mathbf{D}_{i+3} \quad (i = 1, 2, 3, 4), \quad \bar{\mathbf{F}} = \mathbf{F} \left(\mathbf{I} + \sum_{i=1}^4 \beta_i \otimes \mathbf{V}^i \right), \quad (4.11.124)$$

where $\bar{\mathbf{F}}$ is the volume averaged deformation gradient and thus represents the homogeneous deformations whereas β_i are measures of the inhomogeneous deformations. As for the \mathbf{V}^i we have

$$\begin{aligned}
V &= HWL \left(\mathbf{D}_1 \times \mathbf{D}_2 \cdot \mathbf{D}_3 + \frac{H^2}{12} \mathbf{D}_4 \times \mathbf{D}_5 \cdot \mathbf{D}_1 + \frac{W^2}{12} \mathbf{D}_6 \right. \\
&\quad \left. \times \mathbf{D}_4 \cdot \mathbf{D}_2 + \frac{L^2}{12} \mathbf{D}_5 \times \mathbf{D}_6 \cdot \mathbf{D}_3 \right), \\
\mathbf{V}^1 &= V^{-1} HWL \left(\frac{H^2}{12} \mathbf{D}_5 \times \mathbf{D}_1 + \frac{W^2}{12} \mathbf{D}_2 \times \mathbf{D}_6 \right), \\
\mathbf{V}^2 &= V^{-1} HWL \left(\frac{H^2}{12} \mathbf{D}_1 \times \mathbf{D}_4 + \frac{L^2}{12} \mathbf{D}_6 \times \mathbf{D}_3 \right), \\
\mathbf{V}^3 &= V^{-1} HWL \left(\frac{W^2}{12} \mathbf{D}_4 \times \mathbf{D}_2 + \frac{L^2}{12} \mathbf{D}_3 \times \mathbf{D}_5 \right), \\
\mathbf{V}^4 &= \mathbf{0},
\end{aligned} \tag{4.11.125}$$

The velocity gradient consistent with $\bar{\mathbf{F}}$ is given as

$$\bar{\mathbf{L}} = \dot{\bar{\mathbf{F}}} \bar{\mathbf{F}}^{-1} = \mathbf{L} + \sum_{j=1}^4 (\dot{\mathbf{d}}_{j+3} - \mathbf{L} \mathbf{d}_{j+3}) \otimes \mathbf{V}^j \bar{\mathbf{F}}^{-1}, \tag{4.11.126}$$

where $\mathbf{L} = \dot{\mathbf{F}} \mathbf{F}^{-1}$, which in turn gives the rate –of-deformation and spin tensors as

$$\dot{\bar{\boldsymbol{\varepsilon}}} = \frac{1}{2} (\bar{\mathbf{L}} + \bar{\mathbf{L}}^T), \quad \bar{\boldsymbol{\omega}} = \frac{1}{2} (\bar{\mathbf{L}} - \bar{\mathbf{L}}^T). \tag{4.11.127}$$

4.11.4 Stress and Force

On the other hand,

$$\mathbf{L} = \sum_{i=1}^3 \dot{\mathbf{d}}_i \otimes \mathbf{d}^i, \tag{4.11.128}$$

so we can rewrite $\bar{\mathbf{L}}$ as

$$\bar{\mathbf{L}} = \sum_{i=1}^3 \dot{\mathbf{d}}_i \otimes \mathbf{d}^i \left(\bar{\mathbf{F}} - \sum_{j=1}^4 \mathbf{d}_{j+3} \otimes \mathbf{V}^j \right) \bar{\mathbf{F}}^{-1} + \sum_{i=1}^4 \dot{\mathbf{d}}_{i+3} \otimes \mathbf{V}^i \bar{\mathbf{F}}^{-1}. \tag{4.11.129}$$

The Cauchy stress is given by the constitutive law

$$\boldsymbol{\sigma}^\nabla = \mathbf{f}_{\text{hypo}}(\dot{\bar{\boldsymbol{\varepsilon}}}, \dots), \tag{4.11.130}$$

or for the hyperelastic case

$$\boldsymbol{\sigma} = \mathbf{f}_{\text{hyper}}(\bar{\mathbf{F}}, \dots), \tag{4.11.131}$$

so the nonzero internal force vectors are given as

$$\mathbf{t}_\sigma^i = \bar{J} V \sigma \bar{\mathbf{F}}^{-T} \left(\bar{\mathbf{F}}^T - \sum_{j=1}^4 \mathbf{V}^j \otimes \mathbf{d}_{j+3} \right) \mathbf{d}^i, \quad \mathbf{t}_\sigma^{i+3} = V \sigma \bar{\mathbf{F}}^{-T} \mathbf{V}^i, \quad i = 1, 2, 3. \quad (4.11.132)$$

The first expression above can in turn be rewritten as

$$\mathbf{t}_\sigma^i = \left(\bar{J} V \sigma - \sum_{j=1}^4 \mathbf{t}_\sigma^{j+3} \otimes \mathbf{d}_{j+3} \right) \mathbf{d}^i, \quad i = 1, 2, 3. \quad (4.11.133)$$

4.11.5 CPE3D10 modification

For the 10-noded tetrahedron, a modified deformation gradient is used in the constitutive law, given by

$$\tilde{\mathbf{F}} = \left(\tilde{J} / \bar{J} \right)^{1/3} \bar{\mathbf{F}}, \quad (4.11.134)$$

meaning it has been modified for the volumetric response. Here

$$\tilde{J} = \bar{J} + \eta J, \quad (4.11.135)$$

This gives a consistent velocity gradient as

$$\tilde{\mathbf{L}} = \bar{\mathbf{L}} + \frac{1}{3} \frac{J}{\tilde{J}} \left(\dot{\eta} + \eta \left\{ \frac{\dot{J}}{\bar{J}} - \frac{\dot{\tilde{J}}}{\tilde{J}} \right\} \right) \mathbf{I}. \quad (4.11.136)$$

The Cauchy stress is now given by the constitutive law

$$\sigma^\nabla = \mathbf{f}_{\text{hypo}}(\dot{\tilde{\mathbf{E}}}, \dots), \quad \dot{\tilde{\mathbf{E}}} = \frac{1}{2} (\tilde{\mathbf{L}} + \tilde{\mathbf{L}}^T), \quad (4.11.137)$$

or for the hyperelastic case

$$\sigma = \mathbf{f}_{\text{hyper}}(\tilde{\mathbf{F}}, \dots), \quad (4.11.138)$$

and the corresponding internal force can be identified through a principle of virtual work

$$\sum_{i=0}^9 \mathbf{t}_\sigma^i \cdot \mathbf{d}_i = \tilde{J} V \tilde{\mathbf{L}} \cdot \sigma. \quad (4.11.139)$$

For the hyperelastic special case we have

$$\sigma = \frac{\partial \Sigma}{\partial \tilde{J}} \mathbf{I} + 2 \tilde{J}^{-1} \left[\bar{\mathbf{F}}' \frac{\partial \Sigma}{\partial \bar{\mathbf{C}}'} \bar{\mathbf{F}}'^T - \frac{1}{3} \left(\frac{\partial \Sigma}{\partial \bar{\mathbf{C}}'} \cdot \bar{\mathbf{C}}' \right) \mathbf{I} \right], \quad (4.11.140)$$

and since

$$\begin{aligned}
\tilde{\mathbf{L}} = & \sum_{i=1}^3 \dot{\mathbf{d}}_i \otimes \mathbf{d}^i \left(\bar{\mathbf{F}} - \sum_{j=1}^6 \mathbf{d}_{j+3} \otimes \mathbf{V}^j \right) \bar{\mathbf{F}}^{-1} + \sum_{i=1}^6 \dot{\mathbf{d}}_{i+3} \otimes \mathbf{V}^i \bar{\mathbf{F}}^{-1} \\
& + \frac{1}{3} \frac{J}{\bar{J}} \sum_{j=1}^6 \sum_{k=1}^3 \left[\frac{\partial \eta}{\partial b_j^k} \left((\mathbf{d}^k)^T \dot{\mathbf{d}}_{j+3} - \sum_{i=1}^3 \mathbf{d}^i \cdot \mathbf{d}_{j+3} \mathbf{d}^k \cdot \dot{\mathbf{d}}_i \right) \right] \mathbf{I} \\
& + \frac{1}{3} \frac{J}{\bar{J}} \left[\eta \left(\sum_{j=1}^6 \sum_{i=1}^3 \mathbf{d}^i \cdot \mathbf{d}_{j+3} \dot{\mathbf{d}}_i \cdot \bar{\mathbf{F}}^{-T} \mathbf{V}^j - \sum_{i=1}^6 \dot{\mathbf{d}}_{i+3} \cdot \bar{\mathbf{F}}^{-T} \mathbf{V}^i \right) \right] \mathbf{I},
\end{aligned} \tag{4.11.141}$$

putting back these expressions into the virtual work expression above and adding the hourglass internal forces from below we get the same as in [2].

4.11.6 Hourglass

The hourglass resistance is based on a strain energy potential given as

$$\psi = \frac{V\mu}{12(1-\nu)} \sum_{i=1}^4 \sum_{j=1}^3 \sum_{k=1}^4 \sum_{l=1}^3 b_i^j B_{jl}^{ik} b_k^l, \tag{4.11.142}$$

where the inhomogeneous (hourglass) strain quantities are defined as

$$b_i^j = \boldsymbol{\beta}_i \cdot \mathbf{D}^j, \quad i = 1, 2, 3, 4, \quad j = 1, 2, 3, \tag{4.11.143}$$

and the constant symmetric matrix \mathbf{B} contains geometry and constitutive information for obtaining accurate results for small deformations. Furthermore, μ represents the shear modulus of the material and ν is the Poisson's ratio. The hourglass force is then given as

$$\mathbf{t}_h^i = \left(\frac{\partial \psi}{\partial \mathbf{d}_i} \right)^T = \sum_{j=1}^4 \sum_{k=1}^3 \frac{\partial \psi}{\partial b_j^k} \left(\frac{\partial b_j^k}{\partial \mathbf{d}_i} \right)^T, \quad i = 0, 1, \dots, 7. \tag{4.11.144}$$

Differentiating the strain quantities results in

$$\begin{aligned}
\frac{\partial b_j^k}{\partial \mathbf{d}_i} &= -\mathbf{d}^i \cdot \mathbf{d}_{j+3} (\mathbf{d}^k)^T, \quad i = 1, 2, 3 \\
\frac{\partial b_j^k}{\partial \mathbf{d}_{i+3}} &= \delta_i^j (\mathbf{d}^k)^T, \quad i = 1, 2, 3, 4
\end{aligned} \tag{4.11.145}$$

which inserted into the expression for the force yields

$$\begin{aligned}
\mathbf{t}_h^i &= - \left(\sum_{j=1}^4 \mathbf{t}_h^{j+3} \otimes \mathbf{d}_{j+3} \right) \mathbf{d}^i, \quad i = 1, 2, 3 \\
\mathbf{t}_h^{i+3} &= \sum_{j=1}^3 \frac{\partial \psi}{\partial b_i^j} \mathbf{d}^j, \quad i = 1, 2, 3, 4
\end{aligned} \tag{4.11.146}$$

4.11.7 Comparison to Jabareen & Rubin

Putting the material and hourglass force together yields

$$\mathbf{t}^i = \mathbf{t}_\sigma^i + \mathbf{t}_h^i, \quad i = 0, 1, \dots, 7, \quad (4.11.147)$$

hence

$$\begin{aligned} \mathbf{t}^i &= \left(\bar{J} V \boldsymbol{\sigma} - \sum_{j=1}^4 \mathbf{t}^{j+3} \otimes \mathbf{d}_{j+3} \right) \mathbf{d}^i \quad i = 1, 2, 3 \\ \mathbf{t}^{i+3} &= \bar{J} V \boldsymbol{\sigma} \bar{\mathbf{F}}^{-T} \mathbf{V}^i + \sum_{j=1}^3 \frac{\partial \psi}{\partial b_i^j} \mathbf{d}^j \quad i = 1, 2, 3, 4 \end{aligned} \quad (4.11.148)$$

In [1], the hourglass strains b_i^j are augmented with the geometry parameters H, W and L , whereas here we have merged this information into the constitutive matrix \mathbf{B} for the purpose of simplifying the presentation. In [1] a hyperelastic constitutive law is assumed with a strain energy potential $\Sigma = \Sigma(\bar{\mathbf{C}})$ and the Cauchy stress is then given as

$$\boldsymbol{\sigma} = 2\bar{J}^{-1} \bar{\mathbf{F}} \frac{\partial \Sigma}{\partial \bar{\mathbf{C}}} \bar{\mathbf{F}}^T. \quad (4.11.149)$$

Taking all this into account, and consulting (2.13-2.14) in [1], the implemented CPE element in LS-DYNA should be consistent with the theory.

4.11.8 Nodal formulation

To be of use in LS-DYNA, the CPE element has to be formulated in terms of the nodal variables, meaning that the internal forces that are conjugate to $\dot{\mathbf{x}}_i$, $i = 0, 1, \dots, 7$, with respect to internal energy rate are given as

$$\mathbf{f}^i = \sum_{j=0}^7 A_j^i \mathbf{t}^j, \quad i = 0, 1, \dots, 7. \quad (4.11.150)$$

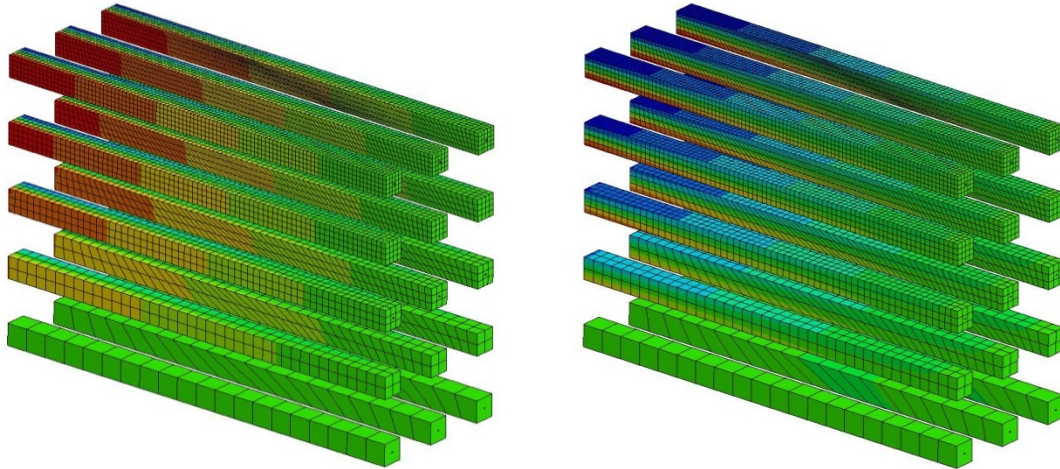


Figure 4.11. Stress profile for tip loading in two directions for various mesh densities and distortions

4.11.9 Cantilever Beam Example

First a mesh distortion test for small deformations where a cantilever beam with various mesh densities and distortions were simulated. The stress profiles for loading in two directions are shown in [Figure 4.11](#). The tip displacements were monitored and compared to the analytical solution, and the results for the CPE element is very promising as is shown in figure below and [Table 4.11](#) when compared to the Belytschko-Bindeman and Puso element [3,4].

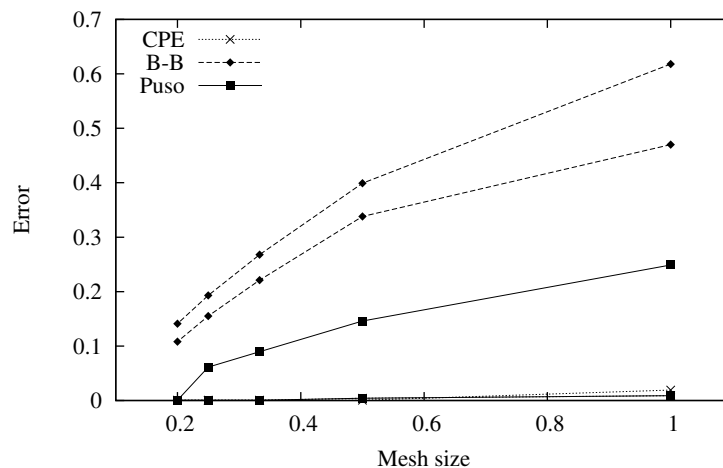


Figure 4.11. Mesh convergence rate for different element formulations

Cosserat	Belytschko-Bindeman	Puso
1.7%	61.8%	24.8%
0.8%	46.8%	14.7%
0.6%	40.0%	14.5%
0.3%	39.8%	9.2%
0.2%	33.9%	8.5%
0.2%	27.0%	6.2%
0.1%	24.6%	5.3%
0.1%	22.3%	3.6%
0.1%	19.0%	0.9%
0.1%	15.4%	0.3%

Table 4.11. Comparison to analytical solution for tip loading

4.11.10 Compression Test for Rubber Example

The second example is a plane strain deformation of an incompressible rubber block on a frictionless surface when partially loaded by a rigid plate. The block is modeled with quadratic tetrahedrons and a large deformation formulation applies. Five different mesh topologies are investigated for two different element formulations (CPE and full integration) where the orientation of the tetrahedral elements is different in each mesh whereas the mesh density is kept constant. The outer edges of the block for the different meshes are depicted in the figure above and indicate once again that the CPE element formulation is relatively insensitive to the mesh.

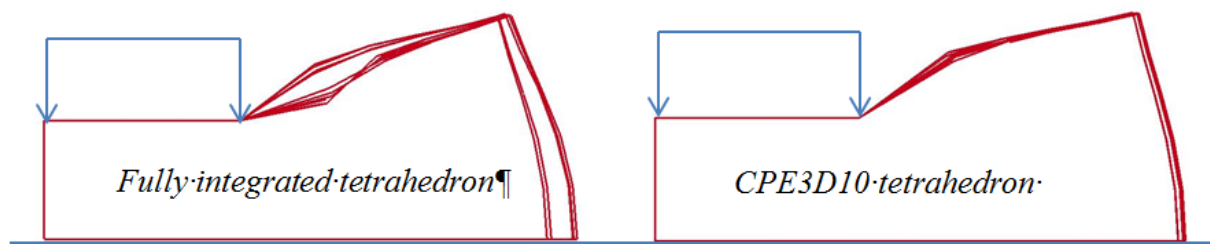


Figure 4.11. Insensitivity illustration of the 10-noded CPE tetrahedron

4.11.11 References

- [1]M. Jabareen and M.B. Rubin, A Generalized Cosserat Point Element (CPE) for Isotropic Nonlinear Elastic Materials including Irregular 3-D Brick and Thin Structures, J. Mech. Mat. Struct. 3-8, pp. 1465-1498, 2008.
- [2]M. Jabareen, E. Hanukah and M.B. Rubin, A Ten Node Tetrahedral Cosserat Point Element (CPE) for Nonlinear Isotropic Elastic Materials. Comput Mech 52, pp 257-285, 2013.

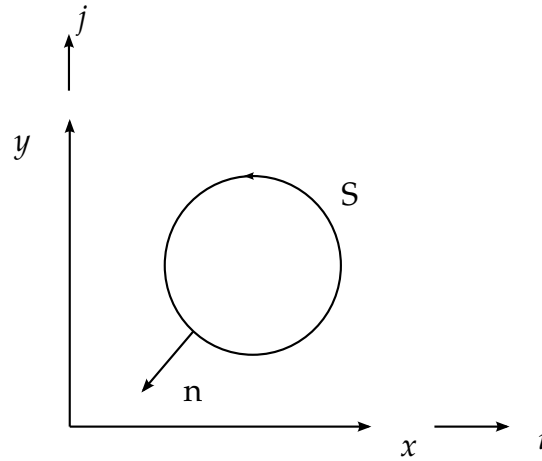


Figure 4.11. The contour S encloses an area A .

- [3]M.A. Puso, A Highly Efficient Enhanced Assumed Strain Physically Stabilized Hexahedral Element, In. J. Num. Methods. Eng 49-8, pp. 1029-1064, 2000.
- [4]T. Belytschko and L.P. Bindeman , Assumed Strain Stabilization of the Eight Node Hexahedral Element, Comp. Methods Appl. Mech. Eng. 105-2, pp. 225-260, 1993.

4.12 Integral Difference Scheme as Basis For 2D Solids

Two dimensional solid element in LS-DYNA include:

- Plane stress 2D element
- Plane strain 2D shell element
- Axisymmetric 2D Petrov-Galerkin (area weighted) element
- Axisymmetric 2D Galerkin (volume weighted) element

These elements have their origins in the integral difference method of Noh [1964] which is also used the HEMP code developed by Wilkins [1964, 1969]. In LS-DYNA, both two dimensional planar and axisymmetric geometries are defined in the xy plane. In axisymmetric geometry, however, the x axis corresponds to the radial direction and the y axis becomes the axis of symmetry. The integral difference method defines the components of the gradient of a function F in terms of the line integral about the contour S which encloses the area A :

$$\begin{aligned}\frac{\partial F}{\partial x} &= \lim_{A \rightarrow 0} \frac{\int_c F(\hat{\mathbf{n}} \cdot \hat{\mathbf{x}}) dS}{|A|}, \\ \frac{\partial F}{\partial y} &= \lim_{A \rightarrow 0} \frac{\int_c F(\hat{\mathbf{n}} \cdot \hat{\mathbf{y}}) dS}{|A|}.\end{aligned}\tag{4.151}$$

Here, $\hat{\mathbf{n}}$ is the normal vector to S and $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are unit vectors in the x and y directions, respectively. See [Figure 4.11](#).

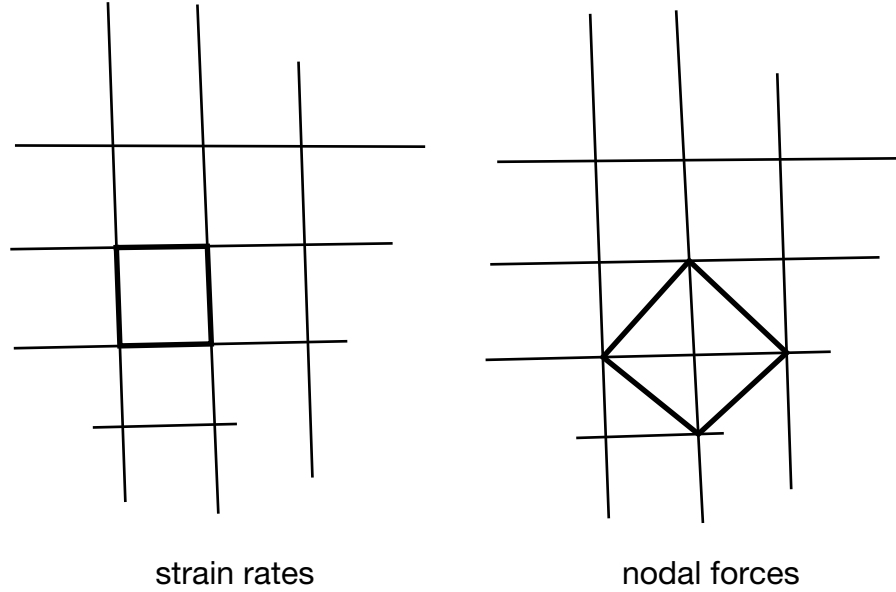


Figure 4.12. Strain rates are element centered and nodal forces are node centered.

In this approach the velocity gradients which define the strain rates are element centered, and the velocities and nodal forces are node centered. See [Figure 4.12](#). Noting that the normal vector $\hat{\mathbf{n}}$ is defined as:

$$\hat{\mathbf{n}} = \frac{\partial y}{\partial S} \hat{\mathbf{x}} + \frac{\partial x}{\partial S} \hat{\mathbf{y}}, \quad (4.152)$$

and referring to [Figure 4.13](#), we can expand the numerator in [equation \(4.64\)](#):

$$\begin{aligned} \int_c F(\hat{\mathbf{n}} \cdot \hat{\mathbf{x}}) dS &= \int F \frac{\partial y}{\partial S} dS \\ &= F_{23}(y_3 - y_2) + F_{34}(y_4 - y_3) + F_{41}(y_1 - y_4) + F_{12}(y_2 - y_1), \end{aligned} \quad (4.153)$$

where $F_{ij} = (F_i + F_j)/2$.

Therefore, letting A again be the enclosed area, the following expressions are obtained:

$$\begin{aligned} \frac{\partial F}{\partial x} &= \frac{F_{23}(y_3 - y_2) + F_{34}(y_4 - y_3) + F_{41}(y_1 - y_4) + F_{12}(y_2 - y_1)}{A} \\ &= \frac{(F_2 - F_4)(y_3 - y_1) + (y_2 - y_4)(F_3 - F_1)}{2A}. \end{aligned} \quad (4.154)$$

Hence, the strain rates in the x and y directions become:

$$\begin{aligned} \frac{\partial F}{\partial x} &= \frac{F_{23}(y_3 - y_2) + F_{34}(y_4 - y_3) + F_{41}(y_1 - y_4) + F_{12}(y_2 - y_1)}{A} \\ &= \frac{(F_2 - F_4)(y_3 - y_1) + (y_2 - y_4)(F_3 - F_1)}{2A}, \end{aligned} \quad (4.155)$$

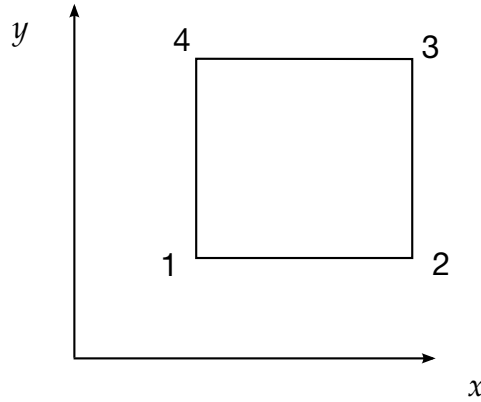


Figure 4.13. Element numbering.

and

$$\varepsilon_{yy} = \frac{\partial \dot{y}}{\partial y} = \frac{(\dot{y}_2 - \dot{y}_4)(x_3 - x_1) + (x_2 - x_4)(\dot{y}_3 - \dot{y}_1)}{2A}. \quad (4.156)$$

The shear strain rate is given by:

$$\varepsilon_{xy} = \frac{1}{2} \left(\frac{\partial \dot{y}}{\partial x} + \frac{\partial \dot{x}}{\partial y} \right), \quad (4.157)$$

where

$$\begin{aligned} \frac{\partial \dot{y}}{\partial x} &= \frac{(\dot{y}_2 - \dot{y}_4)(y_3 - y_1) + (y_2 - y_4)(\dot{y}_3 - \dot{y}_1)}{2A}, \\ \frac{\partial \dot{x}}{\partial y} &= \frac{(\dot{x}_2 - \dot{x}_4)(x_3 - x_1) + (x_2 - x_4)(\dot{x}_3 - \dot{x}_1)}{2A}. \end{aligned} \quad (4.158)$$

The zero energy modes, called hourglass modes, as in the three dimensional solid elements, can be a significant problem. Consider the velocity field given by: $\dot{x}_3 = \dot{x}_1$, $\dot{x}_2 = \dot{x}_4$, $\dot{y}_3 = \dot{y}_1$, and $\dot{y}_2 = \dot{y}_4$. As can be observed from [Equations \(4.97\) and \(4.98\)](#), $\varepsilon_{xx} = \varepsilon_{yy} = \varepsilon_{xy} = 0$ and the element "hourglass" irrespective of the element geometry. In the two-dimensional case, two modes exist versus twelve in three dimensions. The hourglass treatment for these modes is identical to the approach used for the shell elements, which are discussed later.

In two-dimensional planar geometries for plane stress and plane strain, the finite element method and the integral finite difference method are identical. The velocity strains are computed for the finite element method from the equation:

$$\dot{\varepsilon} = \mathbf{B}\mathbf{v}, \quad (4.159)$$

where $\dot{\varepsilon}$ is the velocity strain vector, \mathbf{B} is the strain displacement matrix, and \mathbf{v} is the nodal velocity vector. [Equation \(4.100a\)](#) exactly computes the same velocity strains as the integral difference method if

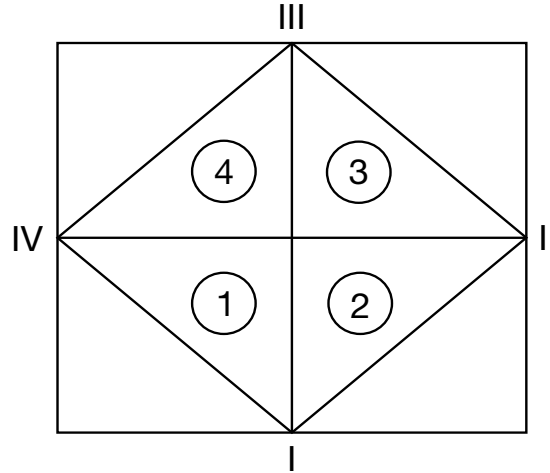


Figure 4.14. The finite difference stencil for computing nodal forces is shown.

$$\mathbf{B} = \mathbf{B}(s, t)|_{s=t=0}. \quad (4.160)$$

The update of the nodal forces also turns out to be identical. The momentum equations in two-dimensional planar problems are given by

$$\begin{aligned} \frac{1}{\rho} \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \right) &= \ddot{x}, \\ \frac{1}{\rho} \left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \right) &= \ddot{y}. \end{aligned} \quad (4.161)$$

Referring to [Figure 4.14](#), the integral difference method gives [Equation \(4.113\)](#):

$$\frac{1}{\rho} \frac{\partial \sigma_{xx}}{\partial x} = \frac{\sigma_{xx_1}(y_I - y_{IV}) + \sigma_{xx_2}(y_{II} - y_I) + \sigma_{xx_3}(y_{III} - y_{II}) + \sigma_{xx_4}(y_{IV} - y_{III})}{\frac{1}{2}(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)}. \quad (4.162)$$

An element wise assembly of the discretized finite difference equations is possible leading to a finite element like finite difference program. This approach is used in the DYNA2D program by Hallquist [1980].

In axisymmetric geometries additional terms arise that do not appear in planar problems:

$$\begin{aligned} \frac{1}{\rho} \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\sigma_{xx} - \sigma_{\theta\theta}}{x} \right) &= \ddot{x}, \\ \frac{1}{\rho} \left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{x\theta}}{\partial x} \right) &= \ddot{y}, \end{aligned} \quad (4.163)$$

where again note that y is the axis of symmetry and x is the radial direction. The only difference between finite element approach and the finite difference method is in the

treatment of the terms, which arise from the assumption of axisymmetry. In the finite difference method the radial acceleration is found from the calculation:

$$\ddot{x} = \frac{1}{2} \left[\frac{\sigma_{xx_1}(y_I - y_{IV}) + \sigma_{xx_2}(y_{II} - y_I) + \sigma_{xx_3}(y_{III} - y_{II}) + \sigma_{xx_4}(y_{IV} - y_{III})}{(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} - \frac{\sigma_{xy_1}(x_I - x_{IV}) + \sigma_{xy_2}(x_{II} - x_I) + \sigma_{xy_3}(x_{III} - x_{II}) + \sigma_{xy_4}(x_{IV} - x_{III})}{(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} \right] + \beta, \quad (4.164)$$

where β_{fe} is found by a summation over the four surrounding elements:

$$\beta = \frac{1}{4} \sum_{i=1}^4 \left[\frac{\sigma_{xx_i} - \sigma_{\theta\theta_i}}{(\rho x)_i} \right]. \quad (4.165)$$

x_i is the centroid of the i th element defined as the ratio of its volume V_i and area A_i :

$$x_i = \frac{V_i}{A_i}, \quad (4.166)$$

$\sigma_{\theta\theta_i}$ is the hoop stress, and ρ_i is the *current* density.

When applying the Petrov-Galerkin finite element approach, the weighting functions are divided by the radius, r :

$$\int \frac{1}{r} \Phi^T (\nabla \cdot \sigma + \mathbf{b} - \rho \ddot{\mathbf{u}}) dV = \int \Phi^T (\nabla \cdot \sigma + \mathbf{b} - \rho \ddot{\mathbf{u}}) dA = 0, \quad (4.167)$$

where the integration is over the current geometry. This is sometimes referred to as the "Area Galerkin" method. This approach leads to a time dependent mass vector. LS-DYNA also has an optional Galerkin axisymmetric element, which leads to a time independent mass vector. For structural analysis problems where pressures are low the Galerkin approach works best, but in problems of hydrodynamics where pressures are a large fraction of the elastic modulus, the Petrov-Galerkin approach is superior since the behavior along the axis of symmetry is correct.

The Petrov-Galerkin approach leads to equations similar to finite differences. The radial acceleration is given by.

$$\ddot{x} = \frac{1}{2} \left[\frac{\sigma_{xx_1}(y_I - y_{IV}) + \sigma_{xx_2}(y_{II} - y_I) + \sigma_{xx_3}(y_{III} - y_{II}) + \sigma_{xx_4}(y_{IV} - y_{III})}{(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} - \frac{\sigma_{xy_1}(x_I - x_{IV}) + \sigma_{xy_2}(x_{II} - x_I) + \sigma_{xy_3}(x_{III} - x_{II}) + \sigma_{xy_4}(x_{IV} - x_{III})}{(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} \right] + \beta_{fe}, \quad (4.168)$$

where β_{fe} is now area weighted.

$$\beta_{fe} = \frac{1}{4(\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4)} \sum_{i=1}^4 \left[\frac{(\sigma_{xx_i} - \sigma_{\theta\theta_i}) A_i}{x_i} \right]. \quad (4.169)$$

In LS-DYNA, the two-dimensional solid elements share the same constitutive sub-routines with the three-dimensional elements. The plane stress element calls the plane

stress constitutive models for shells. Similarly, the plane strain and axisymmetric elements call the full three-dimensional constitutive models for solid elements. Slight overheads exist since the strain rate components $\dot{\epsilon}_{yz}$ and $\dot{\epsilon}_{zx}$ are set to zero in the two-dimensional case prior to updating the six stress component; consequently, the additional work is related to having six stresses whereas only four are needed. A slowdown of LS-DYNA compared with DYNA2D of fifteen percent has been observed; however, some of the added cost is due to the internal and hourglass energy calculations, which were not done in DYNA2D.

4.12.1 Rezoning With 2D Solid Elements

Lagrangian solution techniques generally function well for problems when element distortions are moderate. When distortions become excessive or when material breaks up, i.e., simply connected regions become multi-connected, these codes break down, and an Eulerian approach is a necessity. Between these two extremes, applications exist for which either approach may be appropriate but Lagrangian techniques are usually preferred for speed and accuracy. Rezoning may be used to extend the domain of application for Lagrangian codes.

Rezoning capability was added to DYNA2D in 1980 and to LS-DYNA in version 940. In the current implementation the rezoning can be done interactively and used to relocate the nodal locations within and on the boundary of parts. This method is sometimes referred to as *r*-adaptive.

The rezoning is accomplished in three steps listed below:

1. Generate nodal values for all variables to be remapped
2. Rezone one or more materials either interactively or automatically with command file.
3. Initialize remeshed regions by interpolating from nodal point values of old mesh.

In the first step each variable is approximated globally by a summation over the number of nodal points n :

$$g(r, z) = \sum_{i=1}^n g_i \Phi_i(r, z), \quad (4.170)$$

where

Φ_i = set of piecewise continuous global basis functions
 g_i = nodal point values

Given a variable to be remapped $h(r, z)$, a least squares best fit is found by minimizing the functional

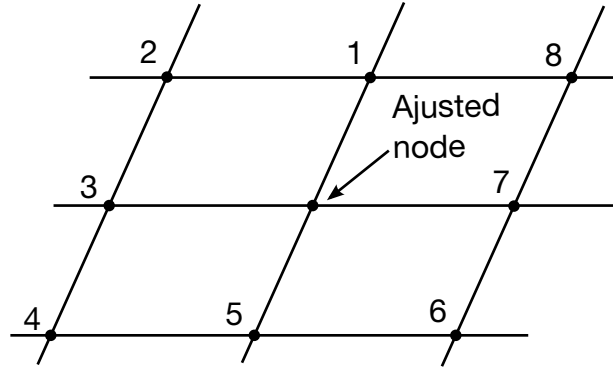


Figure 4.15. The stencil used to relax an interior nodal point.

$$\Pi = \int (g - h)^2 dA, \quad (4.171)$$

i.e.,

$$\frac{d\Pi}{dg_i} = 0, \quad i = 1, 2, \dots, n. \quad (4.172)$$

This yields the set of matrix equations

$$\mathbf{M}\mathbf{g} = \mathbf{f}, \quad (4.173)$$

where

$$\begin{aligned} \mathbf{M} &= \sum \mathbf{M}^e = \sum_e \int \Phi \Phi^T dA, \\ \mathbf{f} &= \sum \mathbf{f}^e = \sum_e \int h \Phi dA. \end{aligned} \quad (4.174)$$

Lumping the mass makes the calculation of g trivial

$$\begin{aligned} M_i &= \sum_j^n M_{ij}, \\ g_i &= \frac{f_i}{M_i}. \end{aligned} \quad (4.175)$$

In step 2, the interactive rezoning phase permits:

- Plotting of solution at current time
- Deletion of elements and slidelines
- Boundary modifications via dekinks, respacing nodes, etc.
- Mesh smoothing

A large number of interactive commands are available and are described in the Help package. Current results can be displayed by

- Color fringes
- Contour lines

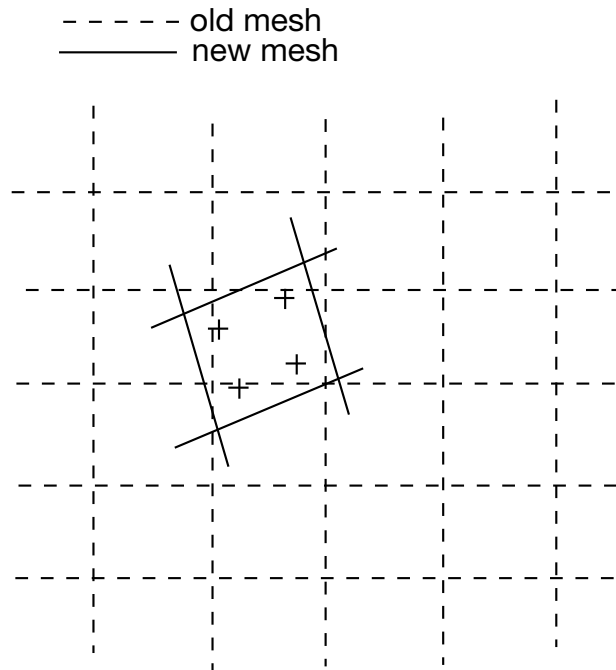


Figure 4.16. A four point Gauss quadrature rule over the new element is used to determine the new element centered value.

- Vectors plots
- Principal stress lines
- Deformed meshes and material outlines
- Profile plots
- Reaction forces
- Interface pressures along 2D contact interfaces

Three methods are available for smoothing:

- Equipotential
- Isoparametric
- Combination of equipotential and isoparametric.

In applying the relaxation, the new nodal positions are found and given by [Equation \(1.176\)](#)

$$\begin{aligned}
 x &= \frac{\sum_{i=1}^8 \xi_i x_i}{\sum_{i=1}^8 \xi_i}, \\
 y &= \frac{\sum_{i=1}^8 \xi_i y_i}{\sum_{i=1}^8 \xi_i},
 \end{aligned}
 \tag{4.176}$$

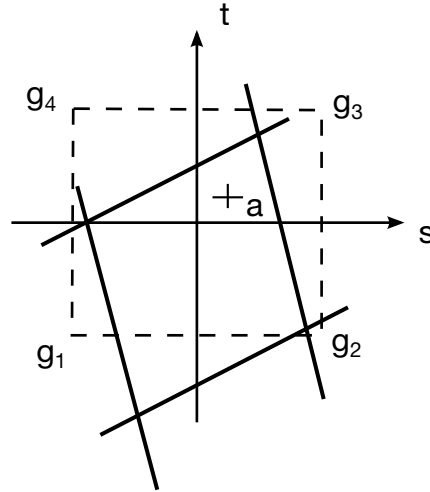


Figure 4.17. A four point Gauss quadrature rule over the new element is used to determine the new element centered value.

where the nodal positions relative to the node being moved are shown in the sketch in [Figure 4.15](#).

The weights, $\tilde{\xi}_i$, for equipotential smoothing are

$$\tilde{\xi}_1 = \tilde{\xi}_5 = \frac{1}{4} [(x_7 - x_3)^2 + (y_7 - y_3)^2], \quad (4.177a)$$

$$\tilde{\xi}_4 = \tilde{\xi}_8 = -\tilde{\xi}_2, \quad (4.177b)$$

$$\tilde{\xi}_2 = \tilde{\xi}_6 = \frac{1}{2} [(x_1 - x_5)(x_7 - x_3) + (y_1 - y_5)(y_7 - y_3)], \quad (4.177c)$$

$$\tilde{\xi}_3 = \tilde{\xi}_7 = \frac{1}{4} [(x_1 - x_5)^2 + (y_1 - y_5)^2], \quad (4.177d)$$

and are given by

$$\begin{aligned} \tilde{\xi}_1 &= \tilde{\xi}_3 = \tilde{\xi}_5 = \tilde{\xi}_7 = .50, \\ \tilde{\xi}_2 &= \tilde{\xi}_4 = \tilde{\xi}_6 = \tilde{\xi}_8 = -.25, \end{aligned} \quad (4.178)$$

for isoparametric smoothing. Since logical regularity is not assumed in the mesh, we construct the nodal stencil for each interior node and then relax it. The nodes are iteratively moved until convergence is obtained. In Chapter 14 of this manual, the smoothing procedures are discussed for three-dimensional applications.

The new element centered values, h^* , computed in [Equation \(1.179\)](#) are found by a 4 point Gauss Quadrature as illustrated in [Figure 4.16](#).

$$h^* = \frac{\int g dA}{\int dA}. \quad (4.179)$$

The Gauss point values are interpolated from the nodal values according to [Equation \(1.180\)](#). This is also illustrated by [Figure 4.17](#).

$$g_a = \sum \phi_i(s_a, t_a) g_i. \quad (4.180)$$

4.13 Solid Elements 18 and -18 – Incompatible Modes

4.13.1 Introduction

Isoparametric Strain Elements

The design of a finite element is essentially the design of its deformation gradient F , the mother of all strains. The deformation gradient is in a continuum mechanical context defined as

$$F = \frac{\partial x}{\partial X} \quad (4.13.181)$$

where x is the current spatial location for a given reference material coordinate X , and is a measure of the stretch and rotation of an infinitesimal line element in the body. Based on F , examples of different strain measures are

$$\begin{aligned} \varepsilon &= \ln F^T F && \text{(natural strain)} \\ E &= \frac{1}{2} (F^T F - I) && \text{(Green strain)} \\ D &= \text{sym} \dot{F} F^{-1} && \text{(rate of deformation)} \\ e &= \frac{1}{2} (I - F^{-T} F^{-1}) && \text{(Almansi strain).} \end{aligned} \quad (4.13.182)$$

For a given strain measure we can imagine that all possible strains generated by arbitrary deformations constitutes a *strain space* \mathcal{E} , so for instance $E \in \mathcal{E}$ if we happen to work with the Green strain measure. Now, when a continuum body is discretized into finite elements, each *element* is in the simplest setting represented by m coordinate points x_i and associated isoparametric shape functions $N_i, i = 1, \dots, m$. The spatial location of a material point in the element is written as

$$x = x_i N_i \quad (4.13.183)$$

and a direct application results in

$$F = x_i \frac{\partial N_i}{\partial X}. \quad (4.13.184)$$

The consequence of the discretization is that the *numerical* strains using this latter F are restricted to a finite dimensional subspace \mathcal{E}_h of the space of *continuum* strains \mathcal{E} , and the *quality* of the element comes down to the richness of this subspace $\mathcal{E}_h \subset \mathcal{E}$. To this end, for a given level of discretization h , the subspace is to most extent defined by the choice of shape functions N_i . It is for instance known that high order polynomials tend to improve the situation as they appropriately enlarge the space of strains.

Considering low order hexahedral elements, for which the shape functions are the standard tri-linear isoparametric shape functions, the methodology presented above inevitably leads to *locking* phenomena, i.e., the element becomes too stiff. In short, this is explained by the element's inability to represent a physical deformation mode without exhibiting spurious strain. Typically, a pure bending deformation is accompanied with either excessive volumetric or shear strain that absorbs energy and stiffens the response. To use a phrasing that reconnects to the discussion above, the strain space \mathcal{E}_h is too small.

Assumed Strain Elements

A way to deal with locking is to augment the deformation gradient based on reasonable assumptions, leading to an *assumed strain element* with deformation gradient F_{as} . There are restrictions associated with such a modification, a crucial one is to maintain frame invariance by making sure that the assumed velocity gradient $L_{as} = \dot{F}_{as}F_{as}^{-1}$ is skew-symmetric whenever elements undergo rigid body motion. A not so severe restriction is that the element satisfies various *patch tests* commonly found in the literature. A first example of such an approach is to assume that the deformation gradient is constant in an element and given as either its value in the center, $F_{as} = F_0$, or its integrated mean, $F_{as} = \bar{F}$. This will on the one hand alleviate locking but on the other render an incomplete element in the sense that non-rigid deformation modes won't generate strains, i.e., the symmetric part of L_{as} is zero. Such elements require artificial (hourglass) stabilization and are available as the type 1 element in LS-DYNA. A complete element is resulted from assuming that only the volumetric part of the deformation gradient is constant, and the deviatoric part unaffected. This is the so called *selective reduced integration element* available as type 2 in LS-DYNA. We either have $F_{as} = (J_0/J)^{1/3}F$ or $F_{as} = (\bar{J}/J)^{1/3}F$ depending on the choice of averaging, where $J = \det F$ is the jacobian. The choice of \bar{J} is preferred over J_0 as this element satisfies the patch test, the other does not. This element alleviates volumetric locking but still exhibits stiff behavior in shear, in particular for elements with poor aspect ratio. A heuristic approach to deal with the latter is presented in Borrvall (2009) and the corresponding element is available as types -1 and -2 in LS-DYNA. Those elements do not satisfy the patch test.

Enhanced Assumed Strain Elements

Pioneered by Juan Simo and co-workers in the late 80's and early 90's, the concept of *enhanced* assumed strain elements emerged in the finite element community, see Simo et.al (1990,1993). For these elements the deformation gradient is modified by adding an enhanced contribution F_e

$$F_{eas} = F + F_e. \quad (4.13.185)$$

The enhanced contribution F_e is in turn a function of an independent field α that can be seen as representing deformation modes in which the element needs to be relaxed, i.e., this is a way to enlarge the strain space. The presence of α implies the existence of a work conjugate field f_α , and appropriate application of the virtual work principle results in that an additional set of equations, $f_\alpha = 0$, needs to be solved. Fortunately, α is assumed

discontinuous over element boundaries, which means that this compatibility equation can be solved locally for each element and the size of the global system of equations is thus unaffected. Nevertheless, solving many local sets of nonlinear equations adds a significant overall cost to evaluating the nodal forces, and is the main reason why it has not been given much attention in an *explicit* finite element context. For *implicit* time integration the story may be told differently, the price of spending more time in element routines may give payback in the currency of better results and larger time steps.

4.13.2 Theory

General

A framework for the design of enhanced assumed strain elements sufficient for satisfying frame invariance and patch tests was introduced by Simo and Rafai (1990). Within this framework, Simo et.al. (1993) developed some specific elements with interesting properties, and this is an attempt to summarize the equations behind these. For this we treat a single element occupying a region Ω in space, and emphasize that the notation used from hereon is independent of the notation used in the introduction.

Let x denote the coordinate vector for the nodes in the current configuration, x_n the coordinate vector in the previous (converged) configuration. We then let $u = x - x_n$ and $v = u/\Delta t$ be the displacement and velocity vectors, respectively, where Δt is the time step. All vectors mentioned so far are of dimension 24, 8 nodes times 3 degrees of freedom per node. The element is characterized by a vector α representing the enhanced deformation modes. There are essentially 4 enhanced modes, each represented by three triplets $\alpha_i, i = 1,2,3$ and a fourth α_4 which is either 3 (for type 18) or 4 (for type -18) items long.

The incremental enhanced deformation gradient is denoted

$$\Delta F = \Delta F(x, x_n, \alpha) \quad (4.13.186)$$

and we form the current deformation gradient recursively as

$$F = \Delta F F_n \quad (4.13.187)$$

and the constitutive law takes a strain measure as input to compute or increment the resulting Cauchy stress σ . For hyperelastic based materials, F is used directly, whereas for hypoelastic materials we use the velocity gradient approximated as

$$L = \frac{2}{\Delta t} (\Delta F - I)(\Delta F + I)^{-1}. \quad (4.13.188)$$

Formulating the theory in the incremental instead of the total deformation gradient assures that the element satisfies incremental patch tests. Through a work principle, where we use δ_1 to denote the first variation, the identification of f_x and f_α can be made by the relation¹

¹ Note that we can use F or ΔF in the variational expressions, since the results would turn out the same, for the sake of convenience we use F .

$$\delta_1 \mathbf{x}^T \mathbf{f}_x + \delta_1 \boldsymbol{\alpha}^T \mathbf{f}_\alpha = \int_{\Omega} (\delta_1 \mathbf{F} \mathbf{F}^{-1}) : \boldsymbol{\sigma} d\Omega. \quad (4.13.189)$$

The stiffness matrix terms can likewise be identified through the second variation δ_2 of this work principle

$$\begin{aligned} \delta_1 \mathbf{x}^T \frac{\partial \mathbf{f}_x}{\partial \mathbf{x}} \delta_2 \mathbf{x} + \delta_1 \mathbf{x}^T \frac{\partial \mathbf{f}_x}{\partial \boldsymbol{\alpha}} \delta_2 \boldsymbol{\alpha} + \delta_1 \boldsymbol{\alpha}^T \frac{\partial \mathbf{f}_\alpha}{\partial \mathbf{x}} \delta_2 \mathbf{x} + \delta_1 \boldsymbol{\alpha}^T \frac{\partial \mathbf{f}_\alpha}{\partial \boldsymbol{\alpha}} \delta_2 \boldsymbol{\alpha} = \dots \\ = \int_{\Omega} \left(\delta_2 \delta_1 \mathbf{F} \mathbf{F}^{-1} + (\delta_2 \mathbf{F} \mathbf{F}^{-1})^T (\delta_1 \mathbf{F} \mathbf{F}^{-1}) \right) : \boldsymbol{\sigma} d\Omega \\ + \int_{\Omega} (\delta_1 \mathbf{F} \mathbf{F}^{-1}) : \mathbf{C} : (\delta_2 \mathbf{F} \mathbf{F}^{-1}) d\Omega. \end{aligned} \quad (4.13.190)$$

Worth noticing here is that for this relation to hold, \mathbf{C} is the constitutive modulus relating the *Truesdell* rate of Cauchy stress to the rate-of-deformation. Following Simo et.al. (1993) all integrals above are for element 18 numerically integrated using a 9 point integration scheme, while element -18 uses the traditional 8 Gauss point integration scheme.

The first of the force vectors, \mathbf{f}_x , should balance the other forces in the problem, while the second, \mathbf{f}_α , should solve the compatibility equation $\mathbf{f}_\alpha = \mathbf{0}$ for the current value of $\boldsymbol{\alpha}$. For this, we use a Newton method and the corresponding Hessian $\frac{\partial \mathbf{f}_\alpha}{\partial \boldsymbol{\alpha}}$. The elimination of $\boldsymbol{\alpha}$ on the element level requires that the global tangent stiffness matrix is condensed accordingly, we have

$$\mathbf{K}_x = \frac{d\mathbf{f}_x}{d\mathbf{x}} = \frac{\partial \mathbf{f}_x}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}_x}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{x}} \quad (4.13.191)$$

where $\frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{x}}$ is obtained through the variation of the compatibility equation

$$\frac{d\mathbf{f}_\alpha}{d\mathbf{x}} = \frac{\partial \mathbf{f}_\alpha}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}_\alpha}{\partial \boldsymbol{\alpha}} \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{x}} = \mathbf{0} \Rightarrow \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{x}} = - \left(\frac{\partial \mathbf{f}_\alpha}{\partial \boldsymbol{\alpha}} \right)^{-1} \frac{\partial \mathbf{f}_\alpha}{\partial \mathbf{x}} \quad (4.13.192)$$

so

$$\mathbf{K}_x = \frac{\partial \mathbf{f}_x}{\partial \mathbf{x}} - \frac{\partial \mathbf{f}_x}{\partial \boldsymbol{\alpha}} \left(\frac{\partial \mathbf{f}_\alpha}{\partial \boldsymbol{\alpha}} \right)^{-1} \frac{\partial \mathbf{f}_\alpha}{\partial \mathbf{x}}. \quad (4.13.193)$$

Element 18

The assumed incremental deformation gradient for element 18 (the Simo element) can be expressed as

$$\begin{aligned} \Delta \mathbf{F} = \mathbf{x}_i \frac{\partial N_i}{\partial \boldsymbol{\xi}} \Big|_0 \mathbf{J}_0^{-1} + \mathbf{x}_i \gamma_i \left(\frac{J_0}{J_{\bar{\xi}}} \right) \frac{\partial \mathcal{H}}{\partial \boldsymbol{\xi}} \mathbf{J}_0^{-1} + \boldsymbol{\alpha}_j \left(\frac{J_0}{J_{\bar{\xi}}} \right) \frac{\partial \tilde{N}_j}{\partial \boldsymbol{\xi}} \mathbf{J}_0^{-1} \\ + \left[\left(\frac{J_0}{J_{\bar{\xi}}} \right) \frac{\partial \tilde{N}_4}{\partial \boldsymbol{\xi}} \mathbf{J}_0^{-1} \boldsymbol{\alpha}_4 \right] \mathbf{x}_i \frac{\partial N_i}{\partial \boldsymbol{\xi}} \Big|_0 \mathbf{J}_0^{-1}, \end{aligned} \quad (4.13.194)$$

where the first two terms correspond to a modified Galerkin and the remaining are the enhanced contributions. The shape functions associated with the enhanced contributions are the so called Wilson incompatible shape functions

$$\tilde{N}_j = \frac{1}{2}(\tilde{\zeta}_j^2 - 1), \quad (4.13.195)$$

for $j = 1, 2, 3$, to relax bending deformations, and a function intended to alleviate volumetric locking

$$\tilde{N}_4 = \tilde{\zeta}_1 \tilde{\zeta}_2 \tilde{\zeta}_3. \quad (4.13.196)$$

Note that γ_i are the familiar γ -vectors (row vector of size 4 for each i) and

$$\mathcal{H} = \left(\frac{\partial \tilde{N}_4}{\partial \tilde{\zeta}} \quad \tilde{N}_4 \right)^T \quad (4.13.197)$$

the associated hourglass functions commonly used in the representation of low order hexahedral elements. We also use $J_{\tilde{\zeta}}/J_{\tilde{\zeta}}$ and J_0/J_0 to denote the jacobian matrix/determinant between the iso-parametric and previous configuration domains (at step n), and subscript 0 means that a quantity is evaluated at the center point of the domain. Noting that $\frac{\partial(\cdot)}{\partial \tilde{\zeta}}$ is a row vector of dimension 3, all operations in the expression for ΔF can be viewed as standard matrix notation. The variation operator is denoted δ_1 , and this applied on ΔF becomes

$$\begin{aligned} \delta_1 \Delta F = \delta_1 x_i \left\{ \frac{\partial N_i}{\partial \tilde{\zeta}} \Big|_0 J_0^{-1} + \gamma_i \left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \mathcal{H}}{\partial \tilde{\zeta}} J_0^{-1} + \left[\left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_4}{\partial \tilde{\zeta}} J_0^{-1} \alpha_4 \right] \frac{\partial N_i}{\partial \tilde{\zeta}} \Big|_0 J_0^{-1} \right\} \\ + \delta_1 \alpha_j \left\{ \left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_j}{\partial \tilde{\zeta}} J_0^{-1} \right\} + \left[\left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_4}{\partial \tilde{\zeta}} J_0^{-1} \delta_1 \alpha_4 \right] \left\{ x_i \frac{\partial N_i}{\partial \tilde{\zeta}} \Big|_0 J_0^{-1} \right\}, \end{aligned} \quad (4.13.198)$$

where sum over i is from 1 to 8, while sum over j is from 1 to 3. Because of the product term involving x and α , the second variation is non-vanishing, denoting this δ_2 we have

$$\delta_2 \delta_1 F = \left\{ \left[\left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_4}{\partial \tilde{\zeta}} J_0^{-1} \delta_1 \alpha_4 \right] \delta_2 x_i \frac{\partial N_i}{\partial \tilde{\zeta}} \Big|_0 J_0^{-1} + \left[\left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_4}{\partial \tilde{\zeta}} J_0^{-1} \delta_2 \alpha_4 \right] \delta_1 x_i \frac{\partial N_i}{\partial \tilde{\zeta}} \Big|_0 J_0^{-1} \right\}. \quad (4.13.199)$$

These variations, (4.13.198) and (4.13.199) can be inserted into (4.13.189) and (4.13.190), which completes the basic theory of this element.

Element -18

The assumed incremental deformation gradient for element -18 (the C3D8I element) can be expressed as

$$\Delta F = \left(x_i \frac{\partial N_i}{\partial \tilde{\zeta}} J_{\tilde{\zeta}}^{-1} + \alpha_j \left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_j}{\partial \tilde{\zeta}} J_0^{-1} \right) \exp \left(\left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \mathcal{H}^T \alpha_4 \right) \quad (4.13.200)$$

where \tilde{N}_j is given by (4.13.195), \mathcal{H} is given by (4.13.197) and again the sum over i is between 1 and 8 while the sum over j is between 1 and 3. The variation of (4.13.200) is

$$\delta_1 \Delta F = \left(\delta_1 x_i \frac{\partial N_i}{\partial \tilde{\zeta}} J_{\tilde{\zeta}}^{-1} + \delta_1 \alpha_j \left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \frac{\partial \tilde{N}_j}{\partial \tilde{\zeta}} J_0^{-1} \right) \exp \left(\left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \mathcal{H}^T \alpha_4 \right) + \Delta F \left(\frac{J_0}{J_{\tilde{\zeta}}} \right) \mathcal{H}^T \delta_1 \alpha_4 \quad (4.13.201)$$

while the second variation is given by

$$\begin{aligned}
\delta_2 \delta_1 \Delta F &= \delta_1 \Delta F \left(\frac{J_0}{J_\xi} \right) \mathcal{H}^T \delta_2 \alpha_4 + \delta_2 \Delta F \left(\frac{J_0}{J_\xi} \right) \mathcal{H}^T \delta_1 \alpha_4 \\
&\quad - \Delta F \left(\frac{J_0}{J_\xi} \right) \mathcal{H}^T \delta_1 \alpha_4 \left(\frac{J_0}{J_\xi} \right) \mathcal{H}^T \delta_2 \alpha_4.
\end{aligned} \tag{4.13.202}$$

Just as for element 18, inserting (4.13.201) and (4.13.202) into (4.13.189) and (4.13.190) makes the description complete.

A Final Remark

We should point out that when using ΔF in (4.13.189) and (4.13.190), it should be with respect to the current configuration. This means that ΔF in (4.13.198), (4.13.199), (4.13.201) and (4.13.202) should be replaced by identity I , and the jacobians J_0 and J_ξ evaluated in the current configurations.

5

Connectors

Various types of joining techniques are essential for establishing realistic simulation models. Examples of actual physical connections include bolts, spotwelds and rivets, and these may be modelled numerically by interpolation constraints, tied contacts, joints or SPR techniques. We here use the term *connectors* as an umbrella for the collection of these features, and this section aims to present the theory behind some of the most frequently occurring ones.

5.1 Self Piercing Rivet (SPR)

The illustration of an undeformed and deformed SPR (Self Piercing Rivet) configuration is shown in [sprinidef](#). Given a *rivet point* p and a *spotweld radius* R , two clusters of nodes are determined. The first set of nodes is associated with sheet A , while the other is associated with sheet B , and the rivet is supposed to form a connection between these two sheets. In the following we will use $*$ to denote either A or B when there can be no ambiguity, and explicitly write them out only when there is risk for confusion or needed. Furthermore, sum over repeated indices is implicitly assumed.

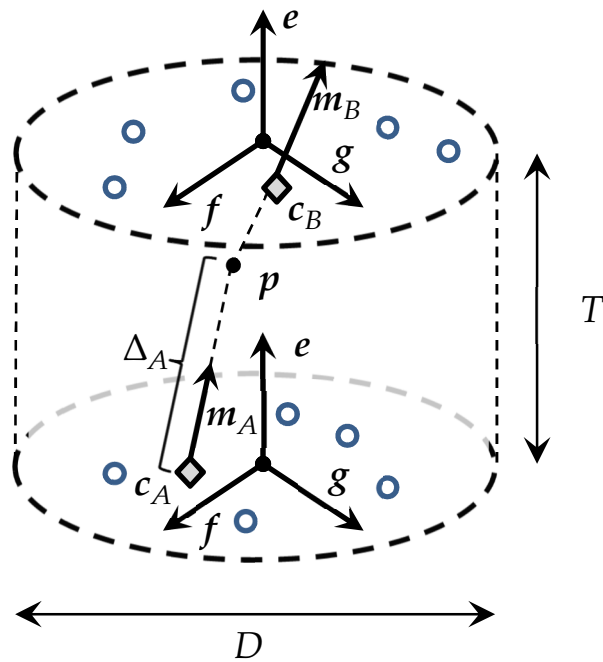
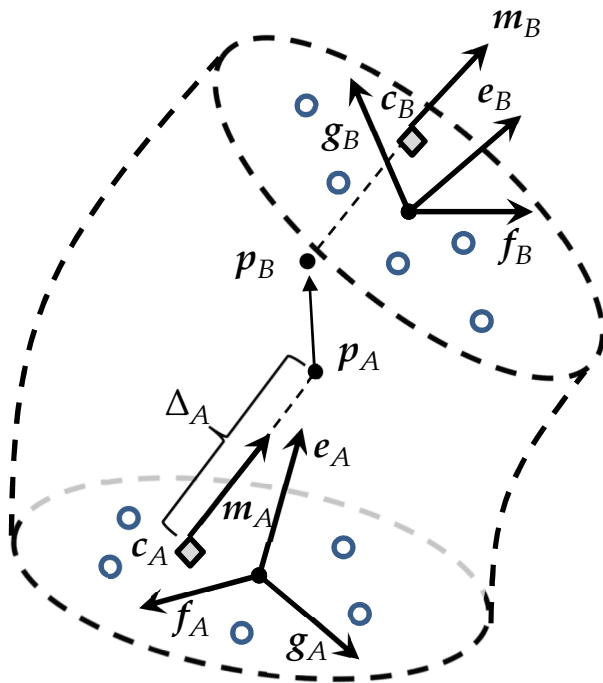
$t = 0$

 $t > 0$


Figure 1.1. Initial and deformed configurations of an SPR connection

The center of each cluster can be computed as

$$c_* = w_*^i x_*^i \quad (5.1)$$

where w_*^i and x_*^i are weights and nodal coordinates, respectively. The nodal weights are determined from user options, are positive and sum to unity. Note that the centers of clusters c_* , indicated by diamond shapes in the figure, do not necessarily align with the normal of the two sheets, the whereabouts are rather a consequence of the mesh topology and choice of weights. From this we determine the directed normals m_* as

$$m_A = \frac{p - c_A}{\|p - c_A\|}, \quad m_B = \frac{c_B - p}{\|c_B - p\|}, \quad (5.2)$$

assigned to each to the two sides, note that each is pointing from side A towards side B . From this the initial rivet point can be expressed in terms of both sides A and B ,

$$p = p_* = c_* + \Delta_* m_* \quad (5.3)$$

for constant scalars Δ_* , Δ_A being positive while Δ_B is negative. For the sake of clarity, only Δ_A is annotated in the figures above. Note that $p_A = p_B = p$ to begin with, meaning that there is no initial separation in translation.

We continue by constructing two instances of an orthonormal set of basis vectors $E = \{e, f, g\}$, one for each of the two clusters. These are constructed so that e is aligned with the normal of the two sheets, and again $E_A = E_B = E$ at the start of the simulation. These orthonormal sets are used for defining the local kinematics, and the fact that they are equal at the beginning means that there is no initial separation in rotation.

The nodal coordinates will evolve with the deformation of the sheets, and consequently the rivet points, directed normals and two orthonormal sets will evolve independently. The incremental update of the normals and sets is determined by estimating the angle increment $\Delta\varphi_*$ from

$$\Delta x_*^i - \Delta c_* = \Delta\varphi_* \times r_*^i = -R_*^i \Delta\varphi_* \quad (5.4)$$

where we introduced R_*^i to denote the specific cross product operator of

$$r_*^i = x_*^i - \frac{1}{2} \Delta x_*^i - c_* + \frac{1}{2} \Delta c_* \quad (5.5)$$

and an arbitrary vector. Here r_*^i vector from the center to node i at the mid step. This is a least squares problem for which a solution is

$$\Delta\varphi_* = A^{-1}b \quad (5.6)$$

with

$$A = w_*^i (R_*^i)^T R_*^i \quad (5.7)$$

$$b = w_*^i R_*^i (\Delta x_*^i - \Delta c_*). \quad (5.8)$$

From $\Delta\boldsymbol{\varphi}_*$ we compute the finite rotation matrix $\Delta\boldsymbol{\Phi}_*$ and update the orthonormal triad from the previous (here denoted by superscript 0) to the next step as

$$\mathbf{E}_* = \Delta\boldsymbol{\Phi}_* \mathbf{E}_*^0. \quad (5.9)$$

The update of the directed normal \mathbf{m}_* of each side is treated the same, and it follows that the rivet points evolve independently on each side and are given as

$$\mathbf{p}_* = \mathbf{c}_* + \Delta_* \mathbf{m}_*, \quad (5.10)$$

where we emphasize that Δ_* remains constant throughout the simulation. At any given configuration, we define the average normal of the rivet as

$$\mathbf{n} = \frac{\mathbf{e}_A + \mathbf{e}_B}{\|\mathbf{e}_A + \mathbf{e}_B\|} \quad (5.11)$$

and the generalized strains as

Tension	:	$\varepsilon_{\text{ten}} = \mathbf{n}^T (\mathbf{p}_B - \mathbf{p}_A)$	(length)
Shear	:	$\varepsilon_{\text{shr}} = (\mathbf{I} - \mathbf{n}\mathbf{n}^T) (\mathbf{p}_B - \mathbf{p}_A)$	(length)
Bending	:	$\varepsilon_{\text{ben}} = \mathbf{e}_A \times \mathbf{e}_B$	(sine angle)
Torsion	:	$\varepsilon_{\text{tor}} = \frac{1}{2} \mathbf{n}^T (\mathbf{f}_A \times \mathbf{f}_B + \mathbf{g}_A \times \mathbf{g}_B)$	(sine angle)

There is no preferred direction in the plane of the sheets, i.e., no planar anisotropy of the rivet, so there is no need to further specialize the shear or bending strains. The basis vectors \mathbf{f} and \mathbf{g} can therefore be determined arbitrarily at $t = 0$, still the torsional strain is invariant to whatever choice is made.

The elastic constitutive law computes the kinetic quantities as

Tension	:	$\tilde{\sigma}_{\text{ten}} = \kappa_{\text{ten}} \varepsilon_{\text{ten}}$	(force)
Shear	:	$\tilde{\sigma}_{\text{shr}} = \kappa_{\text{shr}} \varepsilon_{\text{shr}}$	(force)
Bending	:	$\tilde{\sigma}_{\text{ben}} = \kappa_{\text{ben}} \varepsilon_{\text{ben}}$	(torque)
Torsion	:	$\tilde{\sigma}_{\text{tor}} = \kappa_{\text{tor}} \varepsilon_{\text{tor}}$	(torque)

which, after plasticity and damage treatment results in σ_{ten} , σ_{shr} , σ_{ben} and σ_{tor} .

The nodal forces are given by the virtual work principle

$$\mathbf{f}_A^i \delta \mathbf{x}_A^i + \mathbf{f}_B^i \delta \mathbf{x}_B^i = \sigma_{\text{ten}} \delta \varepsilon_{\text{ten}} + \sigma_{\text{shr}} \delta \varepsilon_{\text{shr}} + \sigma_{\text{ben}} \delta \varepsilon_{\text{ben}} + \sigma_{\text{tor}} \delta \varepsilon_{\text{tor}} \quad (5.12)$$

where δ is the variation operator.

6

Cohesive Elements

The cohesive elements are used for modelling cohesive interfaces between faces of solid elements (types 19 and 21), faces of shell elements (types 20 and 22) and edges of shell elements (type ± 29), typically for treating delamination. Types 19, 20, 21 and 22 can be accompanied with a membrane treatment that essentially converts them into gasket elements.

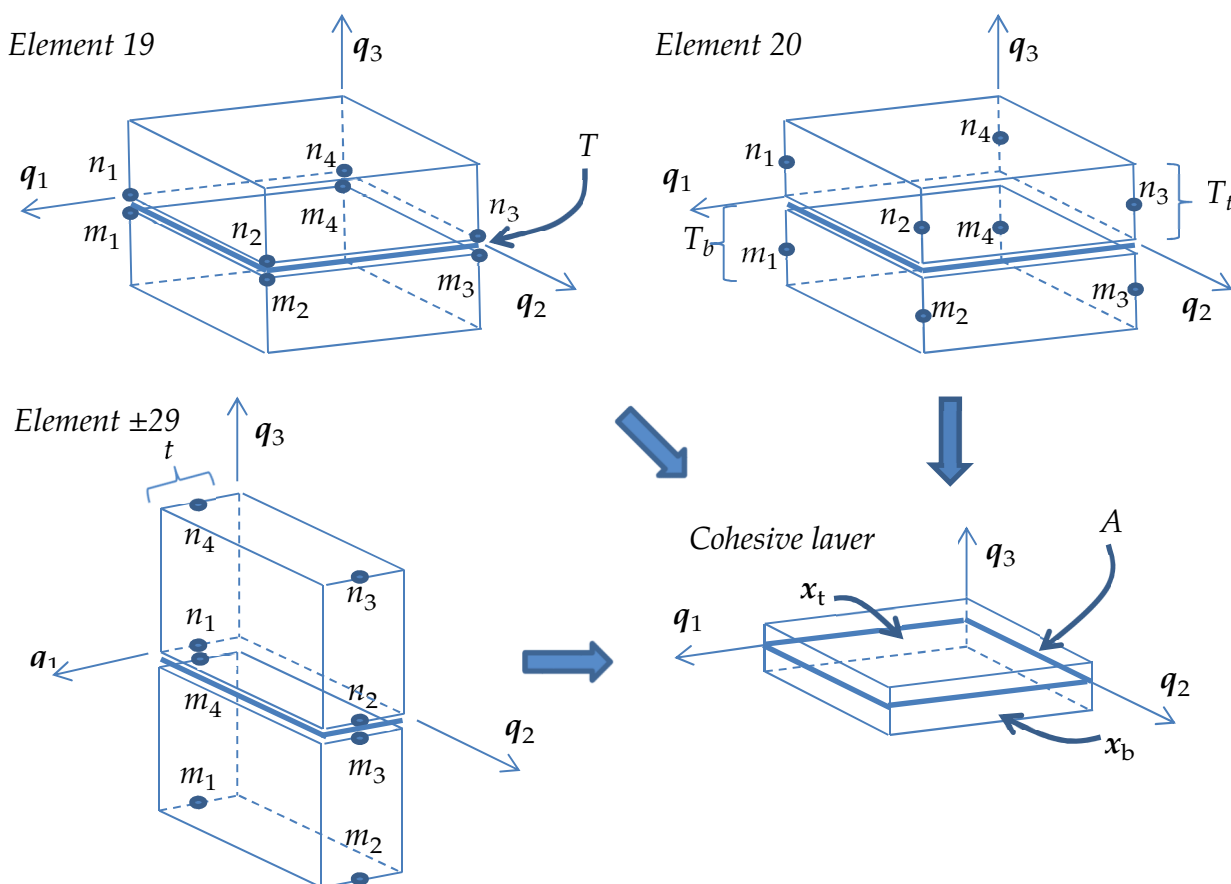


Figure 6.1. Illustration of cohesive elements 19, 20 and ± 29

As a comparison to other elements, the cohesive “strain” is the separation distance (length) in the cohesive layer, and the cohesive stress (force per area) is its conjugate with respect to the energy surface density (energy per area). Cohesive elements 21 and 22 are pentahedral versions of elements 19 and 20, respectively, where the top and bottom surface are triangles. These elements are not treated here per se, but the only difference is that the iso-parametric interpolation functions change.

6.1 Kinematics

For this presentation we refer to [Figure 6.1](#) for an illustration. Let

$$\mathbf{d} = \mathbf{Q}^T (\mathbf{x}_t - \mathbf{x}_b), \quad (6.1)$$

be the separation of the cohesive layer in the local system, where

$$\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \mathbf{q}_3], \quad (6.2)$$

is the local coordinate system and \mathbf{x}_t and \mathbf{x}_b are global coordinates initially located in the middle of the cohesive layer for a given iso-parametric coordinate (ξ, η) . As will be shown below, these coordinates are chosen so that $\mathbf{d} = \mathbf{0}$ at the beginning of the simulation.

The isoparametric shape functions are given by

$$N^i(\xi, \eta) = \frac{1}{4} (1 + \xi^i \xi) (1 + \eta^i \eta) \quad (6.3)$$

for $i = 1, \dots, 4$, and

$$\begin{aligned} \xi^* &= [-1, 1, 1, -1], \\ \eta^* &= [-1, -1, 1, 1]. \end{aligned} \quad (6.4)$$

From here and onwards, sub/superscripts n and m denote the top and bottom surfaces, respectively, and thus \mathbf{x}_i^n and \mathbf{x}_i^m are the nodal coordinates associated with the two surfaces.

For cohesive elements 19 and 20 we have (sum over i)

$$\begin{aligned} \mathbf{x}_t &= \left\{ \mathbf{x}_i^n - \frac{t_t}{2} \mathbf{n}_i^n \right\} N^i(\xi, \eta), \\ \mathbf{x}_b &= \left\{ \mathbf{x}_i^m + \frac{t_b}{2} \mathbf{n}_i^m \right\} N^i(\xi, \eta), \end{aligned} \quad (6.5)$$

where the initial distances between the top/bottom surfaces and middle of the cohesive layer are denoted t_t and t_b , respectively. Initially, $\mathbf{n}_i^n = \mathbf{n}_i^m$ are unit vectors so that

$$\mathbf{x}_i^n - \frac{t_t}{2} \mathbf{n}_i^n = \mathbf{x}_i^m + \frac{t_b}{2} \mathbf{n}_i^m \quad (6.6)$$

which ensures that $\mathbf{d} = \mathbf{0}$ at the beginning, but they may evolve independently with time. For cohesive element 19 we have $T = t_b = t_t$, which roughly corresponds to the distance between the top and bottom surface, while for element 20 they may be chosen different in case the cohesive layer is between two shells of different thicknesses. We introduce the parameter $\alpha \in]-1,1[$ that should be selected so that

$$\alpha = \frac{T_b - T_t}{T_b + T_t} = \frac{t_b - t_t}{t_b + t_t} \quad (6.7)$$

where T_b and T_t are the bottom and top shell thicknesses. This is the relative location of the cohesive layer with respect to the bottom and top surfaces.

The evolution of unit vectors differ between element 19 and 20, for the latter they evolve with the nodal rotational velocities ω_i^m and ω_i^n as

$$\begin{aligned} \dot{\mathbf{n}}_i^n &= \omega_i^n \times \mathbf{n}_i^n \\ \dot{\mathbf{n}}_i^m &= \omega_i^m \times \mathbf{n}_i^m. \end{aligned} \quad (6.8)$$

For element 19, which does not possess rotational degrees of freedom, we instead enforce the vectors to be constant in the top and bottom local systems \mathbf{Q}_n and \mathbf{Q}_m , respectively. This leads us to

$$\begin{aligned} \dot{\mathbf{n}}_i^n &= \dot{\mathbf{Q}}_n \mathbf{Q}_n^T \mathbf{n}_i^n \\ \dot{\mathbf{n}}_i^m &= \dot{\mathbf{Q}}_m \mathbf{Q}_m^T \mathbf{n}_i^m. \end{aligned} \quad (6.9)$$

The local systems \mathbf{Q}_n and \mathbf{Q}_m are calculated in exact analogue with \mathbf{Q} as described further down.

For cohesive element ± 29 we instead have

$$\begin{aligned} \dot{\mathbf{x}}_t &= \left\{ \dot{\mathbf{x}}_1^n + \zeta \frac{t}{2} \omega_1^n \times \mathbf{n}_t \right\} \frac{1-\eta}{2} + \left\{ \dot{\mathbf{x}}_2^n + \zeta \frac{t}{2} \omega_2^n \times \mathbf{n}_t \right\} \frac{1+\eta}{2}, \\ \dot{\mathbf{x}}_b &= \left\{ \dot{\mathbf{x}}_4^m + \zeta \frac{t}{2} \omega_4^m \times \mathbf{n}_b \right\} \frac{1-\eta}{2} + \left\{ \dot{\mathbf{x}}_3^m + \zeta \frac{t}{2} \omega_3^m \times \mathbf{n}_b \right\} \frac{1+\eta}{2}. \end{aligned} \quad (6.10)$$

Here we note that for evaluation the velocities of \mathbf{x}_t and \mathbf{x}_b we assume that the fiber pointing from assumed mid layer coincides with that of the coordinate axes. This is in analogy to how the Belytschko-Tsay element is treating the fiber vectors and presumably enhances robustness of the elements. Note also that the shell normal are in this case initially equal to \mathbf{q}_1 .

For the local coordinate system, cohesive elements 19 and 20 evaluate this according to the invariant node numbering approach for shells using the current location of the mid cohesive layer node coordinates

$$\bar{\mathbf{x}}_i = \frac{(1+\alpha)\mathbf{x}_i^n + (1-\alpha)\mathbf{x}_i^m}{2}, i = 1, \dots, 4, \quad (6.11)$$

as follows. First let

$$\begin{aligned} e_1 &= \frac{\bar{x}_3 - \bar{x}_1}{|\bar{x}_3 - \bar{x}_1|}, \\ e_2 &= \frac{\bar{x}_4 - \bar{x}_2}{|\bar{x}_4 - \bar{x}_2|}, \end{aligned} \quad (6.12)$$

and then

$$\begin{aligned} q_1 &= -\frac{e_1 + e_2}{|e_1 + e_2|}, \\ q_2 &= \frac{e_1 - e_2}{|e_1 - e_2|}, \end{aligned} \quad (6.13)$$

followed by

$$q_3 = q_1 \times q_2. \quad (6.14)$$

Cohesive element +29 starts by computing

$$q_2 = \frac{x_2^n + x_3^m - x_1^n - x_4^m}{|x_2^n + x_3^m - x_1^n - x_4^m|}, \quad (6.15)$$

followed by

$$q = x_4^n + x_3^n - x_1^m - x_2^m, \quad (6.16)$$

$$q_3 = \frac{q - q_2 q^T q_2}{|q - q_2 q^T q_2|}, \quad (6.17)$$

and

$$q_1 = q_2 \times q_3. \quad (6.18)$$

Cohesive element -29 starts by computing

$$q_2 = \frac{x_2^n + x_3^m - x_1^n - x_4^m}{|x_2^n + x_3^m - x_1^n - x_4^m|}, \quad (6.19)$$

followed by

$$q = \frac{1}{2} \left(\frac{(x_3^n - x_1^n) \times (x_4^n - x_2^n)}{|(x_3^n - x_1^n) \times (x_4^n - x_2^n)|} + \frac{(x_3^m - x_1^m) \times (x_4^m - x_2^m)}{|(x_3^m - x_1^m) \times (x_4^m - x_2^m)|} \right), \quad (6.20)$$

$$q_1 = \frac{q - q_2 q^T q_2}{|q - q_2 q^T q_2|}, \quad (6.21)$$

and

$$q_3 = q_1 \times q_2. \quad (6.22)$$

Thus, in pure out-of-plane shear, type +29 will initially have pure tangential traction in the q_1 -direction, that turns into a normal traction in the q_3 -direction, as the separation increases. Element type -29, however, will only have tangential traction in this case.

6.2 Constitutive law

The cohesive constitutive law amounts to determine the normal and shear stress, expressed here as the stress vector σ , as function of the separation vector d , $\sigma = \sigma(d, \dots)$. The typical appearance of each component of this vector is illustrated in [Figure 6.2.](#), the interface behaves elastically up to a critical separation distance d_e and peak stress σ_e after which damage commences. The interface is damaged and failure occur at a certain critical separation distance d_c , the unloading is typically elastic as indicated by the dashed arrow.

For detailed information on individual cohesive constitutive laws we refer to the materials section.

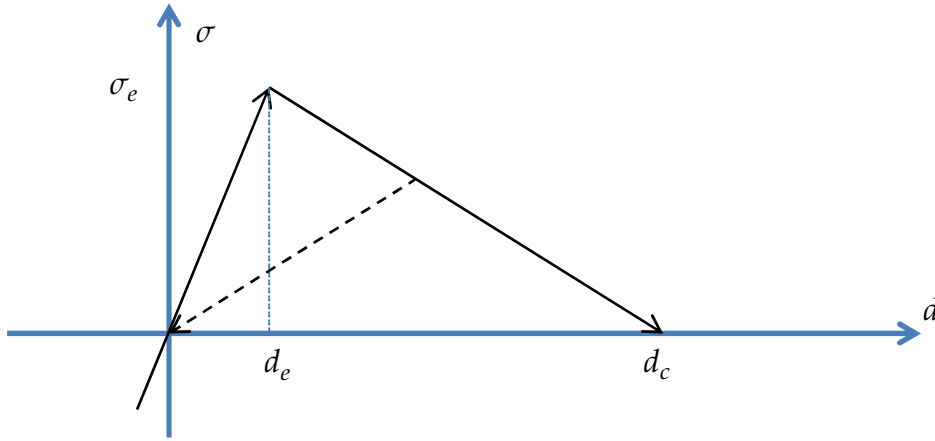


Figure 6.2. Common stress versus separation for a cohesive interface

6.3 Nodal forces

The principle of virtual work states that (sum over i)

$$\int_A \dot{\mathbf{d}}^T \sigma dA = \{\dot{\mathbf{x}}_i^m\}^T \mathbf{f}_i^m + \{\dot{\mathbf{x}}_i^n\}^T \mathbf{f}_i^n + \{\dot{\boldsymbol{\omega}}_i^m\}^T \mathbf{r}_i^m + \{\dot{\boldsymbol{\omega}}_i^n\}^T \mathbf{r}_i^n, \quad (6.23)$$

where \mathbf{f}_i^j and \mathbf{r}_i^j is the nodal force and moment for node i on element j , respectively. The area A represents the cohesive mid layer spanned by the iso-parametric representation and this is used to identify the nodal forces and moments. In the following we work out the details for cohesive element ± 29 . Cohesive elements 19 and 20 are treated analogous. Using [Equation Error! Reference source not found.](#) we also have

$$\int_A \dot{\mathbf{d}}^T \sigma dA = \int_A \dot{\mathbf{x}}_i^T \mathbf{Q} \sigma dA - \int_A \dot{\mathbf{x}}_b^T \mathbf{Q} \sigma dA - \int_A \dot{\mathbf{d}}^T \dot{\mathbf{Q}}^T \mathbf{Q} \sigma dA, \quad (6.24)$$

and before continuing we rewrite [\(6.10\)](#) as

$$\begin{aligned}
\dot{\mathbf{x}}_t^T &= \{\dot{\mathbf{x}}_1^n\}^T \frac{1-\eta}{2} - \{\omega_1^n\}^T \mathbf{R}_t^T \frac{1-\eta}{2} \xi \frac{t}{2} + \{\dot{\mathbf{x}}_2^n\}^T \frac{1+\eta}{2} - \{\omega_2^n\}^T \mathbf{R}_t^T \frac{1+\eta}{2} \xi \frac{t}{2}, \\
\dot{\mathbf{x}}_b^T &= \{\dot{\mathbf{x}}_4^m\}^T \frac{1-\eta}{2} - \{\omega_4^m\}^T \mathbf{R}_b^T \frac{1-\eta}{2} \xi \frac{t}{2} + \{\dot{\mathbf{x}}_3^m\}^T \frac{1+\eta}{2} - \{\omega_3^m\}^T \mathbf{R}_b^T \frac{1+\eta}{2} \xi \frac{t}{2},
\end{aligned} \tag{6.25}$$

where \mathbf{R}_* is the linear operator defined by

$$\mathbf{R}_* \boldsymbol{\omega} = \mathbf{n}_* \times \boldsymbol{\omega} \tag{6.26}$$

and insert this into the right of (6.24) which after some simplifications gives

$$\begin{aligned}
\int_A \mathbf{d}^T \boldsymbol{\sigma} dA &= - \int_A \mathbf{d}^T \dot{\mathbf{Q}}^T \mathbf{Q} \boldsymbol{\sigma} dA \\
&+ \{\dot{\mathbf{x}}_1^n\}^T \int_A \left(\frac{1-\eta}{2} \right) \mathbf{Q} \boldsymbol{\sigma} dA + \{\omega_1^n\}^T \int_A \left(-\frac{1-\eta}{2} \xi \frac{t}{2} \right) \mathbf{R}_t^T \mathbf{Q} \boldsymbol{\sigma} dA \\
&+ \{\dot{\mathbf{x}}_2^n\}^T \int_A \left(\frac{1+\eta}{2} \right) \mathbf{Q} \boldsymbol{\sigma} dA + \{\omega_2^n\}^T \int_A \left(-\frac{1+\eta}{2} \xi \frac{t}{2} \right) \mathbf{R}_t^T \mathbf{Q} \boldsymbol{\sigma} dA \\
&+ \{\dot{\mathbf{x}}_4^m\}^T \int_A \left(-\frac{1-\eta}{2} \right) \mathbf{Q} \boldsymbol{\sigma} dA + \{\omega_4^m\}^T \int_A \left(\frac{1-\eta}{2} \xi \frac{t}{2} \right) \mathbf{R}_b^T \mathbf{Q} \boldsymbol{\sigma} dA \\
&+ \{\dot{\mathbf{x}}_3^m\}^T \int_A \left(-\frac{1+\eta}{2} \right) \mathbf{Q} \boldsymbol{\sigma} dA + \{\omega_3^m\}^T \int_A \left(\frac{1+\eta}{2} \xi \frac{t}{2} \right) \mathbf{R}_b^T \mathbf{Q} \boldsymbol{\sigma} dA.
\end{aligned} \tag{6.27}$$

If the first term on the right hand side of (6.27) is neglected we can, using (6.23) and (6.27), identify the nonzero nodal forces and moments

$$\begin{aligned}
f_1^n &= \int_A \left(\frac{1-\eta}{2} \right) \mathbf{Q} \boldsymbol{\sigma} dA, \mathbf{r}_1^n = -\frac{t}{2} \int_A \left(\frac{1-\eta}{2} \xi \right) \mathbf{R}_t^T \mathbf{Q} \boldsymbol{\sigma} dA, \\
f_2^n &= \int_A \left(\frac{1+\eta}{2} \right) \mathbf{Q} \boldsymbol{\sigma} dA, \mathbf{r}_2^n = -\frac{t}{2} \int_A \left(\frac{1+\eta}{2} \xi \right) \mathbf{R}_t^T \mathbf{Q} \boldsymbol{\sigma} dA, \\
f_3^m &= -f_2^n, \mathbf{r}_3^m = \frac{t}{2} \int_A \left(\frac{1+\eta}{2} \xi \right) \mathbf{R}_b^T \mathbf{Q} \boldsymbol{\sigma} dA, \\
f_4^m &= -f_1^n, \mathbf{r}_4^m = \frac{t}{2} \int_A \left(\frac{1-\eta}{2} \xi \right) \mathbf{R}_b^T \mathbf{Q} \boldsymbol{\sigma} dA.
\end{aligned} \tag{6.28}$$

In the implementation, the force integrals are evaluated using either Newton-Cotes or Gaussian 4-point quadrature, where the integration point locations are given by either

$$\begin{aligned}
\xi_* &= [-1, 1, 1, -1], \\
\eta_* &= [-1, -1, 1, 1],
\end{aligned} \tag{6.29}$$

or

$$\begin{aligned}\xi_* &= \left[-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right], \\ \eta_* &= \left[-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right].\end{aligned}\tag{6.30}$$

The choice of quadrature is controlled by the value of INTFAIL on the cohesive material card. However, for cohesive elements ± 29 , Newton-Cotes quadrature is always used.

Thus, a force integral is evaluated as

$$\int_A \phi(\xi, \eta) \sigma dA \approx \sum_{i=1}^4 \phi(\xi_i, \eta_i) \sigma_i A_i,\tag{6.31}$$

where ϕ is an arbitrary function of the iso-parametric coordinates, A_i in the right-hand side stands for the area of the cohesive layer and σ_i is the cohesive interface stress, both evaluated at and with respect to integration point i .

6.4 Drilling constraint in shell ± 29

From (6.10) and the picture of shell type ± 29 in Figure 6.1, we see that rotational velocities with respect to the adjacent shell normals will not induce translational velocities in the integration points, so a stabilization scheme is applied. To this end we introduce the generalized drilling strains δ_i^n and δ_j^m , $i = 1, 2, j = 3, 4$, distances that are incremented by their respective velocities

$$\begin{aligned}\delta_i^n &= \mathbf{n}_i^T \{ \omega_i^n d_{21} - \mathbf{R}_{21} (\dot{\mathbf{x}}_2^n - \dot{\mathbf{x}}_1^n) \}, i = 1, 2, \\ \delta_j^m &= \mathbf{n}_b^T \{ \omega_j^m d_{34} - \mathbf{R}_{34} (\dot{\mathbf{x}}_3^m - \dot{\mathbf{x}}_4^m) \}, j = 3, 4,\end{aligned}\tag{6.32}$$

where

$$\begin{aligned}d_{21} &= |\mathbf{x}_2^n - \mathbf{x}_1^n|, \\ d_{34} &= |\mathbf{x}_3^m - \mathbf{x}_4^m|,\end{aligned}\tag{6.33}$$

and we make use of the following definitions for arbitrary vector \mathbf{v} ,

$$\begin{aligned}\mathbf{R}_{21}\mathbf{v} &= \frac{1}{d_{21}}(\mathbf{x}_2^n - \mathbf{x}_1^n) \times \mathbf{v}, \\ \mathbf{R}_{34}\mathbf{v} &= \frac{1}{d_{34}}(\mathbf{x}_3^m - \mathbf{x}_4^m) \times \mathbf{v}.\end{aligned}\tag{6.34}$$

A characteristic material stiffness E , typically a fraction of the elastic stiffness of the underlying cohesive material, is used to set up the drilling stress

$$\begin{aligned}\zeta_i^n &= E\delta_i^n, \quad i = 1,2, \\ \zeta_j^m &= E\delta_j^m, \quad j = 3,4,\end{aligned}\tag{6.35}$$

and the stabilization nodal forces are evaluated as

$$\begin{aligned}\mathbf{f}_1^n &= A(\zeta_1^n + \zeta_2^n)\mathbf{R}_{21}^T\mathbf{n}_t, \mathbf{r}_1^n = A\zeta_1^n d_{21}\mathbf{n}_t, \\ \mathbf{f}_2^n &= -\mathbf{f}_1^n, \mathbf{r}_2^n = A\zeta_2^n d_{21}\mathbf{n}_t, \\ \mathbf{f}_3^m &= -\mathbf{f}_4^m, \mathbf{r}_3^m = A\zeta_3^m d_{34}\mathbf{n}_b, \\ \mathbf{f}_4^m &= A(\zeta_3^m + \zeta_4^m)\mathbf{R}_{34}^T\mathbf{n}_b, \mathbf{r}_4^m = A\zeta_4^m d_{34}\mathbf{n}_b,\end{aligned}\tag{6.36}$$

and these forces and moments are added to the structural ones in the previous section.

6.5 Rotational masses in shell ± 29

The rotational mass in shell ± 29 is determined from a simple energy criterion. Assuming that the cohesive layer is thin and the layer spins with a rotational velocity ω around axis q_2 , the kinetic energy from the nodal rotational masses m_r is

$$W = \frac{1}{2}(m_r + m_r + m_r + m_r)\omega^2 = 2m_r\omega^2.\tag{6.37}$$

This is compared to the corresponding kinetic energy for an equivalent solid type 19 cohesive layer, using that the shell type ± 29 translational nodal mass m_t is twice that of the solid type 19 nodal masses,

$$W = \frac{1}{2}\left(\frac{m_t}{2} + \frac{m_t}{2} + \frac{m_t}{2} + \frac{m_t}{2} + \frac{m_t}{2} + \frac{m_t}{2} + \frac{m_t}{2} + \frac{m_t}{2}\right)\left(\frac{t}{2}\omega\right)^2 = \frac{1}{2}m_t t^2 \omega^2,\tag{6.38}$$

which results in

$$m_r = \frac{1}{4}m_t t^2.\tag{6.39}$$

6.6 Stiffness matrix in shell ±29

The stiffness matrix is the sum of contributions from the constitutive law and the drilling force, and is for simplicity implemented for a force free configuration, i.e.,

$$\begin{aligned}\sigma &= \mathbf{0}, \\ \zeta &= \mathbf{0},\end{aligned}\tag{6.40}$$

where we have used the notation

$$\zeta = \begin{bmatrix} \zeta_1^n \\ \zeta_2^n \\ \zeta_3^m \\ \zeta_4^m \end{bmatrix}.\tag{6.41}$$

For a given integration point and referring to [Equation \(6.28\)](#) and [\(6.36\)](#), we collect these nodal force triplets into complete nodal vectors f and g , associated with the constitutive and drilling part, respectively. That is,

$$f = \begin{bmatrix} f_1^n \\ r_1^n \\ f_2^n \\ r_2^n \\ f_3^m \\ r_3^m \\ f_4^m \\ r_4^m \end{bmatrix},\tag{6.42}$$

and the same expression holds for g . Likewise we let v be the collection of nodal velocities, i.e.,

$$v = \begin{bmatrix} \dot{x}_1^n \\ \omega_1^n \\ \dot{x}_2^n \\ \omega_2^n \\ \dot{x}_3^m \\ \omega_3^m \\ \dot{x}_4^m \\ \omega_4^m \end{bmatrix},\tag{6.43}$$

and note that we can identify generalized strain-displacement matrices B_f (3 by 24 matrix) and B_g (4 by 24 matrix) from [bm2224](#) and [\(6.10\)](#) as well as [\(6.32\)](#) so that

$$\begin{aligned}\dot{d} &= B_f v, \\ \dot{\delta} &= B_g v,\end{aligned}\tag{6.44}$$

where we have collected the drilling kinematic velocities in a vector

$$\dot{\delta} = \begin{bmatrix} \dot{\delta}_1^n \\ \dot{\delta}_2^n \\ \dot{\delta}_3^m \\ \dot{\delta}_4^m \end{bmatrix} \quad (6.45)$$

With this notation we can rewrite (6.28) and (6.36) into a more compact form

$$\begin{aligned} \mathbf{f} &= \mathbf{B}_f^T \boldsymbol{\sigma} A, \\ \mathbf{g} &= \mathbf{B}_g^T \boldsymbol{\zeta} A, \end{aligned} \quad (6.46)$$

with A here being the area of the integration point of interest. The stiffness matrix is then simply the differentiation of these force vectors with respect to the nodal coordinates, and by using (6.35), (6.40) and the chain rule of differentiation we get

$$\begin{aligned} \mathbf{K}_f &= \mathbf{B}_f^T \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{d}} \mathbf{B}_f A, \\ \mathbf{K}_g &= E \mathbf{B}_g^T \mathbf{B}_g A, \end{aligned} \quad (6.47)$$

where $\partial \boldsymbol{\sigma} / \partial \mathbf{d}$ is the constitutive tangent from the cohesive material used.

7

Belytschko Beam

The Belytschko beam element formulation [Belytschko et al. 1977] is part of a family of structural finite elements, by Belytschko and other researchers that employ a ‘co-rotational technique’ in the element formulation for treating large rotation. This section discusses the co-rotational formulation, since the formulation is most easily described for a beam element, and then describes the beam theory used to formulate the co-rotational beam element.

7.1 Co-rotational Technique

In any large displacement formulation, the goal is to separate the deformation displacements from the rigid body displacements, as only the deformation displacements give rise to strains and the associated generation of strain energy. This separation is usually accomplished by comparing the current configuration with a reference configuration.

The current configuration is a complete description of the deformed body in its current spatial location and orientation, giving locations of all points (nodes) comprising the body. The reference configuration can be either the initial configuration of the body, i.e., nodal locations at time zero, or the configuration of the body at some other state (time). Often the reference configuration is chosen to be the previous configuration, say at time $t^n = t^{n+1} - \Delta t$.

The choice of the reference configuration determines the type of deformations that will be computed: total deformations result from comparing the current configuration with the initial configuration, while incremental deformations result from comparing with the previous configuration. In most time stepping (numerical) Lagrangian formulations, incremental deformations are used because they result in significant simplifications of other algorithms, chiefly constitutive models.

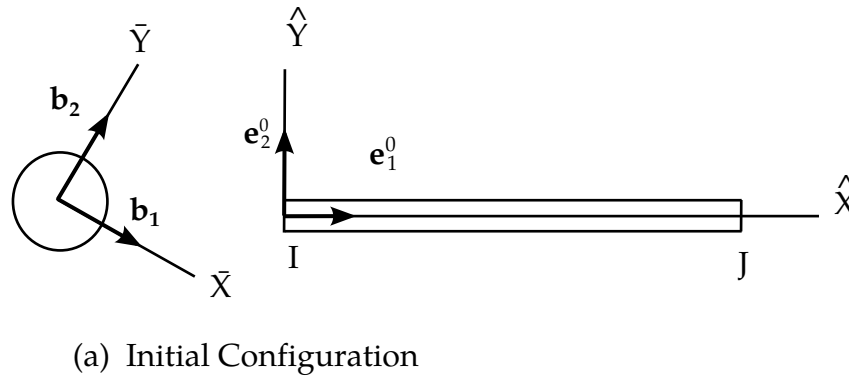


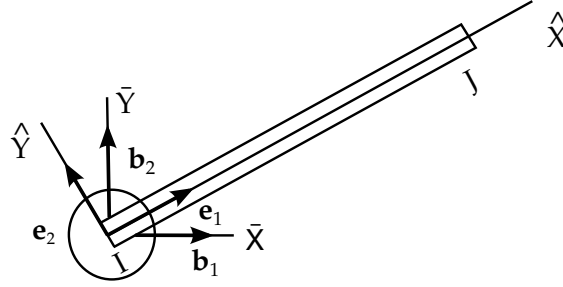
Figure 7.1. Co-rotational coordinate system: (a) initial configuration, (b) rigid rotational configuration and (c) deformed configuration.

A direct comparison of the current configuration with the reference configuration does not result in a determination of the deformation, but rather provides the total (or incremental) displacements. We will use the unqualified term displacements to mean either the total displacements or the incremental displacements, depending on the choice of the reference configuration as the initial or the last state. This is perhaps most obvious if the reference configuration is the initial configuration. The direct comparison of the current configuration with the reference configuration yields displacements, which contain components due to deformations and rigid body motions. The task remains of separating the deformation and rigid body displacements. The deformations are usually found by subtracting from the displacements an estimate of the rigid body displacements. Exact rigid body displacements are usually only known for trivial cases where they are prescribed a priori as part of a displacement field. The co-rotational formulations provide one such estimate of the rigid body displacements.

The co-rotational formulation uses two types of coordinate systems: one system associated with each element, i.e., element coordinates which deform with the element, and another associated with each node, i.e., body coordinates embedded in the nodes. (The term ‘body’ is used to avoid possible confusion from referring to these coordinates as ‘nodal’ coordinates. Also, in the more general formulation presented in [Belytschko et al., 1977], the nodes could optionally be attached to rigid bodies. Thus the term ‘body coordinates’ refers to a system of coordinates in a rigid body, of which a node is a special case.) These two coordinate systems are shown in the upper portion of [Figure 7.1](#).

The element coordinate system is defined to have the local x-axis \hat{x} originating at node I and terminating at node J ; the local y-axis \hat{y} and, in three dimension, the local z-axis \hat{z} , are constructed normal to \hat{x} . The element coordinate system $(\hat{x}, \hat{y}, \hat{z})$ and associated unit vector triad $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ are updated at every time step by the same technique used to construct the initial system; thus the unit vector \mathbf{e}_1 deforms with the element since it always points from node I to node J .

The embedded body coordinate system is initially oriented along the principal inertial axes; either the assembled nodal mass or associated rigid body inertial tensor is



(b) Rigid Rotation Configuration

Figure 7.2. Co-rotational coordinate system: (a) initial configuration, (b) rigid rotational configuration and (c) deformed configuration.

used in determining the inertial principal values and directions. Although the initial orientation of the body axes is arbitrary, the selection of a principal inertia coordinate system simplifies the rotational equations of motion, i.e., no inertial cross product terms are present in the rotational equations of motion. Because the body coordinates are fixed in the node, or rigid body, they rotate and translate with the node and are updated by integrating the rotational equations of motion, as will be described subsequently.

The unit vectors of the two coordinate systems define rotational transformations between the global coordinate system and each respective coordinate system. These transformations operate on vectors with global components $\mathbf{A} = (A_x, A_y, A_z)$, body coordinates components $\bar{\mathbf{A}} = (\bar{A}_x, \bar{A}_y, \bar{A}_z)$, and element coordinate components $\hat{\mathbf{A}} = (\hat{A}_x, \hat{A}_y, \hat{A}_z)$ which are defined as:

$$\mathbf{A} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{bmatrix} b_{1x} & b_{2x} & b_{3x} \\ b_{1y} & b_{2y} & b_{3y} \\ b_{1z} & b_{2z} & b_{3z} \end{bmatrix} \begin{Bmatrix} \bar{A}_x \\ \bar{A}_y \\ \bar{A}_z \end{Bmatrix} = [\lambda] \{\bar{\mathbf{A}}\}, \quad (7.1)$$

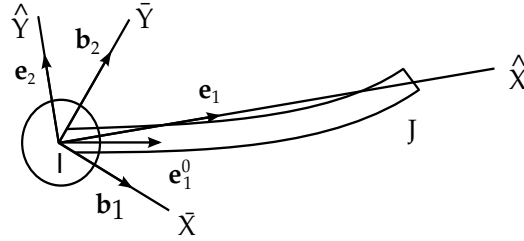
where b_{ix}, b_{iy}, b_{iz} are the global components of the body coordinate unit vectors. Similarly for the element coordinate system:

$$\mathbf{A} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = [\mu] \{\hat{\mathbf{A}}\}, \quad (7.2)$$

where e_{ix}, e_{iy}, e_{iz} are the global components of the element coordinate unit vectors. The inverse transformations are defined by the matrix transpose, i.e.,

$$\begin{aligned} \{\bar{\mathbf{A}}\} &= [\lambda]^T \{\mathbf{A}\} \\ \{\hat{\mathbf{A}}\} &= [\mu]^T \{\mathbf{A}\}, \end{aligned} \quad (7.3)$$

since these are proper rotational transformations.



(c) Deformed Configuration

Figure 7.3. Co-rotational coordinate system: (a) initial configuration, (b) rigid rotational configuration and (c) deformed configuration.

The following two examples illustrate how the element and body coordinate system are used to separate the deformations and rigid body displacements from the displacements:

Rigid Rotation. First, consider a rigid body rotation of the beam element about node I , as shown in the center of Figure 7.2, i.e., consider node I to be a pinned connection. Because the beam does not deform during the rigid rotation, the orientation of the unit vector \mathbf{e}_1 in the initial and rotated configuration will be the same with respect to the body coordinates. If the body coordinate components of the initial element unit vector \mathbf{e}_1^0 were stored, they would be identical to the body coordinate components of the current element unit vector \mathbf{e}_1 .

Deformation Rotation. Next, consider node I to be constrained against rotation, i.e., a clamped connection. Now node J is moved, as shown in the lower portion of Figure 7.3, causing the beam element to deform. The updated element unit vector \mathbf{e}_1 is constructed and its body coordinate components are compared to the body coordinate components of the original element unit vector \mathbf{e}_1^0 . Because the body coordinate system did not rotate, as node I was constrained, the original element unit vector and the current element unit vector are not colinear. Indeed, the angle between these two unit vectors is the amount of rotational deformation at node I , i.e.,

$$\mathbf{e}_1 \times \mathbf{e}_1^0 = \theta_\ell \mathbf{e}_3. \quad (7.4)$$

Thus the co-rotational formulation separates the deformation and rigid body deformations by using:

- a coordinate system that deforms with the element, i.e., the element coordinates;
- or a coordinate system that rigidly rotates with the nodes, i.e., the body coordinates;

Then it compares the current orientation of the element coordinate system with the initial element coordinate system, using the rigidly rotated body coordinate system, to determine the deformations.

7.2 Belytschko Beam Element Formulation

The deformation displacements used in the Belytschko beam element formulation are:

$$\hat{\mathbf{d}}^T = \{\delta_{IJ}, \hat{\theta}_{xJI}, \hat{\theta}_{yI}, \hat{\theta}_{yJ}, \hat{\theta}_{zI}, \hat{\theta}_{zJ}\}, \quad (7.5)$$

where,

$$\begin{aligned} \delta_{IJ} &= \text{length change} \\ \hat{\theta}_{xJI} &= \text{torsional deformation} \\ \hat{\theta}_{yI}, \hat{\theta}_{yJ}, \hat{\theta}_{zI}, \hat{\theta}_{zJ} &= \text{bending rotation deformations} \end{aligned}$$

The superscript \wedge emphasizes that these quantities are defined in the local element coordinate system, and I and J are the nodes at the ends of the beam.

The beam deformations, defined above in [Equation \(7.5\)](#), are the usual small displacement beam deformations (see, for example, [Przemieniecki 1986]). Indeed, one advantage of the co-rotational formulation is the ease with which existing small displacement element formulations can be adapted to a large displacement formulation having small deformations in the element system. Small deformation theories can be easily accommodated because the definition of the local element coordinate system is independent of rigid body rotations and hence deformation displacement can be defined directly.

7.2.1 Calculation of Deformations

The elongation of the beam is calculated directly from the original nodal coordinates (X_I, Y_I, Z_I) and the total displacements (u_{xI}, u_{yI}, u_{zI}) :

$$\delta_{IJ} = \frac{1}{l + l^0} [2(X_{JI}u_{xJI} + Y_{JI}u_{yJI} + Z_{JI}u_{zJI}) + u_{xJI}^2 + u_{yJI}^2 + u_{zJI}^2], \quad (7.6)$$

where

$$\begin{aligned} X_{JI} &= X_J - X_I \\ u_{xJI} &= u_{xJ} - u_{xI}, \text{ etc.} \end{aligned} \quad (7.7)$$

The deformation rotations are calculated using the body coordinate components of the original element coordinate unit vector along the beam axis, i.e., \mathbf{e}_1^0 , as outlined in the previous section. Because the body coordinate components of initial unit vector \mathbf{e}_1^0 rotate with the node, in the deformed configuration it indicates the direction of the beam's

axis if no deformations had occurred. Thus comparing the initial unit vector \mathbf{e}_1^0 with its current orientation \mathbf{e}_1 indicates the magnitude of deformation rotations. Forming the vector cross product between \mathbf{e}_1^0 and \mathbf{e}_1 :

$$\mathbf{e}_1 \times \mathbf{e}_1^0 = \hat{\theta}_y \mathbf{e}_2 + \hat{\theta}_z \mathbf{e}_3, \quad (7.8)$$

where

$$\begin{aligned} \hat{\theta}_y &= \text{is the incremental deformation about the local } \hat{y} \text{ axis} \\ \hat{\theta}_z &= \text{is the incremental deformation about the local } \hat{z} \text{ axis} \end{aligned}$$

The calculation is most conveniently performed by transforming the body components of the initial element vector into the current element coordinate system:

$$\begin{Bmatrix} \hat{e}_{1x}^0 \\ \hat{e}_{1y}^0 \\ \hat{e}_{1z}^0 \end{Bmatrix} = [\boldsymbol{\mu}]^T [\boldsymbol{\lambda}] \begin{Bmatrix} \bar{e}_{1x}^0 \\ \bar{e}_{1y}^0 \\ \bar{e}_{1z}^0 \end{Bmatrix}. \quad (7.9)$$

Substituting the above into Equation (4.10)

$$\mathbf{e}_1 \times \mathbf{e}_1^0 = \det \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ 1 & 0 & 0 \\ \hat{e}_{1x}^0 & \hat{e}_{1y}^0 & \hat{e}_{1z}^0 \end{bmatrix} = -\hat{e}_{1z}^0 \mathbf{e}_2 + \hat{e}_{1y}^0 \mathbf{e}_3 = \hat{\theta}_y \mathbf{e}_2 + \hat{\theta}_z \mathbf{e}_3. \quad (7.10)$$

Thus,

$$\begin{aligned} \hat{\theta}_y &= -\hat{e}_{1z}^0 \\ \hat{\theta}_z &= \hat{e}_{1y}^0. \end{aligned} \quad (7.11)$$

The torsional deformation rotation is calculated from the vector cross product of initial unit vectors, from each node of the beam, that were normal to the axis of the beam, i.e., \hat{e}_{2I}^0 and \hat{e}_{2J}^0 ; note that \hat{e}_{3I}^0 and \hat{e}_{3J}^0 could also be used. The result from this vector cross product is then projected onto the current axis of the beam, i.e.,

$$\hat{\theta}_{xIJ} = \mathbf{e}_1 \cdot (\hat{\mathbf{e}}_{2I}^0 \times \hat{\mathbf{e}}_{2J}^0) = \mathbf{e}_1 \cdot \det \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ \hat{e}_{x2I}^0 & \hat{e}_{y2I}^0 & \hat{e}_{z2I}^0 \\ \hat{e}_{x2J}^0 & \hat{e}_{y2J}^0 & \hat{e}_{z2J}^0 \end{bmatrix} = \hat{e}_{y2I}^0 \hat{e}_{z2J}^0 - \hat{e}_{y2J}^0 \hat{e}_{z2I}^0. \quad (7.12)$$

Note that the body components of \bar{e}_{2I}^0 and \bar{e}_{2J}^0 are transformed into the current element coordinate system before performing the indicated vector products.

7.2.2 Calculation of Internal Forces

There are two methods for computing the internal forces for the Belytschko beam element formulation:

1. functional forms relating the overall response of the beam, e.g., moment-curvature relations,
2. direct through-the-thickness integration of the stress.

Currently only the former method, as explained subsequently, is implemented; the direct integration method is detailed in [Belytschko et al., 1977].

Axial Force. The internal axial force is calculated from the elongation of the beam δ , as given by Equation (7.6), and an axial stiffness:

$$\hat{f}_{xI} = K^a \delta, \quad (7.13)$$

where

$$K^a = \frac{AE}{l^0} = \text{is the axial stiffness}$$

A = cross sectional area of the beam
 E = Young's Modulus
 l^0 = original length of the beam

Bending Moments. The bending moments are related to the deformation rotations by

$$\begin{Bmatrix} \hat{m}_{yI} \\ \hat{m}_{yJ} \end{Bmatrix} = \frac{K_y^b}{1 + \phi_y} \begin{bmatrix} 4 + \phi_y & 2 - \phi_y \\ 2 - \phi_y & 4 + \phi_y \end{bmatrix} \begin{Bmatrix} \hat{\theta}_{yI} \\ \hat{\theta}_{yJ} \end{Bmatrix}, \quad (7.14a)$$

$$\begin{Bmatrix} \hat{m}_{zI} \\ \hat{m}_{zJ} \end{Bmatrix} = \frac{K_z^b}{1 + \phi_z} \begin{bmatrix} 4 + \phi_z & 2 - \phi_z \\ 2 - \phi_z & 4 + \phi_z \end{bmatrix} \begin{Bmatrix} \hat{\theta}_{zI} \\ \hat{\theta}_{zJ} \end{Bmatrix}, \quad (7.14b)$$

where Equation (7.14a) is for bending in the $\hat{x} - \hat{z}$ plane and Equation (7.14b) is for bending in the $\hat{x} - \hat{y}$ plane. The bending constants are given by

$$K_y^b = \frac{EI_{yy}}{l^0} \quad (7.15a)$$

$$K_z^b = \frac{EI_{zz}}{l^0} \quad (7.15b)$$

$$I_{yy} = \int \int \hat{z}^2 d\hat{y}d\hat{z} \quad (7.15c)$$

$$I_{zz} = \int \int \hat{y}^2 d\hat{y}d\hat{z} \quad (7.15d)$$

$$\phi_y = \frac{12EI_{yy}}{GA_s l^2} \quad (7.15e)$$

$$\phi_z = \frac{12EI_{zz}}{GA_s l^2}. \quad (7.15f)$$

Hence ϕ is the shear factor, G the shear modulus, and A_s is the effective area in shear.

Torsional Moment. The torsional moment is calculated from the torsional deformation rotation as

$$\hat{m}_{xJ} = K^t \hat{\theta}_{xJI}, \quad (7.16)$$

where

$$K^t = \frac{GJ}{l^0}, \quad (7.17)$$

and,

$$J = \int \int \hat{y} \hat{z} d\hat{y} d\hat{z}. \quad (7.18)$$

The above forces are conjugate to the deformation displacements given previously in Equation (7.5), i.e.,

$$\hat{\mathbf{d}}^T = \{\delta_{IJ}, \hat{\theta}_{xJI}, \hat{\theta}_{yJI}, \hat{\theta}_{zJI}, \hat{\theta}_{xIJ}, \hat{\theta}_{yIJ}, \hat{\theta}_{zIJ}\}, \quad (7.19)$$

where

$$\hat{\mathbf{d}}^T \hat{\mathbf{f}} = W^{\text{int}}. \quad (7.20)$$

And with

$$\hat{\mathbf{f}}^T = \{\hat{f}_{xJ}, \hat{m}_{xJ}, \hat{m}_{yI}, \hat{m}_{yJ}, \hat{m}_{zI}, \hat{m}_{zJ}\}. \quad (7.21)$$

The remaining internal force components are found from equilibrium:

$$\begin{aligned} \hat{f}_{xI} &= -\hat{f}_{xJ} & \hat{m}_{xI} &= -\hat{m}_{xJ} \\ \hat{f}_{zI} &= -\frac{\hat{m}_{yI} + \hat{m}_{yJ}}{l^0} & \hat{f}_{zJ} &= -\hat{f}_{zI} \\ \hat{f}_{yI} &= -\frac{\hat{m}_{zI} + \hat{m}_{zJ}}{l^0} & \hat{f}_{yJ} &= -\hat{f}_{yI} \end{aligned} \quad (7.22)$$

7.2.3 Updating the Body Coordinate Unit Vectors

The body coordinate unit vectors are updated using the Newmark β -Method [Newmark 1959] with $\beta = 0$, which is almost identical to the central difference method [Belytschko 1974]. In particular, the body component unit vectors are updated using the formula

$$\mathbf{b}_i^{j+1} = \mathbf{b}_i^j + \Delta t \frac{d\mathbf{b}_i^j}{dt} + \frac{\Delta t^2}{2} \frac{d^2\mathbf{b}_i^j}{dt^2}, \quad (7.23)$$

where the superscripts refer to the time step and the subscripts refer to the three unit vectors comprising the body coordinate triad. The time derivatives in the above equation are replaced by their equivalent forms from vector analysis:

$$\begin{aligned} \frac{d\mathbf{b}_i^j}{dt} &= \boldsymbol{\omega} \times \mathbf{b}_i \\ \frac{d^2\mathbf{b}_i^j}{dt^2} &= \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{b}_i) + (\dot{\boldsymbol{\omega}} \times \mathbf{b}_i), \end{aligned} \quad (7.24)$$

where ω and α are vectors of angular velocity and acceleration, respectively, obtained from the rotational equations of motion. With the above relations substituted into Equation (7.23), the update formula for the unit vectors becomes

$$\mathbf{b}_i^{j+1} = \mathbf{b}_i^j + \Delta t(\omega \times \mathbf{b}_i) + \frac{\Delta t^2}{2} \{[\omega \times (\omega \times \mathbf{b}_i) + (\alpha_i \times \mathbf{b}_i)]\}. \quad (7.25)$$

To obtain the formulation for the updated components of the unit vectors, the body coordinate system is temporarily considered to be fixed and then the dot product of Equation (7.25) is formed with the unit vector to be updated. For example, to update the \bar{x} component of \mathbf{b}_3 , the dot product of Equation (7.25), with $i = 3$, is formed with \mathbf{b}_1 , which can be simplified to the relation

$$\bar{b}_{x3}^{j+1} = \mathbf{b}_1^j \cdot \mathbf{b}_3^{j+1} = \Delta t \omega_y^j + \frac{\Delta t^2}{2} (\omega_x^j \omega_z^j + \alpha_y^j), \quad (7.26)$$

Similarly,

$$\begin{aligned} \bar{b}_{y3}^{j+1} &= \mathbf{b}_2^j \cdot \mathbf{b}_3^{j+1} = \Delta t \omega_x^j + \frac{\Delta t^2}{2} (\omega_y^j \omega_z^j + \alpha_x^j) \\ \bar{b}_{z3}^{j+1} &= \mathbf{b}_1^j \cdot \mathbf{b}_2^{j+1} = \Delta t \omega_z^j + \frac{\Delta t^2}{2} (\omega_x^j \omega_y^j + \alpha_z^j). \end{aligned} \quad (7.27)$$

The remaining components \mathbf{b}_3^{j+1} and \mathbf{b}_1^{j+1} are found by using normality and orthogonality, where it is assumed that the angular velocities ω are small during a time step so that the quadratic terms in the update relations can be ignored. Since \mathbf{b}_3^{j+1} is a unit vector, normality provides the relation

$$\bar{b}_{z3}^{j+1} = \sqrt{1 - (\bar{b}_{x3}^{j+1})^2 - (\bar{b}_{y3}^{j+1})^2}. \quad (7.28)$$

Next, if it is assumed that $\bar{b}_{x1}^{j+1} \approx 1$, orthogonality yields

$$\bar{b}_{z1}^{j+1} = -\frac{\bar{b}_{x3}^{j+1} + \bar{b}_{y1}^{j+1} \bar{b}_{y3}^{j+1}}{\bar{b}_{z3}^{j+1}}. \quad (7.29)$$

The component \bar{b}_{x1}^{j+1} is then found by enforcing normality:

$$\bar{b}_{x1}^{j+1} = \sqrt{1 - (\bar{b}_{y1}^{j+1})^2 - (\bar{b}_{z1}^{j+1})^2}. \quad (7.30)$$

The updated components of \mathbf{b}_1 and \mathbf{b}_3 are defined relative to the body coordinates at time step j . To complete the update and define the transformation matrix, Equation (7.1), at time step $j + 1$, the updated unit vectors \mathbf{b}_1 and \mathbf{b}_3 are transformed to the global coordinate system, using Equation (7.1) with $[\lambda]$ defined at step j , and their vector cross product is used to form \mathbf{b}_2 .

8

Hughes-Liu Beam

The Hughes-Liu beam element formulation, based on the shell [Hughes and Liu 1981a, 1981b] discussed later, was the first beam element we implemented. It has several desirable qualities:

- It is incrementally objective (rigid body rotations do not generate strains), allowing for the treatment of finite strains that occur in many practical applications;
- It is simple, which usually translates into computational efficiency and robustness
- It is compatible with the brick elements, because the element is based on a degenerated brick element formulation;
- It includes finite transverse shear strains. The added computations needed to retain this strain component, compare to those for the assumption of no transverse shear strain, are insignificant.

8.1 Geometry

The Hughes-Liu beam element is based on a degeneration of the isoparametric 8-node solid element, an approach originated by Ahmad et al., [1970]. Recall the solid element isoparametric mapping of the biunit cube

$$\mathbf{x}(\xi, \eta, \zeta) = \sum_{a=1}^8 N_a(\xi, \eta, \zeta) \mathbf{x}_a, \quad (8.1)$$

with,

$$N_a(\xi, \eta, \zeta) = \frac{(1 + \xi_a \xi)(1 + \eta_a \eta)(1 + \zeta_a \zeta)}{8}, \quad (8.2)$$

where \mathbf{x} is an arbitrary point in the element, (ξ, η, ζ) are the parametric coordinates, \mathbf{x}_a are the global nodal coordinates of node a , and N_a are the element shape functions evaluated at node a , i.e., (ξ_a, η_a, ζ_a) are (ξ, η, ζ) evaluated at node a .

In the beam geometry, ξ determines the location along the axis of the beam and the coordinate pair (η, ξ) defines a point on the cross section. To degenerate the 8-node brick geometry into the 2-node beam geometry, the four nodes at $\xi = -1$ and at $\xi = 1$ are combined into a single node with three translational and three rotational degrees of freedom. Orthogonal, inextensible nodal fibers are defined at each node for treating the rotational degrees of freedom. Figure 8.1 shows a schematic of the biunit cube and the beam element. The mapping of the biunit cube into the beam element is separated into three parts:

$$\begin{aligned} \mathbf{x}(\xi, \eta, \zeta) &= \bar{\mathbf{x}}(\xi) + \mathbf{X}(\xi, \eta, \zeta), \\ &= \bar{\mathbf{x}}(\xi) + \mathbf{X}_\zeta(\xi, \zeta) + \mathbf{X}_\eta(\xi, \eta), \end{aligned} \quad (8.3)$$

where $\bar{\mathbf{x}}$ denotes a position vector to a point on the reference axis of the beam, and \mathbf{X}_ζ and \mathbf{X}_η are position vectors at point $\bar{\mathbf{x}}$ on the axis that define the fiber directions through that point. In particular,

(8.4)

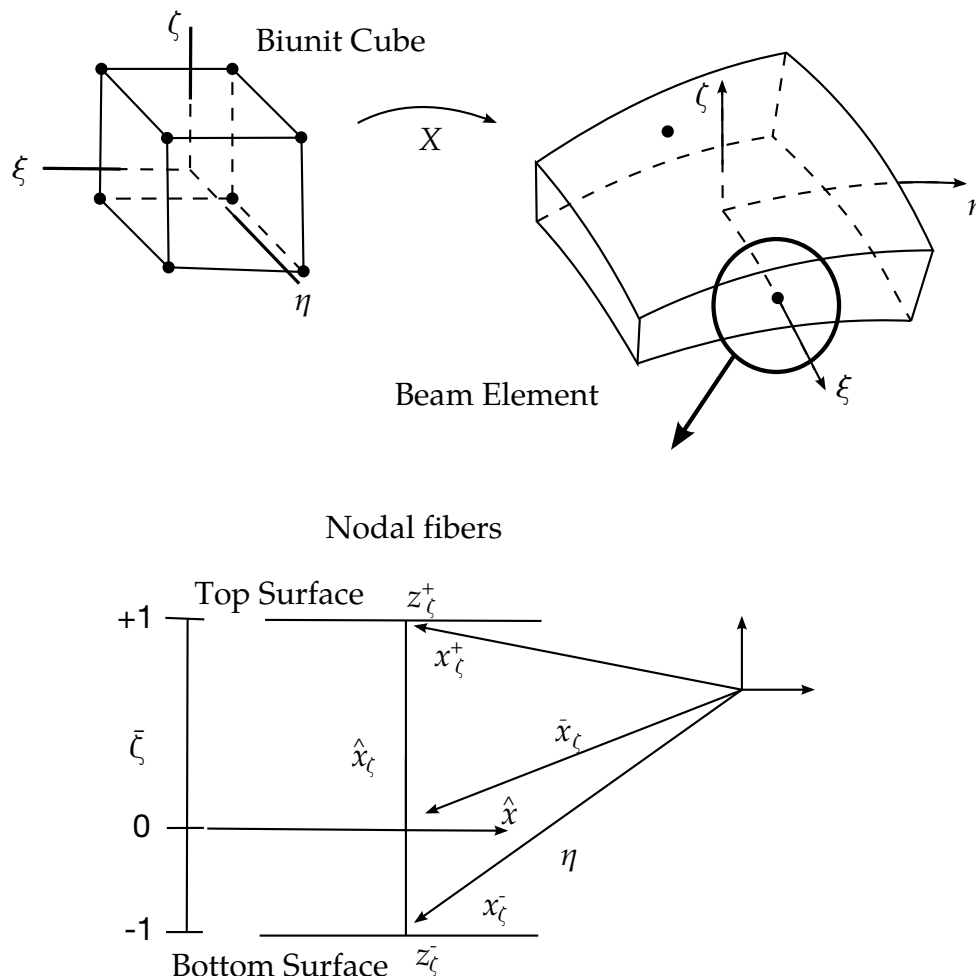


Figure 8.1. Hughes-Liu beam element.

$$\mathbf{X}_\eta(\xi, \eta) = \sum_{a=1}^2 N_a(\xi) \mathbf{X}_{\eta a}(\eta),$$

$$\mathbf{X}_\zeta(\xi, \zeta) = \sum_{a=1}^2 N_a(\xi) \mathbf{X}_{\zeta a}(\zeta).$$

With this description, arbitrary points on the reference line $\bar{\mathbf{x}}$ are interpolated by the one-dimensional shape function $N(\xi)$ operating on the global position of the two beam nodes that define the reference axis, i.e., $\bar{\mathbf{x}}_a$. Points off the reference axis are further interpolated by using a one-dimensional shape function along the fiber directions, i.e., $\mathbf{X}_{\eta a}(\eta)$ and $\mathbf{X}_{\zeta a}(\zeta)$ where

$$\begin{aligned} \mathbf{X}_{\eta a}(\eta) &= z_{\eta a}(\eta) \widehat{\mathbf{X}}_{\eta a} & \mathbf{X}_{\zeta a}(\zeta) &= z_{\zeta a}(\zeta) \widehat{\mathbf{X}}_{\zeta a} \\ z_{\eta a}(\eta) &= N_+(\eta) z_{\eta a}^+ + N_-(\eta) z_{\eta a}^- & z_{\zeta a}(\zeta) &= N_+(\zeta) z_{\zeta a}^+ + N_-(\zeta) z_{\zeta a}^- \\ N_+(\eta) &= \frac{(1+\eta)}{2} & N_+(\zeta) &= \frac{(1+\zeta)}{2} \\ N_-(\eta) &= \frac{(1-\eta)}{2} & N_-(\zeta) &= \frac{(1-\zeta)}{2} \end{aligned} \quad (8.5)$$

where $z_\zeta(\zeta)$ and $z_\eta(\eta)$ are “thickness functions”.

The Hughes-Liu beam formulation uses four position vectors, in addition to ξ , to locate the reference axis and define the initial fiber directions. Consider the two position vectors $\mathbf{x}_{\zeta a}^+$ and $\mathbf{x}_{\zeta a}^-$ located on the top and bottom surfaces, respectively, at node a . Then

$$\begin{aligned} \bar{\mathbf{x}}_{\zeta a} &= \frac{1}{2} (1 - \bar{\zeta}) \mathbf{x}_{\zeta a}^- + (1 + \bar{\zeta}) \mathbf{x}_{\zeta a}^+, & \bar{\mathbf{x}}_{\eta a} &= \frac{1}{2} (1 - \bar{\eta}) \mathbf{x}_{\eta a}^- + (1 + \bar{\eta}) \mathbf{x}_{\eta a}^+, \\ \widehat{\mathbf{X}}_{\zeta a} &= \frac{(\mathbf{x}_{\zeta a}^+ - \mathbf{x}_{\zeta a}^-)}{\|\mathbf{x}_{\zeta a}^+ - \mathbf{x}_{\zeta a}^-\|}, & \widehat{\mathbf{X}}_{\eta a} &= \frac{(\mathbf{x}_{\eta a}^+ - \mathbf{x}_{\eta a}^-)}{\|\mathbf{x}_{\eta a}^+ - \mathbf{x}_{\eta a}^-\|}, \\ z_{\zeta a}^+ &= \frac{1}{2} (1 - \bar{\zeta}) \|\mathbf{x}_{\zeta a}^+ - \mathbf{x}_{\zeta a}^-\|, & z_{\eta a}^+ &= \frac{1}{2} (1 - \bar{\eta}) \|\mathbf{x}_{\eta a}^+ - \mathbf{x}_{\eta a}^-\|, \\ z_{\zeta a}^- &= -\frac{1}{2} (1 + \bar{\zeta}) \|\mathbf{x}_{\zeta a}^+ - \mathbf{x}_{\zeta a}^-\|, & z_{\eta a}^- &= -\frac{1}{2} (1 + \bar{\eta}) \|\mathbf{x}_{\eta a}^+ - \mathbf{x}_{\eta a}^-\|, \end{aligned} \quad (8.6)$$

where $\|\cdot\|$ is the Euclidean norm. The reference surface may be located at the midsurface of the beam or offset at the outer surfaces. This capability is useful in several practical situations involving contact surfaces, connection of beam elements to solid elements, and offsetting elements such as for beam stiffeners in stiffened shells. The reference surfaces are located within the beam element by specifying the value of the parameters $\bar{\eta}$ and $\bar{\zeta}$, (see lower portion of [Figure 8.1](#)). When these parameters take on the values -1 or $+1$, the reference axis is located on the outer surfaces of the beam. If they are set to zero, the reference axis is at the center.

The same parametric representation used to describe the geometry of the beam elements is used to interpolate the beam element displacements, i.e., an isoparametric representation. Again the displacements are separated into the reference axis displacements and rotations associated with the fiber directions:

$$\begin{aligned}\mathbf{u}(\xi, \eta, \zeta) &= \bar{\mathbf{u}}(\xi) + \mathbf{U}(\xi, \eta, \zeta), \\ &= \bar{\mathbf{u}}(\xi) + \mathbf{U}_{\zeta}(\xi, \zeta) + \mathbf{U}_{\eta}(\xi, \eta).\end{aligned}\quad (8.7)$$

The reference axis is interpolated as usual

$$\bar{\mathbf{u}}(\xi) = \sum_{a=1}^2 N_a(\xi) \bar{\mathbf{u}}_a. \quad (8.8)$$

The displacements are also interpolated along the reference axis

$$\begin{aligned}\mathbf{U}_{\eta}(\xi, \eta) &= \sum_{a=1}^2 N_a(\xi) \mathbf{U}_{\eta a}(\eta), \\ \mathbf{U}_{\zeta}(\xi, \zeta) &= \sum_{a=1}^2 N_a(\xi) \mathbf{U}_{\zeta a}(\zeta).\end{aligned}\quad (8.9)$$

The fiber displacement is interpolated consistently with the thickness,

$$\begin{aligned}\mathbf{U}_{\eta a}(\eta) &= z_{\eta a}(\eta) \hat{\mathbf{U}}_{\eta a}, \\ \mathbf{U}_{\zeta a}(\zeta) &= z_{\zeta a}(\zeta) \hat{\mathbf{U}}_{\zeta a},\end{aligned}\quad (8.10)$$

where \mathbf{u} is the displacement of a generic point, $\bar{\mathbf{u}}$ is the displacement of a point on the reference surface, and \mathbf{U} is the “fiber displacement” rotations. The motion of the fibers can be interpreted as either displacements or rotations as will be discussed.

Hughes and Liu introduced the notation that follows, and the associated schematic shown in [Figure 8.2](#), to describe the current deformed configuration with respect to the reference configuration.

$$\begin{aligned}\mathbf{y} &= \bar{\mathbf{y}} + \mathbf{Y}, & \bar{\mathbf{y}} &= \bar{\mathbf{x}} + \bar{\mathbf{u}}, & \mathbf{Y} &= \mathbf{X} + \mathbf{U}, & \hat{\mathbf{Y}}_{\eta a} &= \hat{\mathbf{X}}_{\eta a} + \hat{\mathbf{U}}_{\eta a}, \\ \bar{\mathbf{y}}_a &= \bar{\mathbf{x}}_a + \bar{\mathbf{u}}_a, & \mathbf{Y}_a &= \mathbf{X}_a + \mathbf{U}_a, & \hat{\mathbf{Y}}_{\zeta a} &= \hat{\mathbf{X}}_{\zeta a} + \hat{\mathbf{U}}_{\zeta a},\end{aligned}\quad (8.11)$$

In the above relations, and in [Figure 8.2](#), the \mathbf{x} quantities refer to the reference configuration, the \mathbf{y} quantities refer to the updated (deformed) configuration and the \mathbf{u} quantities are the displacements. The notation consistently uses a superscript bar ($\bar{\cdot}$) to indicate reference surface quantities, a superscript caret ($\hat{\cdot}$) to indicate unit vector quantities, lower case letter for translational displacements, and upper case letters for fiber displacements. Thus to update to the deformed configuration, two vector quantities are needed: the reference surface displacement $\bar{\mathbf{u}}$ and the associated nodal fiber displacement \mathbf{U} . The nodal fiber displacements are defined in the fiber coordinate system, described in the next subsection.

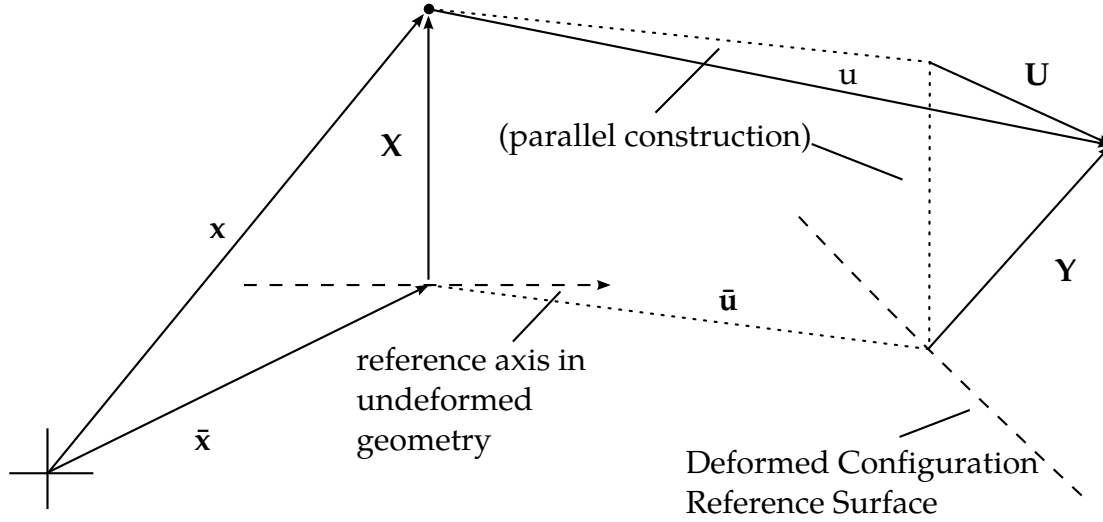


Figure 8.2. Schematic of deformed configuration displacements and position vectors.

8.2 Fiber Coordinate System

For a beam element, the known quantities will be the displacements of the reference surface \bar{u} obtained from the translational equations of motion and the rotational quantities at each node obtained from the rotational equations of motion. What remains to complete the kinematics is a relation between nodal rotations and fiber displacements U . The linearized relationships between the incremental components $\Delta \hat{U}$ the incremental rotations are given by

$$\begin{aligned} \begin{Bmatrix} \Delta \hat{U}_{\eta 1} \\ \Delta \hat{U}_{\eta 2} \\ \Delta \hat{U}_{\eta 3} \end{Bmatrix} &= \begin{bmatrix} 0 & \hat{Y}_{\eta 3} & -\hat{Y}_{\eta 2} \\ -\hat{Y}_{\eta 3} & 0 & \hat{Y}_{\eta 1} \\ \hat{Y}_{\eta 2} & -\hat{Y}_{\eta 1} & 0 \end{bmatrix} \begin{Bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{Bmatrix} = \mathbf{h}_{\eta} \Delta \theta, \\ \begin{Bmatrix} \Delta \hat{U}_{\zeta 1} \\ \Delta \hat{U}_{\zeta 2} \\ \Delta \hat{U}_{\zeta 3} \end{Bmatrix} &= \begin{bmatrix} 0 & \hat{Y}_{\zeta 3} & -\hat{Y}_{\zeta 2} \\ -\hat{Y}_{\zeta 3} & 0 & \hat{Y}_{\zeta 1} \\ \hat{Y}_{\zeta 2} & -\hat{Y}_{\zeta 1} & 0 \end{bmatrix} \begin{Bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \Delta \theta_3 \end{Bmatrix} = \mathbf{h}_{\zeta} \Delta \theta. \end{aligned} \quad (8.12)$$

Equations (8.12) are used to transform the incremental fiber tip displacements to rotational increments in the equations of motion. The second-order accurate rotational update formulation due to Hughes and Winget [1980] is used to update the fiber vectors:

$$\begin{aligned} \hat{Y}_{\eta i}^{n+1} &= R_{ij}(\Delta \theta) \hat{Y}_{\eta i}^n, \\ \hat{Y}_{\zeta i}^{n+1} &= R_{ij}(\Delta \theta) \hat{Y}_{\zeta i}^n, \end{aligned} \quad (8.13)$$

then

$$\begin{aligned}\Delta \widehat{\mathbf{U}}_{\eta a} &= \widehat{\mathbf{Y}}_{\eta a}^{n+1} - \widehat{\mathbf{Y}}_{\eta a}^n, \\ \Delta \widehat{\mathbf{U}}_{\zeta a} &= \widehat{\mathbf{Y}}_{\zeta a}^{n+1} - \widehat{\mathbf{Y}}_{\zeta a}^n,\end{aligned}\tag{8.14}$$

where

$$\begin{aligned}R_{ij}(\Delta\theta) &= \delta_{ij} + \frac{(2\delta_{ij} + \Delta S_{ik})\Delta S_{ik}}{2D}, \\ \Delta S_{ij} &= e_{ikj}\Delta\theta_k, \\ 2D &= 2 + \frac{1}{2}(\Delta\theta_1^2 + \Delta\theta_2^2 + \Delta\theta_3^2).\end{aligned}\tag{8.15}$$

Here δ_{ij} is the Kronecker delta and e_{ikj} is the permutation tensor.

8.2.1 Local Coordinate System

In addition to the above described fiber coordinate system, a local coordinate system is needed to enforce the zero normal stress conditions transverse to the axis. The orthonormal basis with two directions $\hat{\mathbf{e}}_2$ and $\hat{\mathbf{e}}_3$ normal to the axis of the beam is constructed as follows:

$$\begin{aligned}\hat{\mathbf{e}}_1 &= \frac{\bar{\mathbf{y}}_2 - \bar{\mathbf{y}}_1}{\|\bar{\mathbf{y}}_2 - \bar{\mathbf{y}}_1\|}, \\ \mathbf{e}'_2 &= \frac{\widehat{\mathbf{Y}}_{\eta 1} + \widehat{\mathbf{Y}}_{\eta 2}}{\|\widehat{\mathbf{Y}}_{\eta 1} + \widehat{\mathbf{Y}}_{\eta 2}\|}.\end{aligned}\tag{8.16}$$

From the vector cross product of these local tangents.

$$\hat{\mathbf{e}}_3 = \hat{\mathbf{e}}_1 \times \mathbf{e}'_2,\tag{8.17}$$

and to complete this orthonormal basis, the vector

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1,\tag{8.18}$$

is defined. This coordinate system rigidly rotates with the deformations of the element.

The transformation of vectors from the global to the local coordinate system can now be defined in terms of the basis vectors as

$$\widehat{\mathbf{A}} = \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}^T \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = [\mathbf{q}]\{\mathbf{A}\},\tag{8.19}$$

where e_{ix} , e_{iy} , e_{iz} are the global components of the local coordinate unit vectors, $\widehat{\mathbf{A}}$ is a vector in the local coordinates, and \mathbf{A} is the same vector in the global coordinate system.

8.3 Strains and Stress Update

8.3.1 Incremental Strain and Spin Tensors

The strain and spin increments are calculated from the incremental displacement gradient

$$G_{ij} = \frac{\partial \Delta u_i}{\partial y_j}, \quad (8.20)$$

where Δu_i are the incremental displacements and y_j are the deformed coordinates. The incremental strain and spin tensors are defined as the symmetric and skew-symmetric parts, respectively, of G_{ij} :

$$\begin{aligned} \Delta \varepsilon_{ij} &= \frac{1}{2} (G_{ij} + G_{ji}), \\ \Delta \omega_{ij} &= \frac{1}{2} (G_{ij} - G_{ji}). \end{aligned} \quad (8.21)$$

The incremental spin tensor $\Delta \omega_{ij}$ is used as an approximation to the rotational contribution of the Jaumann rate of the stress tensor; in an implicit implementation [Hallquist 1981b] the more accurate Hughes-Winget [1980] transformation matrix is used, [Equation \(8.15\)](#), with the incremental spin tensor for the rotational update. Here the Jaumann rate update is approximated as

$$\underline{\sigma}_{ij} = \sigma_{ij}^n + \sigma_{ip}^n \Delta \omega_{pj} + \sigma_{jp}^n \Delta \omega_{pi}, \quad (8.22)$$

where the superscripts on the stress tensor refer to the updated ($n + 1$) and reference (n) configurations. This update of the stress tensor is applied before the constitutive evaluation, and the stress and strain are stored in the global coordinate system.

8.3.2 Stress Update

To evaluate the constitutive relation, the stresses and strain increments are rotated from the global to the local coordinate system using the transformation defined previously in [Equation \(8.19\)](#), viz.

$$\begin{aligned} \sigma_{ij}^{ln} &= q_{ik} \sigma_{kn} q_{jn}, \\ \Delta \varepsilon_{ij}^l &= q_{ik} \Delta \varepsilon_{kn} q_{jn}, \end{aligned} \quad (8.23)$$

where the superscript l indicates components in the local coordinate system. The stress is updated incrementally:

$$\sigma_{ij}^{ln+1} = \sigma_{ij}^{ln} + \Delta \sigma_{ij}^{ln+\frac{1}{2}}, \quad (8.24)$$

and rotated back to the global system:

$$\sigma_{ij}^{n+1} = q_{ki} \sigma_{kn}^{ln+1} q_{nj}, \quad (8.25)$$

before computing the internal force vector.

8.3.3 Incremental Strain-Displacement Relations

After the constitutive evaluation is completed, the fully updated stresses are rotated back to the global coordinate system. These global stresses are then used to update the internal force vector

$$\mathbf{f}_a^{\text{int}} = \int \mathbf{B}_a^T \boldsymbol{\sigma} dv, \quad (8.26)$$

where $\mathbf{f}_a^{\text{int}}$ are the internal forces at node a and \mathbf{B}_a is the strain-displacement matrix in the global coordinate system associated with the displacements at node a . The \mathbf{B} matrix relates six global strain components to eighteen incremental displacements [three translational displacements per node and the six incremental fiber tip displacements of Equation (8.14)]. It is convenient to partition the \mathbf{B} matrix:

$$\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2]. \quad (8.27)$$

Each B_a sub matrix is further partitioned into a portion due to strain and spin with the following sub matrix definitions:

$$\mathbf{B}_a = \begin{bmatrix} B_1 & 0 & 0 & B_4 & 0 & 0 & B_7 & 0 & 0 \\ 0 & B_2 & 0 & 0 & B_5 & 0 & 0 & B_8 & 0 \\ 0 & 0 & B_3 & 0 & 0 & B_6 & 0 & 0 & B_9 \\ B_2 & B_1 & 0 & B_5 & B_4 & 0 & B_8 & B_7 & 0 \\ 0 & B_3 & B_2 & 0 & B_6 & B_5 & 0 & B_9 & B_8 \\ B_3 & 0 & B_1 & B_6 & 0 & B_4 & B_9 & 0 & B_7 \end{bmatrix}, \quad (8.28)$$

where,

$$B_i = \begin{cases} N_{a,i} = \frac{\partial N_a}{\partial y_i} & i = 1,2,3 \\ (N_a z_{\eta a})_{,i-3} = \frac{\partial (N_a z_{\eta a})}{\partial y_{i-3}} & i = 4,5,6. \\ (N_a z_{\zeta a})_{,i-6} = \frac{\partial (N_a z_{\zeta a})}{\partial y_{i-6}} & i = 7,8,9 \end{cases} \quad (8.29)$$

With respect to the strain-displacement relations, note that:

- The derivative of the shape functions are taken with respect to the global coordinates;
- The \mathbf{B} matrix is computed on the cross-section located at the mid-point of the axis;
- The resulting \mathbf{B} matrix is a 6×18 matrix.

The internal force, \mathbf{f} , given by

$$\mathbf{f}' = \mathbf{T}^T \mathbf{f}_a^{\text{int}} \quad (8.30)$$

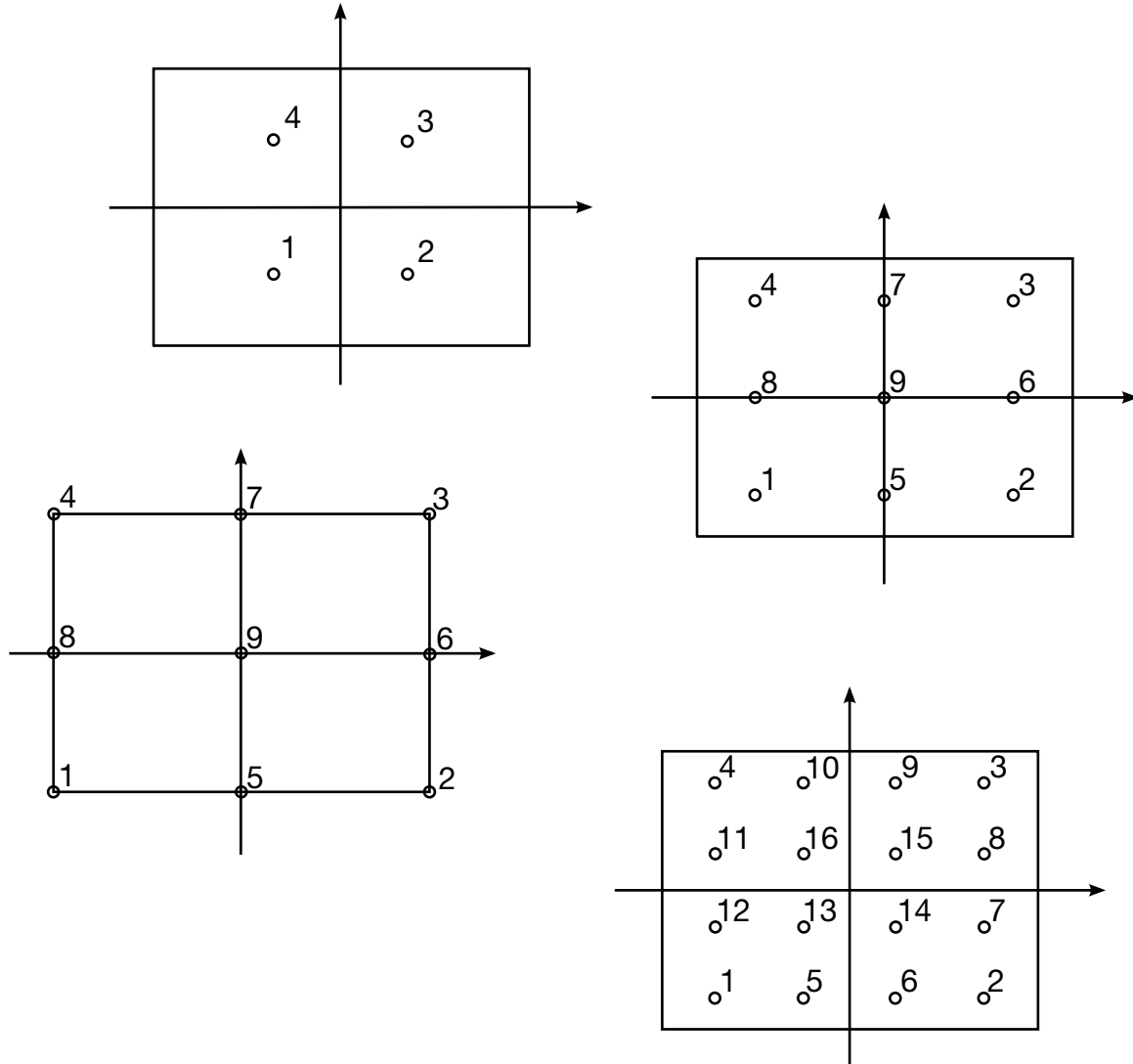


Figure 8.3. Integration possibilities for rectangular cross sections in the Hughes-Liu beam element.

is assembled into the global right hand side internal force vector. \mathbf{T} is defined as (also see [Equation \(8.12\)](#)):

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_\eta \\ \mathbf{0} & \mathbf{h}_\zeta \end{bmatrix}, \quad (8.31)$$

where \mathbf{I} the 3×3 identity matrix.

8.3.4 Spatial Integration

The integration of [Equation \(8.26\)](#) for the beam element is performed with one-point integration along the axis and multiple points in the cross section. For rectangular cross sections, a variety of choices are available as is shown in [Figure 8.3](#). The beam has no zero energy or locking modes.

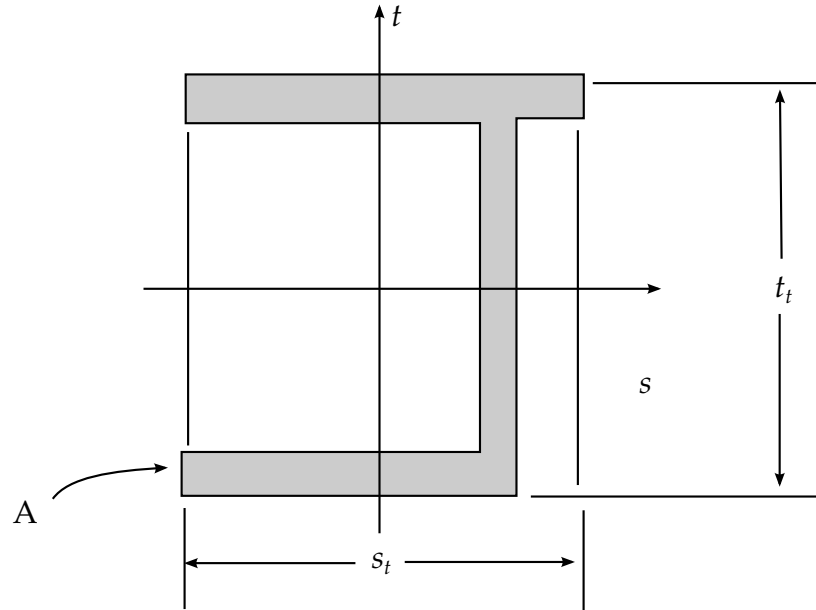


Figure 8.4. Specification of the nodal thickness, s_t and t_t , for a beam with an arbitrary cross-section.

For the user defined rule, it is necessary to specify the number of integration points and the relative area for the total cross section:

$$A_r = \frac{A}{s_t \cdot t_t}$$

where s_t and t_t are the beam thickness specified on either the cross section or beam element cards. The rectangular cross-section which contains s_t and t_t should completely contain the cross-sectional geometry. Figure 8.4 illustrates this for a typical cross-section. In Figure 5.5, the area is broken into twelve integration points. For each integration point, it is necessary to define the s and t parametric coordinates, (s_i, t_i) , of the centroid of the i^{th} integration point and the relative area associated with the point

$$A_{ri} = \frac{A_i}{A}$$

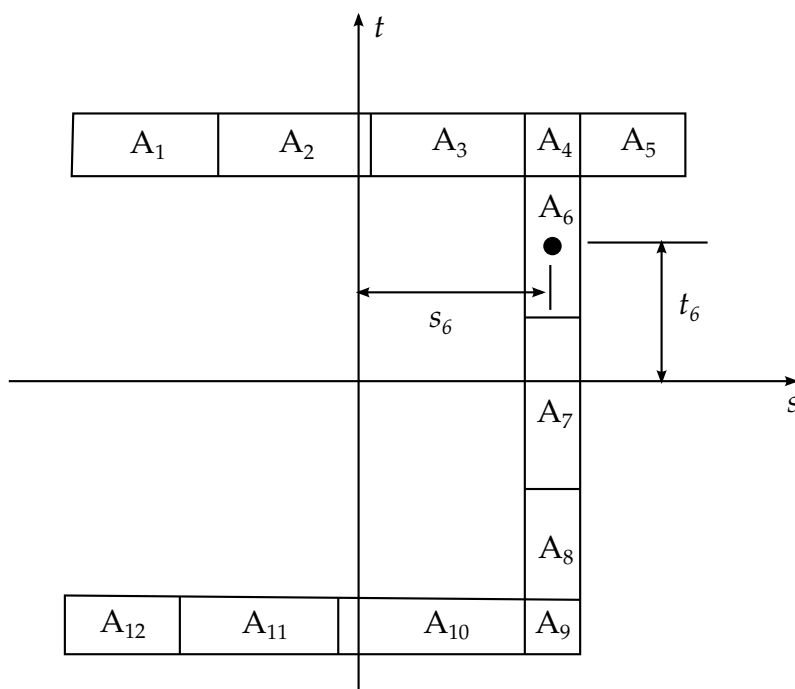


Figure 8.5. A breakdown of the cross section geometry in [Figure 8.4](#) into twelve integration points.

where A_i is the ratio of the area of the integration point and the actual area of the cross-section, A .

9

Warped Beam Elements

9.1 Resultant Warped Beam

9.1.1 Green-Lagrange Strains in Terms of Deformational Displacements

All quantities in this section are referred to the local element coordinate system \mathbf{e}_i , $i = 1, 2, 3$. The origin of the local system is taken at node 1, with \mathbf{e}_1 directed along the line of centroids, while \mathbf{e}_2 , and \mathbf{e}_3 are directed along the principal axes of the cross-section.

With respect to the local system, the Green-Lagrange strain tensor can be written as:

$$\varepsilon_{ij} = e_{ij} + \eta_{ij}, \quad (9.1)$$

where,

$$\begin{aligned} e_{ij} &= 0.5(u_{i,j} + u_{j,i}), \\ \eta_{ij} &= 0.5u_{k,i}u_{k,j}. \end{aligned} \quad (9.2)$$

The geometric assumption of infinite in-plane rigidity implies $\varepsilon_{22} = \varepsilon_{33} = \gamma_{23} = 0$. Then the non-zero strain components which contribute to the strain energy are:

$$\begin{aligned} \varepsilon_{11} &= u_{1,1} + \frac{1}{2}(u_{1,1}^2 + u_{2,1}^2 + u_{3,1}^2), \\ 2\varepsilon_{12} &= u_{1,2} + u_{2,1} + u_{1,1}u_{1,2} + u_{2,1}u_{2,2} + u_{3,1}u_{3,2}, \\ 2\varepsilon_{13} &= u_{1,3} + u_{3,1} + u_{1,1}u_{1,3} + u_{2,1}u_{2,3} + u_{3,1}u_{3,3}. \end{aligned} \quad (9.3)$$

9.1.2 Deformational Displacements After Large Rotations

The position vectors of an arbitrary point P in the initial and current local configurations are:

$$\mathbf{x}_P^0 = \mathbf{x}_C^0 + [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix}, \quad (9.4)$$

$$\mathbf{x}_P = \mathbf{x}_C + [\mathbf{e}'_1 \quad \mathbf{e}'_2 \quad \mathbf{e}'_3] \begin{bmatrix} \omega\phi \\ x_2 \\ x_3 \end{bmatrix}, \quad (9.5)$$

respectively, with

$$[\mathbf{e}'_1 \quad \mathbf{e}'_2 \quad \mathbf{e}'_3] = \left[\mathbf{I} + \boldsymbol{\theta} + \frac{1}{2} \boldsymbol{\theta}^2 \right] [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3], \quad (9.6)$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}, \quad (9.7)$$

and ω is the Saint-Venant warping function about the centroid C . By the transfer theorem, the following relation holds:

$$\omega = \omega + c_2 x_3 - c_3 x_2, \quad (9.8)$$

where ω refers to the shear center S , and c_2 and c_3 are the coordinates of S .

Subtracting Equation (9.4) from Equation (9.5) and neglecting third-order terms, the displacements vector of point P can be computed:

$$\begin{aligned} u_1 &= \bar{u}_1 - x_2 \theta_3 + x_3 \theta_2 + \frac{1}{2} x_2 \theta_1 \theta_2 + \frac{1}{2} x_3 \theta_1 \theta_3 + \omega \phi, \\ u_2 &= \bar{u}_2 - x_3 \theta_1 - \frac{1}{2} x_2 (\theta_1^2 + \theta_3^2) + \frac{1}{2} x_3 \theta_2 \theta_3 + \omega \theta_3 \phi, \\ u_3 &= \bar{u}_3 + x_2 \theta_1 - \frac{1}{2} x_3 (\theta_1^2 + \theta_2^2) + \frac{1}{2} x_2 \theta_2 \theta_3 - \omega \theta_2 \phi, \end{aligned} \quad (9.9)$$

where \bar{u}_1 , \bar{u}_2 , and \bar{u}_3 are the displacements of the centroid C .

9.1.3 Green-Lagrange Strains in terms of Centroidal Displacements and Angular Rotations

From Equations (9.3) and (9.9), a second-order approximation of the Green-Lagrange strains can be evaluated. Neglecting term $\frac{1}{2} u_{1,1}^2$ and the nonlinear strain components generated by warping, the strain components are simplified as

$$\begin{aligned} \varepsilon_{11} &= \varepsilon_0 + x_2 \kappa_2 + x_3 \kappa_3 + \frac{1}{2} (x_2^2 + x_3^2) \theta_{1,1}^2 + \omega \phi_{,1}, \\ 2\varepsilon_{12} &= \gamma_{12} + \omega_{,2} \phi - x_3 \kappa_1, \\ 2\varepsilon_{13} &= \gamma_{13} + \omega_{,3} \phi + x_2 \kappa_1, \end{aligned} \quad (9.10)$$

with

$$\begin{aligned}
\varepsilon_0 &= \bar{u}_{1,1} + \frac{1}{2}(\bar{u}_{2,1}^2 + \bar{u}_{3,1}^2), \\
\kappa_1 &= \theta_{1,1} + \frac{1}{2}(\theta_{2,1}\theta_3 - \theta_{3,1}\theta_2), \\
\kappa_2 &= -\theta_{3,1} + \frac{1}{2}(\theta_1\theta_{2,1} + \theta_{1,1}\theta_2) + \bar{u}_{3,1}\theta_{1,1} - c_3\phi_{,1}, \\
\kappa_3 &= \theta_{2,1} + \frac{1}{2}(\theta_1\theta_{3,1} + \theta_{1,1}\theta_3) - \bar{u}_{2,1}\theta_{1,1} + c_2\phi_{,1}, \\
\gamma_{12} &= \bar{u}_{2,1} - \theta_3 + \frac{1}{2}\theta_1\theta_2 + \bar{u}_{3,1}\theta_1 - \bar{u}_{1,1}\theta_3, \\
\gamma_{13} &= \bar{u}_{3,1} + \theta_2 + \frac{1}{2}\theta_1\theta_3 - \bar{u}_{2,1}\theta_1 + \bar{u}_{1,1}\theta_2.
\end{aligned} \tag{9.11}$$

Numerical testing has shown that neglecting the nonlinear terms in the curvatures $\kappa_1, \kappa_2, \kappa_3$ and bending shear strains γ_{12}, γ_{13} has little effect on the accuracy of the results. Therefore, Equation (9.11) can be simplified to

$$\begin{aligned}
\kappa_1 &= \theta_{1,1}, & \kappa_2 &= -\theta_{3,1} - c_3\phi_{,1}, & \kappa_3 &= \theta_{2,1} + c_2\phi_{,1}, \\
\varepsilon_0 &= \bar{u}_{1,1} + \frac{1}{2}(\bar{u}_{2,1}^2 + \bar{u}_{3,1}^2), & \gamma_{12} &= \bar{u}_{2,1} - \theta_3, & \gamma_{13} &= \bar{u}_{3,1} + \theta_2.
\end{aligned} \tag{9.12}$$

Adopting Bernoulli's assumption ($\gamma_{12} = \gamma_{13} = 0$) and Vlasov's assumption ($\phi = \theta_{1,1}$), Equation (9.10) can be rewritten as:

$$\begin{aligned}
\varepsilon_{11} &= \varepsilon_0 + x_2\kappa_2 + x_3\kappa_3 + \frac{1}{2}r^2\kappa_1^2 + \omega\theta_{1,11}, \\
2\varepsilon_{12} &= (\omega_{,2} - x_3)\kappa_1, \\
2\varepsilon_{13} &= (\omega_{,3} - x_2)\kappa_1,
\end{aligned} \tag{9.13}$$

where

$$r^2 = x_2^2 + x_3^2, \quad \kappa_1 = \theta_{1,1}, \quad \kappa_2 = -\bar{u}_{2,11} - c_3\theta_{1,11}, \quad \kappa_3 = -\bar{u}_{3,11} + c_2\theta_{1,11}. \tag{9.14}$$

To avoid membrane locking, ε_{11} in Equation (9.13) is reformulated as

$$\varepsilon_{11} = \varepsilon_a + x_2\kappa_2 + x_3\kappa_3 + \frac{1}{2}\left(r^2 - \frac{I_o}{A}\right)\kappa_1^2 + \omega\theta_{1,11}, \tag{9.15}$$

where

$$\varepsilon_a = \frac{1}{l} \int_0^1 \left[\bar{u}_{1,1} + \frac{1}{2}(\bar{u}_{2,1}^2 + \bar{u}_{3,1}^2 + \frac{I_o}{A}\kappa_1^2) \right] dx_1. \tag{9.16}$$

9.1.4 Strain Energy

Assuming material is linear elastic, the strain energy can be evaluated from:

$$U = \int_0^L \left(\frac{1}{2} E \int_A \varepsilon_{11}^2 dA + \frac{1}{2} G \int_A [(2\varepsilon_{12})^2 + (2\varepsilon_{13})^2] dA \right) dx_1. \quad (9.17)$$

The following relations are used in integrating the previous equations:

(1) Since the reference frame is located at centroid C with e_2 and e_3 directed along the principal axes,

$$\int_A x_2 dA = 0, \quad \int_A x_3 dA = 0, \quad \int_A x_2 x_3 dA = 0. \quad (9.18)$$

(2) Since sectorial area ω refers to shear center S ,

$$\int_A \omega dA = 0, \quad \int_A x_2 \omega dA = 0, \quad \int_A x_3 \omega dA = 0. \quad (9.19)$$

Integration through the cross-section gives:

$$\begin{aligned} \int_A \varepsilon_{11}^2 dA + A \varepsilon_a^2 + I_{22} \kappa_2^2 + I_{33} \kappa_3^2 + I_\omega \theta_{1,1}^2 + I_{2r} \kappa_2 \kappa_1^2 + I_{3r} \kappa_3 \kappa_1^2 + I_{\omega r} \theta_{1,1} \kappa_1^2 \\ + \frac{1}{4} \left(I_{rr} - \frac{I_o^2}{A} \right) \kappa_1^4 \end{aligned} \quad (9.20)$$

$$\int_A [(2\varepsilon_{12})^2 + (2\varepsilon_{13})^2] dA = J \kappa_1^2 \quad (9.21)$$

with

$$\begin{aligned} I_{22} &= \int_A x_2^2 dA, \quad I_{33} = \int_A x_3^2 dA, \quad I_O = I_{22} + I_{33} \\ I_{2r} &= \int_A x_2 r^2 dA, \quad I_{3r} = \int_A x_3 r^2 dA, \quad I_{rr} = \int_A r^4 dA \\ I_\omega &= \int_A \omega^2 dA, \quad I_{\omega r} = \int_A \omega r^2 dA, \quad J = \int_A [(\bar{\omega}_{,3} + x_2)^2 + (\bar{\omega}_{,2} - x_3)^2] dA \end{aligned} \quad (9.22)$$

6.1.5 Displacement Field

Linear interpolation is used for axial displacement \bar{u} , whereas Hermitian interpolations are used for u_2 , u_3 , and θ_1 , considering the following relations used in deriving the final expression of strain energy:

$$\theta_2 = -u_{3,1} \quad \theta_3 = u_{2,1} \quad \phi = \theta_{1,1}. \quad (9.23)$$

The nodal displacement field is constructed by

$$\bar{u}_1 = \mathbf{N}_1 \mathbf{d}, \quad (9.24)$$

$$\begin{Bmatrix} \bar{u}_2 \\ \bar{u}_3 \\ \theta_1 \end{Bmatrix} = \mathbf{N}_2 \mathbf{d}, \quad (9.25)$$

where

$$\mathbf{d}^T = [0 \quad 0 \quad 0 \quad 0 \quad \theta_{2I} \quad \theta_{3I} \quad \phi_I \quad \bar{u}_{1JI} \quad 0 \quad 0 \quad \theta_{1JI} \quad \theta_{2J} \quad \theta_{3J} \quad \phi_J]^T, \quad (9.26)$$

$$\mathbf{N}_1 = [1 - \zeta \quad \dots \quad \zeta \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot],$$

$$\mathbf{N}_2 = \begin{bmatrix} \cdot f & \cdot \cdot \cdot g & \cdot & \cdot & 1-f & \cdot & \cdot & \cdot & h & \cdot \\ \cdot & f & \cdot & -g & \cdot & \cdot & 1-f & \cdot & -h & \cdot & \cdot \\ \cdot & \cdot & f & \cdot & \cdot & g & \cdot & \cdot & 1-f & \cdot & \cdot & h \end{bmatrix}, \quad (9.27)$$

with

$$f = 1 - 3\zeta^2 + 2\zeta^3 g = l(\zeta - 2\zeta^2 + \zeta^3)h = l(\zeta^3 - \zeta^2). \quad (9.28)$$

Equations (9.25) and (9.27) also imply

$$\theta_{1,1} = \mathbf{N}_3 \mathbf{d}, \quad (9.29)$$

where

$$\mathbf{N}_3 = [\cdot f_{,1} \quad \cdot g_{,1} \quad \dots \quad -f_{,1} \quad \cdot h_{,1}]. \quad (9.30)$$

6.1.6 Strain Energy in Matrix Form

The strain energy due to the average strain ϵ_a defined in Equation (9.16) can be expressed in matrix form as

$$U_1 = \frac{1}{2} E A l \left[\int_0^1 (\mathbf{N}_{1,1} \mathbf{d}) d\zeta + \frac{1}{2} \mathbf{d}^T \left[\int_0^1 (\mathbf{N}_{2,1}^T \mathbf{D} \mathbf{N}_{2,1}) d\zeta \right] \mathbf{d} \right]^2, \quad (9.31)$$

where

$$\mathbf{D} = \text{diag} \left(1, 1, \frac{I_o}{A} \right). \quad (9.32)$$

The strain energy due to the second through fourth terms is

$$U_2 = \frac{1}{2} E l \mathbf{d}^T \left[\int_0^1 \mathbf{N}_{2,11}^T \mathbf{H} \mathbf{N}_{2,11} d\zeta \right] \mathbf{d}, \quad (9.33)$$

where

$$\mathbf{H} = \begin{bmatrix} I_{22} & & I_{22} c_3 \\ & I_{33} & -I_{33} c_2 \\ I_{22} c_3 & -I_{33} c_2 & I'_{\omega} \end{bmatrix}, \quad I'_{\omega} = I_{\omega} + I_{22} c_3^2 + I_{33} c_2^2. \quad (9.34)$$

The strain energy due to the fifth through seventh terms is

$$U_3 = \frac{1}{2} El \left[\int_0^1 (\mathbf{N}_3 \mathbf{d})^2 \mathbf{v} \mathbf{N}_{2,11} d\zeta \right] \mathbf{d}, \quad (9.35)$$

where

$$\mathbf{v} = (-I_{2r} \quad -I_{3r} \quad I'_{\omega r}), \quad I'_{\omega r} = I_{\omega r} - c_3 I_{2r} + c_2 I_{3r}. \quad (9.36)$$

The strain energy due to the eighth and ninth terms is

$$U_4 = \frac{1}{8} El \left(I_{rr} - \frac{I_o^2}{A} \right) \int_0^1 (\mathbf{N}_3 \mathbf{d})^4 d\zeta + \frac{1}{2} GJl \int_0^1 (\mathbf{N}_3 \mathbf{d})^2 d\zeta \quad (9.37)$$

6.1.7 Internal Nodal Force Vector

The internal force can be evaluated from

$$\mathbf{f}_e = EAl \left(\frac{\bar{\mathbf{u}}_{1JL}}{1} + \frac{1}{2} \mathbf{d}^T \mathbf{Q} \right) (\mathbf{P} + \mathbf{Q}) + E \left(\mathbf{R} + \frac{1}{2} \mathbf{S} + \mathbf{T} + \frac{1}{2} \mathbf{V} \right) + \mathbf{GW}, \quad (9.38)$$

where

$$\begin{aligned} \mathbf{P} &= \int_0^1 \mathbf{N}_{1,1}^T d\zeta, \quad \mathbf{Q} = \left[\int_0^1 \mathbf{N}_{2,1}^T \mathbf{D} \mathbf{N}_{2,1} d\zeta \right] \mathbf{d}, \quad \mathbf{R} = l \left[\int_0^1 \mathbf{N}_{2,11}^T \mathbf{H} \mathbf{N}_{2,11} d\zeta \right] \mathbf{d}, \\ \mathbf{S} &= l \left[\int_0^1 (\mathbf{N}_3 \mathbf{d})^2 \mathbf{N}_{2,11}^T d\zeta \right] \mathbf{v}^T, \quad \mathbf{T} = l \int_0^1 (\mathbf{N}_3 \mathbf{d}) (\mathbf{v} \mathbf{N}_{2,11} \mathbf{d}) \mathbf{N}_3^T d\zeta, \\ \mathbf{V} &= \left(I_{rr} - \frac{I_o^2}{A} \right) l \int_0^1 (\mathbf{N}_3 \mathbf{d})^3 \mathbf{N}_3^T d\zeta, \quad \mathbf{W} = Jl \int_0^1 (\mathbf{N}_3 \mathbf{d}) \mathbf{N}_3^T d\zeta, \quad \lambda = \frac{\bar{\mathbf{u}}_{1JL}}{1} + \frac{1}{2} \mathbf{d}^T \mathbf{Q}. \end{aligned} \quad (9.39)$$

With respect to the local coordinate system, there are totally eight independent components in the nodal force vector, in correspondence to the eight nodal displacement components.

Other forces can be calculated by:

$$\begin{aligned} \mathbf{F}_1 &= -\mathbf{F}_8, \quad \mathbf{F}_2 = \frac{\mathbf{F}_6 + \mathbf{F}_{13}}{1}, \quad \mathbf{F}_3 = -\frac{\mathbf{F}_5 + \mathbf{F}_{12}}{1}, \\ \mathbf{F}_4 &= -\mathbf{F}_{11}, \quad \mathbf{F}_9 = -\mathbf{F}_2, \quad \mathbf{F}_{10} = -\mathbf{F}_3. \end{aligned} \quad (9.40)$$

6.2 Integrated Warped Beam

6.2.1 Kinematics

We introduce three coordinate systems that are mutually interrelated. The first coordinate system is the orthogonal Cartesian coordinate system (x, y, z) , for which the y and z -axes lie in the plane of the cross-section and the x -axis parallel to the longitudinal axis of the beam. The second coordinate system is the local plate coordinate system (x, s, n) as shown in Figure 6.1, wherein the n -axis is normal to the middle surface of a plate element, the s -axis is tangent to the middle surface and is directed along the contour line of the cross-section. The (x, s, n) and (x, y, z) coordinate systems are related through

an angle of orientation θ as defined in Figure 40.1. The third coordinate set is the contour coordinate s along the profile of the section with its origin at some point O on the profile section. Point P is called the pole through which the axis parallel to the x -axis is called the pole axis. To derive the analytical model for a thin-walled beam, the following two assumptions are made:

1. The contour of the thin wall does not deform in its own plane.
2. The shear strain γ_{sx} of the middle surface is zero.

According to assumption 1, the midsurface displacement components v and w with respect to the (x, s, n) coordinate system at a point A can be expressed in terms of displacements \mathbf{V} and \mathbf{W} of the pole P in the (x, y, z) coordinate system and the rotation angle ϕ_x about the pole axis

$$\begin{aligned} \mathbf{v}(x, s) &= \mathbf{V}(x)\cos\theta(s) + \mathbf{W}(x)\sin\theta(s) - \mathbf{r}(s)\phi_x(x), \\ \mathbf{w}(x, s) &= -\mathbf{V}(x)\sin\theta(s) + \mathbf{W}(x)\cos\theta(s) - \mathbf{q}(s)\phi_x(x). \end{aligned} \quad (9.41)$$

These equations apply to the whole contour. The out-of-plane displacement u can now be found from assumption 2. On the middle surface

$$\frac{\partial \mathbf{u}}{\partial s} + \frac{\partial \mathbf{v}}{\partial x} = 0, \quad (9.42)$$

which can be written

$$\frac{\partial \mathbf{u}}{\partial s} = -\frac{\partial \mathbf{v}}{\partial x} = -\mathbf{V}'(x)\cos\theta(s) - \mathbf{W}'(x)\sin\theta(s) + \mathbf{r}(s)\phi'_x(x). \quad (9.43)$$

Integrating this relation from point O to an arbitrary point on the contour yields (using t as a dummy for s)

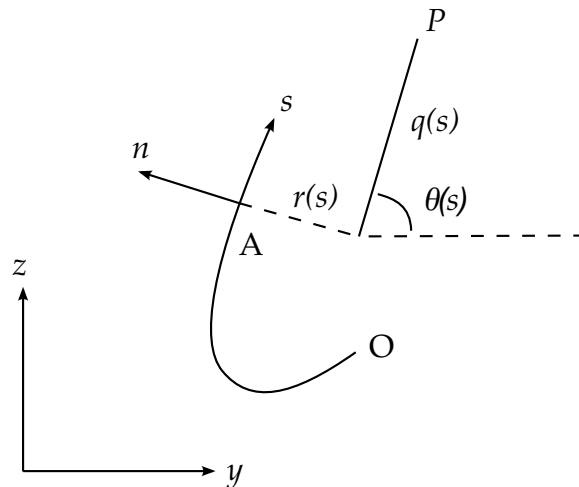


Figure 9.1. Definition of coordinates in thin-walled open section

$$\int_0^s \frac{\partial \mathbf{u}}{\partial t} dt = -\mathbf{V}'(x) \int_0^s \cos\theta(t) dt - \mathbf{W}'(x) \int_0^s \sin\theta(t) dt + \phi'_x(x) \int_0^s \mathbf{r}(t) dt. \quad (9.44)$$

Noting that

$$\begin{aligned} dy &= \cos\theta(t) dt, \\ dz &= \sin\theta(t) dt. \end{aligned} \quad (9.45)$$

we end up with

$$\begin{aligned} u(x, s) &= \underbrace{u(x, 0) + V'(x)y(0) + W'(x)z(0) + \phi'_x(x)\omega}_{=:\bar{U}(x)} \\ &\quad - \underbrace{V'(x)}_{=:\phi_z(x)} y(s) - \underbrace{W'(x)}_{=:-\phi_y(x)} z(s) + \phi'_x(x)(\omega(s) - \omega) \\ &= U(x) - \phi_z(x)y + \phi_y(x)z + \phi'_x(x)(\omega(s) - \omega). \end{aligned} \quad (9.46)$$

where U denotes the average out-of-plane displacement over the section, ϕ_y and ϕ_z denote the rotation angle about the y and z axis², respectively, ω is the sectorial area defined as

$$\omega(s) = \int_0^s r(t) dt, \quad (9.47)$$

and $\bar{\omega}$ is the average of the sectorial area over the section.

The expression for the displacements in the (x, y, z) coordinate system is

$$\begin{aligned} u(x, y, z) &= U(x) - \phi_z(x)y + \phi_y(x)z + \vartheta(x)\omega(y, z), \\ v(x, y, z) &= V(x) - \phi_x(x)z, \\ w(x, y, z) &= W(x) + \phi_x(x)y, \end{aligned} \quad (9.48)$$

where we have introduced ϑ to represent the twist constrained by the condition

$$\vartheta(x) = \phi_x(x). \quad (9.49)$$

and ω denotes the sectorial coordinate that is adjusted for zero average over the section.

6.2.2 Kinetics

The kinetic energy of the beam can be written

$$T = \frac{1}{2} \int_V \rho \{ \dot{u}^2 + \dot{v}^2 + \dot{w}^2 \} dV. \quad (9.50)$$

Taking the variation of this expression leads to

² The substitution of $V'(x)$ for $\phi_z(x)$ and $W'(x)$ for $-\phi_y(x)$ can be seen as a conversion from an Euler-Bernoulli kinematic assumption to that of Timoshenko.

$$\begin{aligned}
\delta T &= \int_V \rho \{ \dot{u} \delta \dot{u} + \dot{v} \delta \dot{v} + \dot{w} \delta \dot{w} \} dV \\
&= \int_V \rho \{ \dot{U} - \dot{\phi}_z y + \dot{\phi}_y z + \dot{\theta} \omega \} \{ \delta \dot{U} - \delta \dot{\phi}_z y + \delta \dot{\phi}_y z + \delta \dot{\theta} \omega \} dV + \\
&\quad \int_V \rho \{ \dot{V} - \dot{\phi}_x z \} \{ \delta \dot{V} - \delta \dot{\phi}_x z \} dV + \int_V \rho \{ \dot{W} + \dot{\phi}_x y \} \{ \delta \dot{W} + \delta \dot{\phi}_x y \} dV \\
&= \int_V \rho \{ \dot{U} \delta \dot{U} + y^2 \dot{\phi}_z \delta \dot{\phi}_z - y \omega \dot{\phi}_z \delta \dot{\theta} + z^2 \dot{\phi}_y \delta \dot{\phi}_y \} dV + \\
&\quad \int_V \rho \{ z \omega \dot{\phi}_y \delta \dot{\theta} - y \omega \dot{\theta} \delta \dot{\phi}_z + z \omega \dot{\theta} \delta \dot{\phi}_y + \omega^2 \dot{\theta} \delta \dot{\theta} \} dV + \\
&\quad \int_V \rho \{ \dot{V} \delta \dot{V} + z^2 \dot{\phi}_x \delta \dot{\phi}_x + \dot{W} \delta \dot{W} + y^2 \dot{\phi}_x \delta \dot{\phi}_x \} dV \\
&= \rho A \int_1 \{ \dot{U} \delta \dot{U} + \dot{V} \delta \dot{V} + \dot{W} \delta \dot{W} \} dV + \rho I_{zz} \int_1 \{ \dot{\phi}_z \delta \dot{\phi}_z + \dot{\phi}_x \delta \dot{\phi}_x \} dV + \\
&\quad \rho I_{yy} \int_1 \{ \dot{\phi}_y \delta \dot{\phi}_y + \dot{\phi}_x \delta \dot{\phi}_x \} dV + \rho I_{y\omega} \int_1 \{ \dot{\phi}_y \delta \dot{\theta} + \dot{\theta} \delta \dot{\phi}_y \} dV - \\
&\quad \rho I_{z\omega} \int_1 \{ \dot{\theta} \delta \dot{\phi}_z + \dot{\phi}_z \delta \dot{\theta} \} dV + \rho I_{\omega\omega} \int_1 \dot{\theta} \delta \dot{\theta} dV,
\end{aligned} \tag{9.51}$$

from which the consistent mass matrix can be read out. Here A is the cross sectional area, I_{zz} and I_{yy} are the second moments of area with respect to the z and y axes, respectively, $I_{\omega\omega}$ is the sectorial second moment and $I_{z\omega}$ and $I_{y\omega}$ are the sectorial product moments. An approximation of this mass matrix can be made by neglecting the off diagonal components. The diagonal components are

$$\begin{aligned}
m_{\text{TRNS}} &= \frac{\rho A l}{3}, \\
m_{\text{RTx}} &= \frac{\rho (I_{yy} + I_{zz}) l}{3}, \\
m_{\text{RTy}} &= \frac{\rho I_{yy} l}{3}, \\
m_{\text{RTz}} &= \frac{\rho I_{zz} l}{3}, \\
m_{\text{TWST}} &= \frac{\rho I_{\omega\omega} l}{3}.
\end{aligned} \tag{9.52}$$

With E as Young's modulus and G as the shear modulus, the strain energy can be written

$$\Pi = \frac{1}{2} (E\varepsilon_{xx}^2 + G\gamma_{xy}^2 + G\gamma_{xz}^2), \quad (9.53)$$

where the infinitesimal strain components are (neglecting the derivatives of sectorial area)

$$\begin{aligned} \varepsilon_{xx} &= U' + \phi_y' z - \phi_z' y + \vartheta' \omega, \\ \gamma_{xy} &= V' - \phi_x' z - \phi_z', \\ \gamma_{xz} &= W' + \phi_x' y + \phi_y'. \end{aligned} \quad (9.54)$$

and the variation of the same can be written

$$\begin{aligned} \delta\varepsilon_{xx} &= \delta U' + \delta\phi_y' z - \delta\phi_z' y + \delta\vartheta' \omega, \\ \delta\gamma_{xy} &= \delta V' - \delta\phi_x' z - \delta\phi_z', \\ \delta\gamma_{xz} &= \delta W' + \delta\phi_x' y + \delta\phi_y'. \end{aligned} \quad (9.55)$$

The variation of the strain energy is

$$\begin{aligned} \delta\Pi &= \int_V \{E\varepsilon_{xx}\delta\varepsilon_{xx} + G\gamma_{xy}\delta\gamma_{xy} + G\gamma_{xz}\delta\gamma_{xz}\} dV \\ &= EA \int_l U' \delta U' dl + GA \int_l V' \delta V' dl + GA \int_l W' \delta W' dl + G(I_{yy} + I_{zz}) \int_l \phi_x' \delta\phi_x' dl + \\ &\quad EI_{yy} \int_l \phi_y' \delta\phi_y' dl + GA \int_l \phi_y \delta\phi_y dl + EI_{zz} \int_l \phi_z' \delta\phi_z' dl + GA \int_l \phi_z \delta\phi_z dl + \\ &\quad EI_{\omega\omega} \int_l \vartheta' \delta\vartheta' dl - GA \int_l (V' \delta\phi_z + \phi_z \delta V') dl + GA \int_l (W' \delta\phi_y + \phi_y \delta W') dl + \\ &\quad EI_{y\omega} \int_l (\phi_y' \delta\vartheta' + \vartheta' \delta\phi_y') dl - EI_{z\omega} \int_l (\phi_z' \delta\vartheta' + \vartheta' \delta\phi_z') dl, \end{aligned} \quad (9.56)$$

where the stiffness matrix can be read. Again the diagonal components are

$$\begin{aligned}
k_{\text{TRNS}} &= \frac{EA}{l}, \\
k_{\text{SHR}} &= \frac{GA}{l}, \\
k_{\text{RTx}} &= \frac{G(I_{yy} + I_{zz})}{l}, \\
k_{\text{RTy}} &= \frac{EI_{yy}}{l} + \frac{GA}{3}, \\
k_{\text{RTz}} &= \frac{EI_{zz}}{l} + \frac{GA}{3}, \\
k_{\text{TWST}} &= \frac{EI_{\omega\omega}}{l}.
\end{aligned} \tag{9.57}$$

From the expressions of the mass and stiffness matrix, the frequencies of the most common modes can be estimated. These are

1. The tensile and twisting modes with frequency $\omega = \frac{\sqrt{3}}{l} \sqrt{\frac{E}{\rho}}$.
2. The transverse shear and torsional mode with frequency $\omega = \frac{\sqrt{3}}{l} \sqrt{\frac{G}{\rho}}$.
3. The bending modes with frequencies $\omega = \sqrt{\frac{3E}{\rho l^2} + \frac{GA}{\rho I_{yy}}}$ and $\omega = \sqrt{\frac{3E}{\rho l^2} + \frac{GA}{\rho I_{zz}}}$.

Which one of these four that is the highest depends on the geometry of the beam element. In LS-DYNA the first of these frequencies is used for calculating a stable time step. We have found no reason for changing approach regarding this element.

6.2.3 Penalty on Twist

The twist is constrained using a penalty that is introduced in the strain energy as

$$\Pi_P = \frac{PEA}{2} \int_l (\phi'_x - \vartheta)^2 dl, \tag{9.58}$$

and the corresponding variation is

$$\delta \Pi = PEA \int_l (\phi'_x - \vartheta)(\delta \phi'_x - \delta \vartheta) dl. \tag{9.59}$$

The diagonal of the stiffness matrix is modified as follows

$$\begin{aligned}
k_{\text{RTx}} &= \frac{G(I_{yy} + I_{zz})}{l} + \frac{PEA}{l}, \\
k_{\text{TWST}} &= \frac{EI_{\omega\omega}}{l} + \frac{PEAl}{3}.
\end{aligned} \tag{9.60}$$

This increases the twist mode frequency to $\sqrt{\frac{3E}{\rho l^2} + \frac{PEA}{\rho I_{\omega\omega}}}$ and the torsional mode to

$$\frac{\sqrt{3}}{l} \sqrt{\frac{G(I_{yy} + I_{zz}) + PEA}{\rho}}. \tag{9.61}$$

Even though this gives an indication of a frequency increase we have made no modifications on the computation of the critical time step in an explicit analysis. We have used $P = 1$ in the implementation. This decision may have to be reconsidered depending on the choice of the parameter μ , in the end it will come down to trial and error from numerical simulations.

6.3 Generalization to Large Displacements

A generalization of the small displacement theory to nonlinear theory is quite straightforward. We have used a corotational formulation where the small strains in the linear theory are used directly as strain rates in the element system. We emphasize that the nonlinear beam formulation is obtained by simply replacing displacements for velocities and strains with strain rates in the previous section.

The nodal velocities for a beam element in the local system is written

$$\mathbf{v} = (v_x^1 \ v_y^1 \ v_z^1 \ \omega_x^1 \ \omega_y^1 \ \omega_z^1 \ \dot{\theta}^1 \ v_x^2 \ v_y^2 \ v_z^2 \ \omega_x^2 \ \omega_y^2 \ \omega_z^2 \ \dot{\theta}^2)^T, \tag{9.62}$$

where the superscript refers to the local node number. These are obtained by transforming the translational velocities and rotational velocities using the local to global transformation matrix q_{ij} . The strain rate – velocity matrix in the local system can be written

$$\mathbf{B}_0 = \begin{bmatrix} -l_0^{-1} & 0 & 0 & 0 & -l_0^{-1}z & l_0^{-1}y & -l_0^{-1}\omega & l_0^{-1} & 0 & 0 & 0 & l_0^{-1}z & -l_0^{-1}y & l_0^{-1}\omega \\ 0 & -l_0^{-1} & 0 & l_0^{-1}z & 0 & -\frac{1}{2} & 0 & 0 & l_0^{-1} & 0 & -l_0^{-1}z & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & -l_0^{-1} & -l_0^{-1}y & \frac{1}{2} & 0 & 0 & 0 & 0 & l_0^{-1} & l_0^{-1}y & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & -l_0^{-1} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & l_0^{-1} & 0 & 0 & -\frac{1}{2} \end{bmatrix}, \tag{9.63}$$

where l_0 is the beam length in the reference configuration, i.e., beginning of the time step. A corresponding matrix w.r.t. the current configuration is

(9.64)

$$\begin{bmatrix} -l^{-1} & 0 & 0 & 0 & -l^{-1}z & -l^{-1}y & -l^{-1}\omega & l^{-1} & 0 & 0 & 0 & l^{-1}z & -l^{-1}y & l^{-1}\omega \\ 0 & -l^{-1} & 0 & l^{-1}z & 0 & -\frac{1}{2} & 0 & 0 & l^{-1} & 0 & -l^{-1}z & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & -l^{-1} & -l^{-1}y & \frac{1}{2} & 0 & 0 & 0 & 0 & l^{-1} & l^{-1}y & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & -l^{-1} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & l^{-1} & 0 & 0 & -\frac{1}{2} \end{bmatrix},$$

where we use the current length of the beam. These matrices are evaluated in each integration point (x, y) of the cross section. To compute the strain rate in the local system we simply apply

$$\dot{\epsilon} = \mathbf{B}_0 \mathbf{v}, \quad (9.65)$$

which is then used to update the local stresses σ . The internal force vector is then assembled as

$$\mathbf{f} = \mathbf{B}^T \sigma. \quad (9.66)$$

Finally the internal force is transformed to the global system using the transformation matrix.

To compute the stiffness matrix for implicit we neglect the geometric contribution and just apply

$$\mathbf{K} = \mathbf{B}^T \mathbf{C} \mathbf{B}, \quad (9.67)$$

where \mathbf{C} is the material tangent modulus. Again the matrix must be transformed to the global system before used in the implicit solver.

10

Belytschko-Lin-Tsay Shell

The Belytschko-Lin-Tsay shell element ([Belytschko and Tsay 1981], [Belytschko et al., 1984a]) was implemented in LS-DYNA as a computationally efficient alternative to the Hughes-Liu shell element. For a shell element with five through thickness integration points, the Belytschko-Lin-Tsay shell elements requires 725 mathematical operations compared to 4050 operations for the under integrated Hughes-Liu element. The selectively reduced integration formulation of the explicit Hughes-Liu element requires 35,350 mathematical operations. Because of its computational efficiency, the Belytschko-Lin-Tsay shell element is usually the shell element formulation of choice. For this reason, it has become the default shell element formulation for explicit calculations.

The Belytschko-Lin-Tsay shell element is based on a combined co-rotational and velocity-strain formulation. The efficiency of the element is obtained from the mathematical simplifications that result from these two kinematical assumptions. The co-rotational portion of the formulation avoids the complexities of nonlinear mechanics by embedding a coordinate system in the element. The choice of velocity-strain or rate-of-deformation in the formulation facilitates the constitutive evaluation, since the conjugate stress is the physical Cauchy stress. We closely follow the notation of Belytschko, Lin, and Tsay in the following development.

10.1 Co-rotational Coordinates

The midsurface of the quadrilateral shell element, or reference surface, is defined by the location of the element's four corner nodes. An embedded element coordinate system (see Figure 7.1) that deforms with the element is defined in terms of these nodal coordinates. Then the procedure for constructing the co-rotational coordinate system begins by calculating a unit vector normal to the main diagonal of the element:

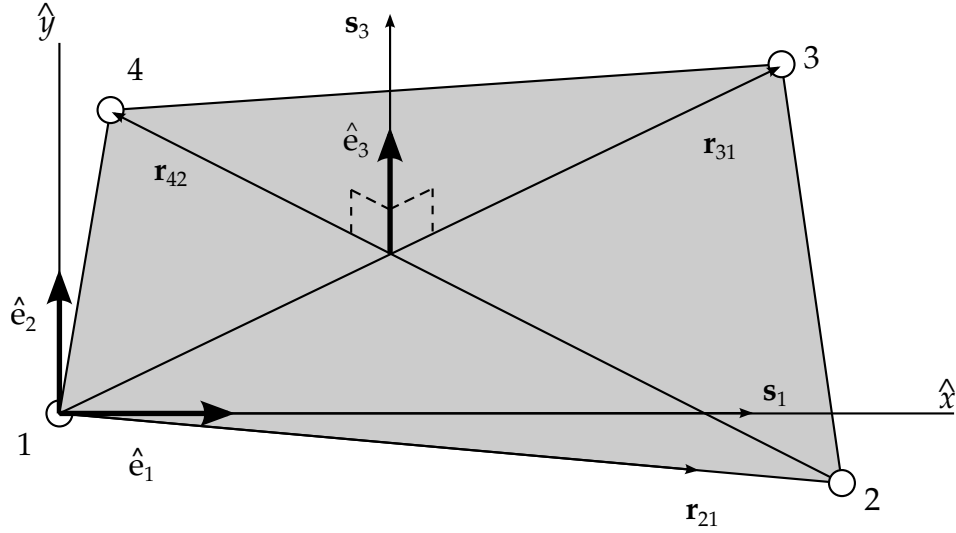


Figure 10.1. Construction of element coordinate system is shown.

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{s}_3}{\|\mathbf{s}_3\|}, \quad (10.1)$$

$$\|\mathbf{s}_3\| = \sqrt{[\mathbf{s}_3]_1^2 + [\mathbf{s}_3]_2^2 + [\mathbf{s}_3]_3^2}, \quad (10.2)$$

$$\mathbf{s}_3 = \mathbf{r}_{31} \times \mathbf{r}_{42}, \quad (10.3)$$

where the superscript caret ($\hat{\cdot}$) is used to indicate the local (element) coordinate system.

It is desired to establish the local x axis \hat{x} approximately along the element edge between nodes 1 and 2. This definition is convenient for interpreting the element stresses, which are defined in the local $\hat{x} - \hat{y}$ coordinate system. The procedure for constructing this unit vector is to define a vector \mathbf{s}_1 that is nearly parallel to the vector \mathbf{r}_{21} , viz.

$$\mathbf{s}_1 = \mathbf{r}_{21} - (\mathbf{r}_{21} \cdot \hat{\mathbf{e}}_3)\hat{\mathbf{e}}_3, \quad (10.4)$$

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{s}_1}{\|\mathbf{s}_1\|}. \quad (10.5)$$

The remaining unit vector is obtained from the vector cross product

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1. \quad (10.6)$$

If the four nodes of the element are coplanar, then the unit vectors $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ are tangent to the midplane of the shell and $\hat{\mathbf{e}}_3$ is in the fiber direction. As the element deforms, an angle may develop between the actual fiber direction and the unit normal $\hat{\mathbf{e}}_3$. The magnitude of this angle may be characterized as

$$|\hat{\mathbf{e}}_3 \cdot \mathbf{f} - 1| < \delta, \quad (10.7)$$

where \mathbf{f} is the unit vector in the fiber direction and the magnitude of δ depends on the magnitude of the strains. According to Belytschko et al., for most engineering applications, acceptable values of δ are on the order of 10-2 and if the condition presented in Equation (10.7) is met, then the difference between the rotation of the co-rotational coordinates $\hat{\mathbf{e}}$ and the material rotation should be small.

The global components of this co-rotational triad define a transformation matrix between the global and local element coordinate systems. This transformation operates on vectors with global components $\mathbf{A} = (A_x \ A_y \ A_z)$ and element coordinate components $\hat{\mathbf{A}} = (\hat{A}_x \ \hat{A}_y \ \hat{A}_z)$, and is defined as:

$$\mathbf{A} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = \boldsymbol{\mu} \hat{\mathbf{A}} = \mathbf{q}^T \hat{\mathbf{A}}, \quad (10.8)$$

where e_{ix}, e_{iy}, e_{iz} are the global components of the element coordinate unit vectors. The inverse transformation is defined by the matrix transpose, i.e.,

$$\hat{\mathbf{A}} = \boldsymbol{\mu}^T \mathbf{A}. \quad (10.9)$$

10.2 Velocity-Strain Displacement Relations

The above small rotation condition, Equation (10.7), does not restrict the magnitude of the element's rigid body rotations. Rather, the restriction is placed on the out-of-plane deformations, and, thus, on the element strain. Consistent with this restriction on the magnitude of the strains, the velocity-strain displacement relations used in the Belytschko-Lin-Tsay shell are also restricted to small strains.

As in the Hughes-Liu shell element, the displacement of any point in the shell is partitioned into a midsurface displacement (nodal translations) and a displacement associated with rotations of the element's fibers (nodal rotations). The Belytschko-Lin-Tsay shell element uses the Mindlin [1951] theory of plates and shells to partition the velocity of any point in the shell as:

$$\mathbf{v} = \mathbf{v}^m - \hat{z} \mathbf{e}_3 \times \boldsymbol{\theta}, \quad (10.10)$$

where \mathbf{v}^m is the velocity of the mid-surface, $\boldsymbol{\theta}$ is the angular velocity vector, and \hat{z} is the distance along the fiber direction (thickness) of the shell element. The corresponding co-rotational components of the velocity strain (rate of deformation) are given by

$$\hat{d}_{ij} = \frac{1}{2} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right). \quad (10.11)$$

Substitution of Equation (10.10) into the above yields the following velocity-strain relations:

$$\hat{d}_x = \frac{\partial \hat{v}_x^m}{\partial \hat{x}} + \hat{z} \frac{\partial \hat{\theta}_y}{\partial \hat{x}}, \quad (10.12)$$

$$\hat{d}_y = \frac{\partial \hat{v}_y^m}{\partial \hat{y}} - \hat{z} \frac{\partial \hat{\theta}_x}{\partial \hat{y}}, \quad (10.13)$$

$$2\hat{d}_{xy} = \frac{\partial \hat{v}_x^m}{\partial \hat{y}} + \frac{\partial \hat{v}_y^m}{\partial \hat{x}} + \hat{z} \left(\frac{\partial \hat{\theta}_y}{\partial \hat{y}} - \frac{\partial \hat{\theta}_x}{\partial \hat{x}} \right), \quad (10.14)$$

$$2\hat{d}_{yz} = \frac{\partial \hat{v}_z^m}{\partial \hat{y}} - \hat{\theta}_x, \quad (10.15)$$

$$2\hat{d}_{xz} = \frac{\partial \hat{v}_z^m}{\partial \hat{x}} + \hat{\theta}_y. \quad (10.16)$$

The above velocity-strain relations need to be evaluated at the quadrature points within the shell. Standard bilinear nodal interpolation is used to define the mid-surface velocity, angular velocity, and the element's coordinates (isoparametric representation). These interpolations relations are given by

$$\begin{aligned} \mathbf{v}^m &= N_I(\xi, \eta) \mathbf{v}_I, \\ \boldsymbol{\theta}^m &= N_I(\xi, \eta) \boldsymbol{\theta}_I, \\ \mathbf{x}^m &= N_I(\xi, \eta) \mathbf{x}_I. \end{aligned} \quad (10.17)$$

where the subscript I is summed over all the nodes of the element and the nodal velocities are obtained by differentiating the nodal coordinates with respect to time, i.e., $v_I = \dot{x}_I$. The bilinear shape functions are

$$N_1 = \frac{1}{4} (1 - \xi)(1 - \eta), \quad (10.18)$$

$$N_2 = \frac{1}{4} (1 + \xi)(1 - \eta), \quad (10.19)$$

$$N_3 = \frac{1}{4} (1 + \xi)(1 + \eta), \quad (10.20)$$

$$N_4 = \frac{1}{4} (1 - \xi)(1 + \eta). \quad (10.21)$$

The velocity-strains at the center of the element, i.e., at $\xi = 0$, and $\eta = 0$, are obtained by substitution of the above relations into the previously defined velocity-strain displacement relations, [Equations \(10.12\)](#) and [\(10.16\)](#). After some algebra, this yields

$$\hat{d}_x = B_{1I} \hat{v}_{xI} + \hat{z} B_{1I} \hat{\theta}_{yI}, \quad (10.22a)$$

$$\hat{d}_y = B_{2I} \hat{v}_{yI} - \hat{z} B_{2I} \hat{\theta}_{xI}, \quad (10.22b)$$

$$2\hat{d}_{xy} = B_{2I} \hat{v}_{xI} + B_{1I} \hat{v}_{yI} + \hat{z} (B_{2I} \hat{\theta}_{yI} - B_{1I} \hat{\theta}_{xI}), \quad (10.22c)$$

$$2\hat{d}_{xz} = B_{1I} \hat{v}_{zI} + N_I \hat{\theta}_{yI}, \quad (10.22d)$$

$$2\hat{d}_{yz} = B_{2I} \hat{v}_{zI} - N_I \hat{\theta}_{xI}, \quad (10.22e)$$

$$B_{1I} = \frac{\partial N_I}{\partial \hat{x}}, \quad (10.22f)$$

$$B_{2I} = \frac{\partial N_I}{\partial \hat{y}}. \quad (10.22g)$$

The shape function derivatives B_{aI} are also evaluated at the center of the element, i.e., at $\xi = 0$, and $\eta = 0$.

10.3 Stress Resultants and Nodal Forces

After suitable constitutive evaluations using the above velocity-strains, the resulting stresses are integrated through the thickness of the shell to obtain local resultant forces and moments. The integration formula for the resultants are

$$\hat{f}_{\alpha\beta}^R = \int \hat{\sigma}_{\alpha\beta} d\hat{z}, \quad (10.23)$$

$$\hat{m}_{\alpha\beta}^R = - \int \hat{z} \hat{\sigma}_{\alpha\beta} d\hat{z}, \quad (10.24)$$

where the superscript, R , indicates a resultant force or moment, and the Greek subscripts emphasize the limited range of the indices for plane stress plasticity.

The above element-centered force and moment resultants are related to the local nodal forces and moments by invoking the principle of virtual power and integrating with a one-point quadrature. The relations obtained in this manner are

$$\hat{f}_{xI} = A(B_{1I} \hat{f}_{xx}^R + B_{2I} \hat{f}_{xy}^R), \quad (10.25)$$

$$\hat{f}_{yI} = A(B_{2I} \hat{f}_{yy}^R + B_{1I} \hat{f}_{xy}^R), \quad (10.26)$$

$$\hat{f}_{zI} = A\kappa(B_{1I} \hat{f}_{xz}^R + B_{2I} \hat{f}_{yz}^R), \quad (10.27)$$

$$\hat{m}_{xI} = A \left(B_{2I} \hat{m}_{yy}^R + B_{1I} \hat{m}_{xy}^R - \frac{\kappa}{4} \hat{f}_{yz}^R \right), \quad (10.28)$$

$$\hat{m}_{yI} = -A \left(B_{1I} \hat{m}_{xx}^R + B_{2I} \hat{m}_{xy}^R - \frac{\kappa}{4} \hat{f}_{xz}^R \right), \quad (10.29)$$

$$\hat{m}_{zI} = 0, \quad (10.30)$$

where A is the area of the element, and κ is the shear factor from the Mindlin theory. In the Belytschko-Lin-Tsay formulation, κ is used as a penalty parameter to enforce the Kirchhoff normality condition as the shell becomes thin.

The above local nodal forces and moments are then transformed to the global coordinate system using the transformation relations given previously as [Equation \(10.8\)](#). The global nodal forces and moments are then appropriately summed over all the nodes and the global equations of motion are solved for the next increment in nodal accelerations.

10.4 Hourglass Control (Belytschko-Lin-Tsay)

In part, the computational efficiency of the Belytschko-Lin-Tsay and the under integrated Hughes-Liu shell elements are derived from their use of one-point quadrature in the plane of the element. To suppress the hourglass deformation modes that accompany one-point quadrature, hourglass viscosity stresses are added to the physical stresses at the local element level. The discussion of the hourglass control that follows pertains to the Hughes-Liu and the membrane elements as well.

The hourglass control used by Belytschko et al., extends an earlier derivation by Flanagan and Belytschko [1981], (see also Kosloff and Frazier [1978], Belytschko and Tsay [1983]). The hourglass shape vector, τ_I , is defined as

$$\tau_I = \mathbf{h}_I - (\mathbf{h}_I \hat{\mathbf{x}}_{aI}) \mathbf{B}_{aI}, \quad (10.31)$$

where

$$\mathbf{h} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}, \quad (10.32)$$

is the basis vector that generates the deformation mode that is neglected by one-point quadrature. In [Equation \(10.31\)](#) and the remainder of this subsection, the Greek subscripts have a range of 2, e.g., $\hat{\mathbf{x}}_{aI} = (\hat{x}_{1I}, \hat{x}_{2I}) = (\hat{x}_I, \hat{y}_I)$.

The hourglass shape vector then operates on the generalized displacements, in a manner similar to Equations (7.11a - e), to produce the generalized hourglass strain rates

$$\dot{\mathbf{q}}_{\alpha}^B = \tau_I \hat{\theta}_{\alpha I}, \quad (10.33)$$

$$\dot{\mathbf{q}}_3^B = \tau_I \hat{v}_{zI}, \quad (10.34)$$

$$\dot{\mathbf{q}}_{\alpha}^M = \tau_I \hat{v}_{\alpha I}, \quad (10.35)$$

where the superscripts **B** and **M** denote bending and membrane modes, respectively. The corresponding hourglass stress rates are then given by

$$\dot{Q}_\alpha^B = \frac{r_\theta Et^3 A}{192} B_{\beta I} B_{\beta I} \dot{q}_\alpha^B, \quad (10.36)$$

$$\dot{Q}_3^B = \frac{r_w \kappa G t^3 A}{12} B_{\beta I} B_{\beta I} \dot{q}_3^B, \quad (10.37)$$

$$\dot{Q}_\alpha^M = \frac{r_m Et A}{8} B_{\beta I} B_{\beta I} \dot{q}_\alpha^M, \quad (10.38)$$

where t is the shell thickness and the parameters, r_θ , r_w , and r_m are generally assigned values between 0.01 and 0.05.

Finally, the hourglass stresses, which are updated from the stress rates in the usual way, i.e.,

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta t \dot{\mathbf{Q}}, \quad (10.39)$$

and the hourglass resultant forces are then

$$\hat{m}_{\alpha I}^H = \tau_I Q_\alpha^B, \quad (10.40)$$

$$\hat{f}_{3I}^H = \tau_I Q_3^B, \quad (10.41)$$

$$\hat{f}_{\alpha I}^H = \tau_I Q_\alpha^M, \quad (10.42)$$

where the superscript H emphasizes that these are internal force contributions from the hourglass deformations. These hourglass forces are added directly to the previously determined local internal forces due to deformations Equations (7.14a - f). These force vectors are orthogonalized with respect to rigid body motion.

10.5 Hourglass Control (Englemann and Whirley)

Englemann and Whirley [1991] developed an alternative hourglass control, which they implemented in the framework of the Belytschko, Lin, and Tsay shell element. We will briefly highlight their procedure here that has proven to be cost effective—only twenty percent more expensive than the default control.

In the hourglass procedure, the in-plane strain field (subscript p) is decomposed into the one point strain field plus the stabilization strain field:

$$\dot{\tilde{\epsilon}}_p = \dot{\tilde{\epsilon}}_p^0 + \dot{\tilde{\epsilon}}_p^s, \quad (10.43)$$

$$\dot{\tilde{\epsilon}}_p^s = \mathbf{W}_m \dot{\mathbf{q}}_m + z \mathbf{W}_b \dot{\mathbf{q}}_b. \quad (10.44)$$

Here, \mathbf{W}_m and \mathbf{W}_b play the role of stabilization strain velocity operators for membrane and bending:

$$\mathbf{W}_m = \begin{bmatrix} f_1^p(\zeta, \eta) & f_4^p(\zeta, \eta) \\ f_2^p(\zeta, \eta) & f_5^p(\zeta, \eta) \\ f_3^p(\zeta, \eta) & f_6^p(\zeta, \eta) \end{bmatrix}, \quad (10.45)$$

$$\mathbf{W}_b = \begin{bmatrix} -f_4^p(\zeta, \eta) & f_1^p(\zeta, \eta) \\ -f_5^p(\zeta, \eta) & f_2^p(\zeta, \eta) \\ -f_6^p(\zeta, \eta) & f_3^p(\zeta, \eta) \end{bmatrix}, \quad (10.46)$$

where the terms $f_i^p(\zeta, \eta)$ $i = 1, 2, \dots, 6$, are rather complicated and the reader is referred to the reference [Englemann and Whirley, 1991].

To obtain the transverse shear assumed strain field, the procedure given in [Bathe and Dvorkin, 1984] is used. The transverse shear strain field can again be decomposed into the one point strain field plus the stabilization field:

$$\dot{\tilde{\epsilon}}_s = \dot{\tilde{\epsilon}}_s^0 + \dot{\tilde{\epsilon}}_s^s, \quad (10.47)$$

that is related to the hourglass velocities by

$$\dot{\tilde{\epsilon}}_s^s = \mathbf{W}_s \dot{\mathbf{q}}_s, \quad (10.48)$$

where the transverse shear stabilization strain-velocity operator \mathbf{W}_s is given by

$$\mathbf{W}_s = \begin{bmatrix} f_1^s(\zeta, \eta) & -g_1^s \zeta & g_2^s \eta & g_3^s \zeta & g_3^s \eta \\ f_2^s(\zeta, \eta) & g_4^s \zeta & g_4^s \eta & -g_2^s \zeta & g_1^s \eta \end{bmatrix}. \quad (10.49)$$

Again, the coefficients $f_1^s(\zeta, \eta)$ and g_1^s are defined in the reference.

In their formulation, the hourglass forces are related to the hourglass velocity field through an incremental hourglass constitutive equation derived from an additive decomposition of the stress into a “one-point stress,” plus a “stabilization stress.” The integration of the stabilization stress gives a resultant constitutive equation relating hourglass forces to hourglass velocities. The in-plane and transverse stabilization stresses are updated according to:

$$\boldsymbol{\tau}_s^{s,n+1} = \boldsymbol{\tau}_s^{s,n} + \Delta t c_s \mathbf{C}_s \dot{\tilde{\epsilon}}_s^s, \quad (10.50)$$

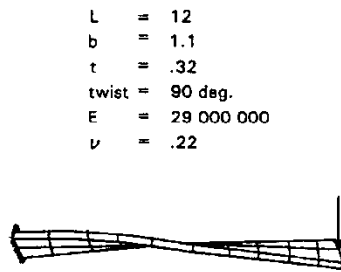
$$\tau_s^{s,n+1} = \tau_s^{s,n} + \Delta t c_s C_s \dot{\tilde{\epsilon}}_s^s, \quad (10.51)$$

where the tangent matrix is the product of a matrix \mathbf{C} , which is constant within the shell domain, and a scalar c that is constant in the plane but may vary through the thickness.

The stabilization stresses can now be used to obtain the hourglass forces:

$$\mathbf{Q}_m = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_A \mathbf{W}_m^T \boldsymbol{\tau}_p^s dA dz, \quad (10.52)$$

Twisted beam problem



Displacement-time history

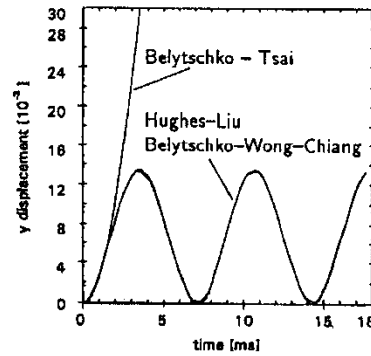


Figure 10.2. The twisted beam problem fails with the Belytschko-Tsay shell element.

$$Q_b = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_A \mathbf{W}_b^T \boldsymbol{\tau}_p^s dA dz, \quad (10.53)$$

$$Q_s = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_A \mathbf{W}_s^T \boldsymbol{\tau}_s^s dA dz. \quad (10.54)$$

10.6 Belytschko-Wong-Chiang Improvements

Since the Belytschko-Tsay element is based on a perfectly flat geometry, warpage is not considered. Although this generally poses no major difficulties and provides for an efficient element, incorrect results in the twisted beam problem, See Figure 7.2, are obtained where the nodal points of the elements used in the discretization are not coplanar. The Hughes-Liu shell element considers non-planar geometry and gives good results on the twisted beam, but is relatively expensive. The effect of neglecting warpage in typical a application cannot be predicted beforehand and may lead to less than accurate results, but the latter is only speculation and is difficult to verify in practice. Obviously, it would be better to use shells that consider warpage if the added costs are reasonable and if this unknown effect is eliminated. In this section we briefly describe the simple and computationally inexpensive modifications necessary in the Belytschko-Tsay shell to include the warping stiffness. The improved transverse shear treatment is also described which is necessary for the element to pass the Kirchhoff patch test. Readers are directed to the references [Belytschko, Wong, and Chang 1989, 1992] for an in depth theoretical background.

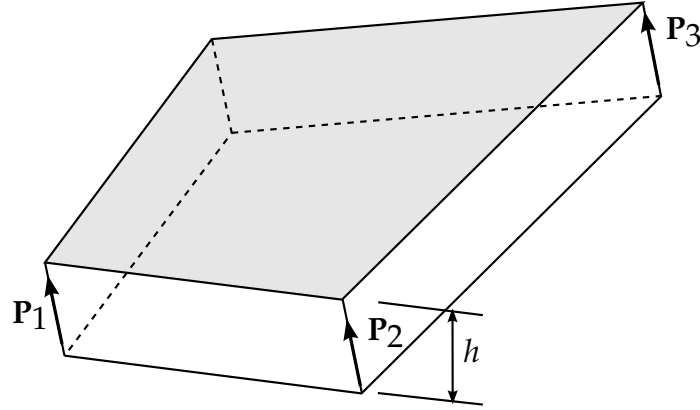


Figure 10.3. Nodal fiber vectors \mathbf{p}_1 , \mathbf{p}_2 , and \mathbf{p}_3 , where h is the thickness.

In order to include warpage in the formulation it is convenient to define nodal fiber vectors as shown in Figure 7.3. The geometry is interpolated over the surface of the shell from:

$$x = x^m + \bar{\zeta} p = (x_I + \bar{\zeta} p_I) N_I(\xi, \eta), \quad (10.55)$$

where

$$\bar{\zeta} = \frac{\zeta h}{2}. \quad (10.56)$$

The in plane strain components are given by:

$$d_{xx} = b_{xI} \hat{v}_{xI} + \bar{\zeta} (b_{xI}^c \hat{v}_{xI} + b_{xI} \dot{p}_{xI}), \quad (10.57)$$

$$d_{yy} = b_{yI} \hat{v}_{yI} + \bar{\zeta} (b_{yI}^c \hat{v}_{yI} + b_{yI} \dot{p}_{yI}), \quad (10.58)$$

$$d_{xy} = \frac{1}{2} b_{xI} \hat{v}_{yI} + b_{yI} \hat{v}_{xI} + \bar{\zeta} (b_{xI}^c \hat{v}_{yI} + b_{xI} \dot{p}_{yI} + b_{yI}^c \hat{v}_{xI} + b_{yI} \dot{p}_{xI}). \quad (10.59)$$

The coupling terms come in through b_{iI}^c : which is defined in terms of the components of the fiber vectors as:

$$\begin{Bmatrix} b_{xI}^c \\ b_{yI}^c \end{Bmatrix} = \begin{bmatrix} p_{\hat{y}2} - p_{\hat{y}4} & p_{\hat{y}3} - p_{\hat{y}1} & p_{\hat{y}4} - p_{\hat{y}2} & p_{\hat{y}1} - p_{\hat{y}3} \\ p_{\hat{x}2} - p_{\hat{x}4} & p_{\hat{x}3} - p_{\hat{x}1} & p_{\hat{x}4} - p_{\hat{x}2} & p_{\hat{x}1} - p_{\hat{x}3} \end{bmatrix}, \quad (10.60)$$

For a flat geometry the normal vectors are identical and no coupling can occur. Two methods are used by Belytschko for computing b_{iI}^c and the reader is referred to his papers for the details. Both methods have been tested in LS-DYNA and comparable results were obtained.

The transverse shear strain components are given as

$$\hat{\gamma}_{xz} = -N_I(\xi, \eta) \bar{\theta}_{\hat{y}I}, \quad (10.61)$$

$$\hat{\gamma}_{yz} = -N_I(\xi, \eta) \bar{\theta}_{\hat{x}I}, \quad (10.62)$$

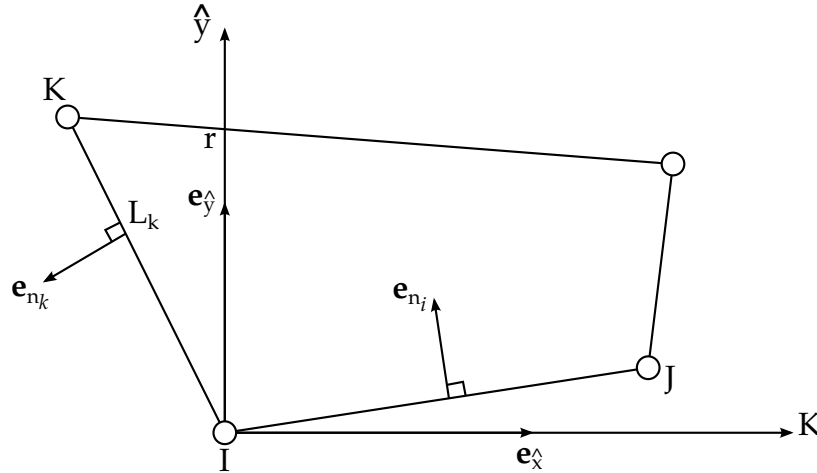


Figure 10.4. Vector and edge definitions for computing the transverse shear strain components.

where the nodal rotational components are defined as:

$$\bar{\theta}_{\hat{x}I} = (\mathbf{e}_n^I \cdot \mathbf{e}_{\hat{x}}) \bar{\theta}_n^I + (\mathbf{e}_n^K \cdot \mathbf{e}_{\hat{x}}) \bar{\theta}_n^K, \quad (10.63)$$

$$\bar{\theta}_{\hat{y}I} = (\mathbf{e}_n^I \cdot \mathbf{e}_{\hat{y}}) \bar{\theta}_n^I + (\mathbf{e}_n^K \cdot \mathbf{e}_{\hat{y}}) \bar{\theta}_n^K, \quad (10.64)$$

$$\bar{\theta}_n^I = \frac{1}{2} (\theta_{nI}^I + \theta_{nJ}^I) + \frac{1}{L^{IJ}} (\hat{v}_{zJ} - \hat{v}_{zI}), \quad (10.65)$$

where the subscript n refers to the normal component of side I as seen in Figure 7.3 and L^{IJ} is the length of side IJ .

11

Triangular Shells

11.1 C^0 Triangular Shell

The C^0 shell element due to Kennedy, Belytschko, and Lin [1986] has been implemented as a computationally efficient triangular element complement to the Belytschko-Lin-Tsay quadrilateral shell element ([Belytschko and Tsay 1981], [Belytschko et al., 1984a]). For a shell element with five through-the-thickness integration points, the element requires 649 mathematical operations (the Belytschko-Lin-Tsay quadrilateral shell element requires 725 mathematical operations) compared to 1417 operations for the Marchertas-Belytschko triangular shell [Marchertas and Belytschko 1974] (referred to as the BCIZ [Bazeley, Cheung, Irons, and Zienkiewicz 1965] triangular shell element in the DYNA3D user's manual).

Triangular shell elements are offered as optional elements primarily for compatibility with local user grid generation and refinement software. Many computer aided design (CAD) and computer aided manufacturing (CAM) packages include finite element mesh generators, and most of these mesh generators use triangular elements in the discretization. Similarly, automatic mesh refinement algorithms are typically based on triangular element discretization. Also, triangular shell element formulations are not subject to zero energy modes inherent in quadrilateral element formulations.

The triangular shell element's origins are based on the work of Belytschko et al., [Belytschko, Stolarski, and Carpenter 1984b] where the linear performance of the shell was demonstrated. Because the triangular shell element formulations parallels closely the formulation of the Belytschko-Lin-Tsay quadrilateral shell element presented in the previous section (Section 7), the following discussion is limited to items related specifically to the triangular shell element.

11.1.1 Co-rotational Coordinates

The mid-surface of the triangular shell element, or reference surface, is defined by the location of the element's three nodes. An embedded element coordinate system (see

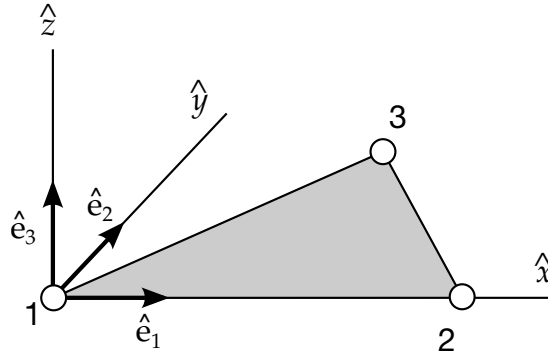


Figure 11.1. Local element coordinate system for C0 shell element.

Figure 11.1) that deforms with the element is defined in terms of these nodal coordinates. The procedure for constructing the co-rotational coordinate system is simpler than the corresponding procedure for the quadrilateral, because the three nodes of the triangular element are guaranteed coplanar.

The local x-axis \hat{x} is directed from node 1 to 2. The element's normal axis \hat{z} is defined by the vector cross product of a vector along \hat{x} with a vector constructed from node 1 to node 3. The local y-axis \hat{y} is defined by a unit vector cross product of \hat{e}_3 with \hat{e}_1 , which are the unit vectors in the \hat{z} directions, respectively. As in the case of the quadrilateral element, this triad of co-rotational unit vectors defines a transformation between the global and local element coordinate systems. See Equations (7.5 a, b).

11.1.2 Velocity-Strain Relations

As in the Belytschko-Lin-Tsay quadrilateral shell element, the displacement of any point in the shell is partitioned into a mid-surface displacement (nodal translations) and a displacement associated with rotations of the element's fibers (nodal rotations). The Kennedy-Belytschko-Lin triangular shell element also uses the Mindlin [Mindlin 1951] theory of plates and shells to partition the velocity of any point in the shell (recall Equation (7.6)):

$$\mathbf{v} = \mathbf{v}^m - \hat{z} \mathbf{e}_3 \times \boldsymbol{\theta}, \quad (11.1)$$

where \mathbf{v}^m is the velocity of the mid-surface, $\boldsymbol{\theta}$ is the angular velocity vector, and \hat{z} is the distance along the fiber direction (thickness) of the shell element. The corresponding co-rotational components of the velocity strain (rate of deformation) were given previously in Equation (7.11 a - e).

Standard linear nodal interpolation is used to define the midsurface velocity, angular velocity, and the element's coordinates (isoparametric representation). These interpolation functions are the area coordinates used in triangular element formulations. Substitution of the nodally interpolated velocity fields into the velocity-strain relations (see Belytschko et al., for details), leads to strain rate-velocity relations of the form

$$\hat{\mathbf{d}} = \mathbf{B} \hat{\mathbf{v}}. \quad (11.2)$$

It is convenient to partition the velocity strains and the \mathbf{B} matrix into membrane and bending contributions. The membrane relations are given by

$$\begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \\ 2\hat{d}_{xy} \end{Bmatrix}^M = \frac{1}{\hat{x}_2\hat{y}_3} \begin{bmatrix} \hat{y}_3 & 0 & \hat{y}_3 & 0 & 0 & 0 \\ 0 & \hat{x}_3 - \hat{x}_2 & 0 & -\hat{x}_3 & 0 & \hat{x}_2 \\ \hat{x}_3 - \hat{x}_2 & -\hat{y}_3 & -\hat{x}_3 & \hat{y}_3 & \hat{x}_2 & 0 \end{bmatrix} \begin{Bmatrix} \hat{v}_{x1} \\ \hat{v}_{y1} \\ \hat{v}_{x2} \\ \hat{v}_{y2} \\ \hat{v}_{x3} \\ \hat{v}_{y3} \end{Bmatrix}, \quad (11.3)$$

$$\hat{\mathbf{d}}^M = \mathbf{B}_M \hat{\mathbf{v}}, \quad (11.4)$$

$$\begin{Bmatrix} \hat{\kappa}_x \\ \hat{\kappa}_y \\ 2\hat{\kappa}_{xy} \end{Bmatrix} = \frac{-1}{\hat{x}_2\hat{y}_3} \begin{bmatrix} 0 & -\hat{y}_3 & 0 & \hat{y}_3 & 0 & 0 \\ \hat{x}_3 - \hat{x}_2 & 0 & \hat{x}_3 & 0 & -\hat{x}_2 & 0 \\ \hat{y}_3 & \hat{x}_3 - \hat{x}_2 & -\hat{y}_3 & -\hat{x}_3 & 0 & \hat{x}_2 \end{bmatrix} \begin{Bmatrix} \hat{\theta}_{x1} \\ \hat{\theta}_{y1} \\ \hat{\theta}_{x2} \\ \hat{\theta}_{y2} \\ \hat{\theta}_{x3} \\ \hat{\theta}_{y3} \end{Bmatrix}, \quad (11.5)$$

or

$$\hat{\mathbf{\kappa}}^M = \mathbf{B}_M \hat{\mathbf{\theta}}^{\text{def}}. \quad (11.6)$$

The local element velocity strains are then obtained by combining the above two relations:

$$\begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \\ 2\hat{d}_{xy} \end{Bmatrix} = \begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \\ 2\hat{d}_{xy} \end{Bmatrix}^M - \hat{z} \begin{Bmatrix} \hat{\kappa}_x \\ \hat{\kappa}_y \\ 2\hat{\kappa}_{xy} \end{Bmatrix} = \hat{\mathbf{d}}^M - \hat{z} \hat{\mathbf{\kappa}}. \quad (11.7)$$

The remaining two transverse shear strain rates are given by

$$\begin{Bmatrix} 2\hat{d}_{xz} \\ 2\hat{d}_{yz} \end{Bmatrix} = \frac{1}{6 \hat{x}_2 \hat{y}_3} \begin{bmatrix} -\hat{y}_3^2 & \hat{y}_3 (2 \hat{x}_2 + \hat{x}_3) & \hat{y}_3^2 & \hat{y}_3 (3 \hat{x}_2 - \hat{x}_3) & 0 & \hat{x}_2 \hat{y}_3 \\ \hat{y}_3 (\hat{x}_2 - 2 \hat{x}_3) & \hat{x}_2^2 - \hat{x}_3^2 & -\hat{y}_3^2 (\hat{x}_2 + \hat{x}_3) & \hat{x}_3 (\hat{x}_3 - 2 \hat{x}_2) & -3 \hat{x}_2 \hat{y}_3 & \hat{x}_2 (2 \hat{x}_3 - \hat{x}_2) \end{bmatrix} \begin{Bmatrix} \hat{v}_{x1} \\ \hat{v}_{y1} \\ \hat{v}_{x2} \\ \hat{v}_{y2} \\ \hat{v}_{x3} \\ \hat{v}_{y3} \end{Bmatrix} \quad (11.8)$$

$$\begin{Bmatrix} \hat{\theta}_{x1} \\ \hat{\theta}_{y1} \\ \hat{\theta}_{x2} \\ \hat{\theta}_{y2} \\ \hat{\theta}_{x3} \\ \hat{\theta}_{y3} \end{Bmatrix}^{\text{def}},$$

$$\hat{\mathbf{d}}^S = \mathbf{B}_S \hat{\boldsymbol{\theta}}^{\text{def}}. \quad (11.9)$$

All of the above velocity-strain relations have been simplified by using one-point quadrature.

In the above relations, the angular velocities $\hat{\boldsymbol{\theta}}^{\text{def}}$ are the deformation component of the angular velocity $\hat{\boldsymbol{\theta}}$ obtained by subtracting the portion of the angular velocity due to rigid body rotation, i.e.,

$$\hat{\boldsymbol{\theta}}^{\text{def}} = \hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^{\text{rig}}, \quad (11.10)$$

The two components of the rigid body angular velocity are given by

$$\hat{\theta}_y^{\text{rig}} = \frac{\hat{v}_{z1} - \hat{v}_{z2}}{\hat{x}_2}, \quad (11.11)$$

$$\hat{\theta}_x^{\text{rig}} = \frac{(\hat{v}_{z3} - \hat{v}_{z1})\hat{x}_2 - (\hat{v}_{z2} - \hat{v}_{z1})\hat{x}_3}{\hat{x}_2\hat{y}_3}. \quad (11.12)$$

The first of the above two relations is obtained by considering the angular velocity of the local x-axis about the local y-axis. Referring to [Figure 11.1](#), by construction nodes 1 and 2 lie on the local x-axis and the distance between the nodes is \hat{x}_2 i.e., the \hat{x} distance from node 2 to the local coordinate origin at node 1. Thus the difference in the nodal \hat{z} velocities divided by the distance between the nodes is an average measure of the rigid body rotation rate about the local y-axis.

The second relation is conceptually identical, but is implemented in a slightly different manner due to the arbitrary location of node 3 in the local coordinate system. Consider the two local element configurations shown in [Figure 11.2](#). For the leftmost configuration, where node 3 is the local y-axis, the rigid body rotation rate about the local x-axis is given by

$$\hat{\theta}_{x\text{-left}}^{\text{rig}} = \frac{\hat{v}_{z3} - \hat{v}_{z1}}{\hat{y}_3}, \quad (11.13)$$

and for the rightmost configuration the same rotation rate is given by

$$\hat{\theta}_{x\text{-right}}^{\text{rig}} = \frac{\hat{v}_{z3} - \hat{v}_{z2}}{\hat{y}_3}. \quad (11.14)$$

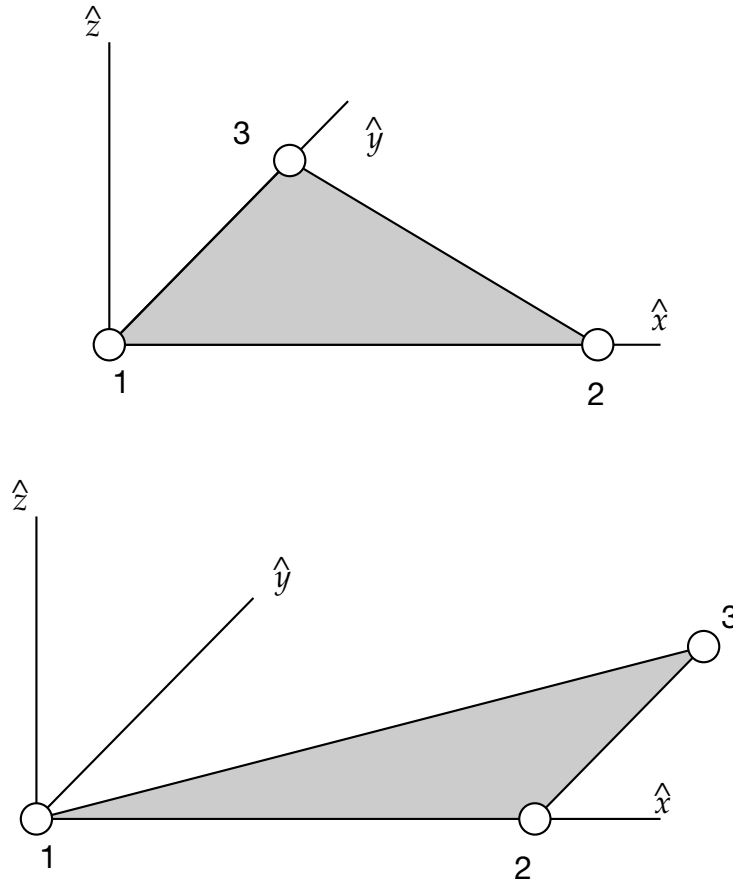


Figure 11.2. Element configurations with node 3 aligned with node 1 (left) and node 3 aligned with node 2 (right).

Although both of these relations yield the average rigid body rotation rate, the selection of the correct relation depends on the configuration of the element, i.e., on the location of node 3. Since every element in the mesh could have a configuration that is different in general from either of the two configurations shown in [Figure 11.2](#), a more robust relation is needed to determine the average rigid body rotation rate about the local x-axis. In most typical grids, node 3 will be located somewhere between the two configurations shown in [Figure 11.2](#). Thus a linear interpolation between these two rigid body rotation rates was devised using the distance \hat{x}_3 as the interpolant:

$$\hat{\theta}_x^{\text{rig}} = \hat{\theta}_{x\text{-left}}^{\text{rig}} \left(1 - \frac{\hat{x}_3}{\hat{x}_2}\right) + \hat{\theta}_{x\text{-right}}^{\text{rig}} \left(\frac{\hat{x}_3}{\hat{x}_2}\right). \quad (11.15)$$

Substitution of [Equations \(11.13\) and \(11.14\)](#) into [\(11.15\)](#) and simplifying produces the relations given previously as [Equation \(11.12\)](#).

11.1.3 Stress Resultants and Nodal Forces

After suitable constitutive evaluation using the above velocity strains, the resulting local stresses are integrated through the thickness of the shell to obtain local resultant forces and moments. The integration formulae for the resultants are

$$\hat{f}_{\alpha\beta}^R = \int \hat{\sigma}_{\alpha\beta} d\hat{z}, \quad (11.16)$$

$$\hat{m}_{\alpha\beta}^R = - \int \hat{z} \hat{\sigma}_{\alpha\beta} d\hat{z}, \quad (11.17)$$

where the superscript R indicates a resultant force or moment and the Greek subscripts emphasize the limited range of the indices for plane stress plasticity.

The above element-centered force and moment resultant are related to the local nodal forces and moments by invoking the principle of virtual power and performing a one-point quadrature. The relations obtained in this manner are

$$\begin{Bmatrix} \hat{f}_{x1} \\ \hat{f}_{y1} \\ \hat{f}_{x2} \\ \hat{f}_{y2} \\ \hat{f}_{x3} \\ \hat{f}_{y3} \end{Bmatrix} = A \mathbf{B}_M^T \begin{Bmatrix} \hat{f}_{xx}^R \\ \hat{f}_{yy}^R \\ \hat{f}_{xy}^R \end{Bmatrix}, \quad (11.18)$$

$$\begin{Bmatrix} \hat{m}_{x1} \\ \hat{m}_{y1} \\ \hat{m}_{x2} \\ \hat{m}_{y2} \\ \hat{m}_{x3} \\ \hat{m}_{y3} \end{Bmatrix} = A \mathbf{B}_M^T \begin{Bmatrix} \hat{m}_{xx}^R \\ \hat{m}_{yy}^R \\ \hat{m}_{xy}^R \end{Bmatrix} + A \mathbf{B}_S^T \begin{Bmatrix} \hat{f}_{xz}^R \\ \hat{f}_{yz}^R \end{Bmatrix}, \quad (11.19)$$

where A is the area of the element ($2A = \hat{x}_2 \hat{y}_3$).

The remaining nodal forces, the \hat{z} component of the force $(\hat{f}_{z3}, \hat{f}_{z2}, \hat{f}_{z1})$, are determined by successively solving the following equilibration equations

$$\hat{m}_{x1} + \hat{m}_{x2} + \hat{m}_{x3} + \hat{y}_3 \hat{f}_{z3} = 0, \quad (11.20)$$

$$\hat{m}_{y1} + \hat{m}_{y2} + \hat{m}_{y3} - \hat{x}_3 \hat{f}_{z3} - \hat{x}_2 \hat{f}_{z2} = 0, \quad (11.21)$$

$$\hat{f}_{z1} + \hat{f}_{z2} + \hat{f}_{z3} = 0, \quad (11.22)$$

which represent moment equilibrium about the local x-axis, moment equilibrium about the local y-axis, and force equilibrium in the local z-direction, respectively.

11.2 Marchertas-Belytschko Triangular Shell

The Marchertas-Belytschko [1974] triangular shell element, or the BCIZ triangular shell element as it is referred to in the LS-DYNA user's manual, was developed in the same time period as the Belytschko beam element [Belytschko, Schwer, and Klein, 1977], see Section 4, forming the first generation of co-rotational structural elements developed by Belytschko and co-workers. This triangular shell element became the first triangular shell implemented in DYNA3D. Although the Marchertas-Belytschko shell element is relatively expensive, i.e., the C^0 triangular shell element with five through-the-thickness integration points requires 649 mathematical operations compared to 1,417 operations for the Marchertas-Belytschko triangular shell, it is maintained in LS-DYNA for compatibility with earlier user models. However, as the LS-DYNA user community moves to application of the more efficient shell element formulations, the use of the Marchertas-Belytschko triangular shell element will decrease.

As mentioned above, the Marchertas-Belytschko triangular shell has a common co-rotational formulation origin with the Belytschko beam element. The interested reader is referred to the beam element description, see Section 4, for details on the co-rotational formulation. In the next subsection a discussion of how the local element coordinate system is identical for the triangular shell and beam elements. The remaining subsections discuss the triangular element's displacement interpolants, the strain displacement relations, and calculations of the element nodal forces. In the report [1974], much greater detail is provided.

11.2.1 Element Coordinates

Figure 11.3(a) shows the element coordinate system, $(\hat{x}, \hat{y}, \hat{z})$ originating at Node 1, for the Marchertas-Belytschko triangular shell. The element coordinate system is associated with a triad of unit vectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ the components of which form a transformation matrix between the global and local coordinate systems for vector quantities. The nodal or body coordinate system unit vectors $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ are defined at each node and are used to define the rotational deformations in the element, see Section 8.4.4.

The unit normal to the shell element \mathbf{e}_3 is formed from the vector cross product

$$\mathbf{e}_3 = \mathbf{l}_{21} \times \mathbf{l}_{31}, \quad (11.23)$$

where \mathbf{l}_{21} and \mathbf{l}_{31} are unit vectors originating at Node 1 and pointing towards Nodes 2 and 3 respectively, see Figure 11.3(b).

Next a unit vector \mathbf{g} , see Figure 11.3(b), is assumed to be in the plane of the triangular element with its origin at Node 1 and forming an angle β with the element side between Nodes 1 and 2, i.e., the vector \mathbf{l}_{21} . The direction cosines of this unit vector are represented by the symbols (g_x, g_y, g_z) . Since \mathbf{g} is the unit vector, its direction cosines will satisfy the equation

$$g_x^2 + g_y^2 + g_z^2 = 1. \quad (11.24)$$

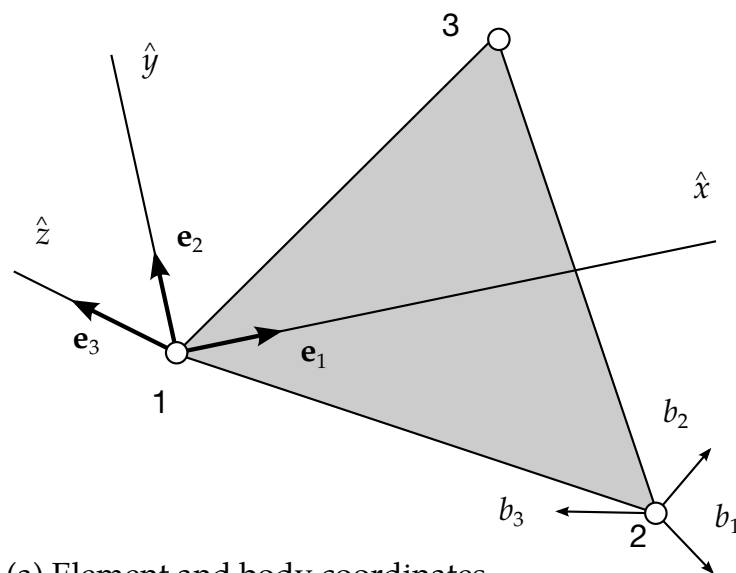
Also, since \mathbf{g} and \mathbf{e}_3 are orthogonal unit vectors, their vector dot product must satisfy the equation

$$e_{3x}g_x + e_{3y}g_y + e_{3z}g_z = 0. \quad (11.25)$$

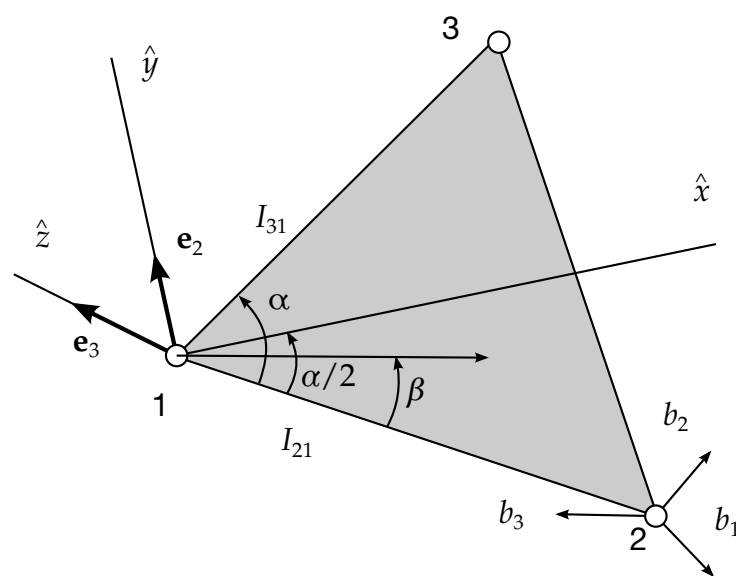
In addition, the vector dot product of the co-planar unit vectors \mathbf{g} and \mathbf{l}_{21} satisfies the equation

$$I_{21x}g_x + I_{21y}g_y + I_{21z}g_z = \cos\beta, \quad (11.26)$$

where $(I_{21x}, I_{21y}, I_{21z})$ are the direction cosines of \mathbf{l}_{21} .



(a) Element and body coordinates



(b) Construction of element coordinates

Figure 11.3. Construction of local element coordinate system.

Solving this system of three simultaneous equation, i.e., [Equation \(11.24\)](#), [\(11.25\)](#), and [\(11.26\)](#), for the direction cosines of the unit vector g yields

$$g_x = l_{21x} \cos \beta + (e_{3y} l_{21z} - e_{3z} l_{21y}) \sin \beta, \quad (11.27)$$

$$g_y = l_{21y} \cos \beta + (e_{3z} l_{21x} - e_{3x} l_{21z}) \sin \beta, \quad (11.28)$$

$$g_z = l_{21z} \cos \beta + (e_{3x} l_{21y} - e_{3y} l_{21x}) \sin \beta. \quad (11.29)$$

These equations provide the direction cosines for any vector in the plane of the triangular element that is oriented at an angle β from the element side between Nodes 1 and 2. Thus the unit vector components of \mathbf{e}_1 and \mathbf{e}_2 are obtained by setting $\beta = \alpha/2$ and $\beta = (\pi + \alpha)/2$ in Equation (8.22), respectively. The angle α is obtained from the vector dot product of the unit vectors \mathbf{l}_{21} and \mathbf{l}_{31} ,

$$\cos \alpha = \mathbf{l}_{21} \cdot \mathbf{l}_{31}. \quad (11.30)$$

11.2.2 Displacement Interpolation

As with the other large displacement and small deformation co-rotational element formulations, the nodal displacements are separated into rigid body and deformation displacements,

$$\mathbf{u} = \mathbf{u}^{\text{rigid}} + \mathbf{u}^{\text{def}}, \quad (11.31)$$

where the rigid body displacements are defined by the motion of the local element coordinate system, i.e., the co-rotational coordinates, and the deformation displacement are defined with respect to the co-rotational coordinates. The deformation displacement are defined by

$$\begin{Bmatrix} \hat{u}_x \\ \hat{u}_y \\ -\frac{\hat{u}_z}{\hat{\theta}} \end{Bmatrix}^{\text{def}} = \begin{bmatrix} \phi_x^m \\ \phi_y^m \\ -\frac{\phi_z^f}{\hat{\theta}} \end{bmatrix} \begin{Bmatrix} \delta \\ \hat{\theta} \end{Bmatrix}, \quad (11.32)$$

$$\delta^T = \{\delta_{12} \quad \delta_{23} \quad \delta_{31}\}, \quad (11.33)$$

are the edge elongations and

$$\hat{\theta} = \{\hat{\theta}_{1x} \quad \hat{\theta}_{1y} \quad \hat{\theta}_{2x} \quad \hat{\theta}_{2y} \quad \hat{\theta}_{3x} \quad \hat{\theta}_{3y}\}, \quad (11.34)$$

are the local nodal rotation with respect to the co-rotational coordinates.

The matrices ϕ_x^m , ϕ_y^m and ϕ_z^f are the membrane and flexural interpolation functions, respectively. The element's membrane deformation is defined in terms of the edge elongations. Marchertas and Belytschko adapted this idea from Argyris et al., [1964], where incremental displacements are used, by modifying the relations for total displacements,

$$\delta_{ij} = \frac{2(x_{ji}u_{jix} + y_{ji}u_{jiy} + z_{ji}u_{jiz}) + u_{jix}^2 + u_{jiy}^2 + u_{jiz}^2}{l_{ij}^0 + l_{ij}}, \quad (11.35)$$

where $x_{ji} = x_j - x_i$, etc.

The non-conforming shape functions used for interpolating the flexural deformations, ϕ_z^f were originally derived by Bazeley, Cheung, Irons, and Zienkiewicz [1965]; hence the LS-DYNA reference to the BCIZ element. Explicit expressions for ϕ_z^f are quite

tedious and are not given here. The interested reader is referred to Appendix G in the original work of Marchertas and Belytschko [1974].

The local nodal rotations, which are interpolated by these flexural shape functions, are defined in a manner similar to those used in the Belytschko beam element. The current components of the original element normal are obtained from the relation

$$\mathbf{e}_3^0 = \boldsymbol{\mu}^T \boldsymbol{\lambda} \bar{\mathbf{e}}_3^0, \quad (11.36)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are the current transformations between the global coordinate system and the element (local) and body coordinate system, respectively. The vector $\bar{\mathbf{e}}_3^0$ is the original element unit normal expressed in the body coordinate system. The vector cross product between this current-original unit normal and the current unit normal,

$$\mathbf{e}_3 \times \mathbf{e}_3^0 = \hat{\theta}_x \mathbf{e}_1 + \hat{\theta}_y \mathbf{e}_2, \quad (11.37)$$

define the local nodal rotations as

$$\hat{\theta}_x = -\hat{\mathbf{e}}_{3y}^0, \quad (11.38)$$

$$\hat{\theta}_y = \hat{\mathbf{e}}_{3x}^0. \quad (11.39)$$

Note that at each node the corresponding $\boldsymbol{\lambda}$ transformation matrix is used in [Equation \(11.36\)](#).

11.2.3 Strain-Displacement Relations

Marchertas-Belytschko impose the usual Kirchhoff assumptions that normals to the midplane of the element remain straight and normal, to obtain

$$e_{xx} = \frac{\partial u_x}{\partial x} - z \frac{\partial^2 u_z}{\partial x^2}, \quad (11.40)$$

$$e_{yy} = \frac{\partial u_y}{\partial y} - z \frac{\partial^2 u_z}{\partial y^2}, \quad (11.41)$$

$$2e_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} - 2z \frac{\partial^2 u_z}{\partial x \partial y}, \quad (11.42)$$

where it is understood that all quantities refer to the local element coordinate system.

Substitution of [Equations \(11.32\)](#) into the above strain-displacement relations yields

$$\boldsymbol{\varepsilon} = \mathbf{E}^m \boldsymbol{\delta} - z \mathbf{E}^f \hat{\boldsymbol{\theta}}, \quad (11.43)$$

where

$$\boldsymbol{\varepsilon}^T = \{\varepsilon_{xx} \quad \varepsilon_{yy} \quad 2\varepsilon_{xy}\}, \quad (11.44)$$

with

$$\mathbf{E}^m = \begin{bmatrix} \frac{\partial \phi_{xi}^m}{\partial x} \\ \frac{\partial \phi_{yi}^m}{\partial y} \\ \frac{\partial \phi_{xi}^m}{\partial y} + \frac{\partial \phi_{yi}^m}{\partial x} \end{bmatrix}, \quad (11.45)$$

and

$$\mathbf{E}^f = \begin{bmatrix} \frac{\partial^2 \phi_{zi}^f}{\partial x^2} \\ \frac{\partial^2 \phi_{zi}^f}{\partial y^2} \\ 2 \frac{\partial^2 \phi_{zi}^f}{\partial x \partial y} \end{bmatrix}. \quad (11.46)$$

Again, the interested reader is referred to Appendices F and G in the original work of Marchertas and Belytschko [1974] for explicit expressions of the above two matrices.

11.2.4 Nodal Force Calculations

The local element forces and moments are found by integrating the local element stresses through the thickness of the element. The local nodal forces are given by

$$\hat{\mathbf{f}} = \int \mathbf{E}^{mT} \hat{\boldsymbol{\sigma}} dV, \quad (11.47)$$

$$\hat{\mathbf{f}}^T = \{f_{12}, f_{23}, f_{31}\}, \quad (11.48)$$

$$\hat{\boldsymbol{\sigma}}^T = \{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}, \quad (11.49)$$

where the side forces and stresses are understood to all be in the local convected coordinate system.

Similarly, the local moments are given by

$$\hat{\mathbf{m}} = - \int z \mathbf{E}^{fT} \hat{\boldsymbol{\sigma}} dV, \quad (11.50)$$

$$\hat{\mathbf{m}}^T = \{\hat{m}_{1x} \quad \hat{m}_{1y} \quad \hat{m}_{2x} \quad \hat{m}_{2y} \quad \hat{m}_{3x} \quad \hat{m}_{3y}\}. \quad (11.51)$$

The through-the-thickness integration portions of the above local force and moment integrals are usually performed with a 3- or 5-point trapezoidal integration. A three-point inplane integration is also used; it is inpart this three-point inplane integration that increases the operation count for this element over the C^0 shell, which used one-point inplane integration with hourglass stabilization.

The remaining transverse nodal forces are obtained from element equilibrium considerations. Moment equilibrium requires

$$\begin{Bmatrix} \hat{f}_{2z} \\ \hat{f}_{3z} \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} -\hat{x}_3 & \hat{y}_3 \\ \hat{x}_2 & -\hat{y}_2 \end{bmatrix} \begin{Bmatrix} \hat{m}_{1x} + \hat{m}_{2x} + \hat{m}_{3x} \\ \hat{m}_{1y} + \hat{m}_{2y} + \hat{m}_{3y} \end{Bmatrix}, \quad (11.52)$$

where A is the area of the element. Next transverse force equilibrium provides

$$\hat{f}_{1z} = -\hat{f}_{2z} - \hat{f}_{3z}. \quad (11.53)$$

The corresponding global components of the nodal forces are obtained from the following transformation

$$\begin{Bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{Bmatrix} = \frac{f_{ij}}{l_{ij}} \begin{Bmatrix} x_{ij} + u_{ijx} \\ y_{ij} + u_{ijy} \\ z_{ij} + u_{ijz} \end{Bmatrix} + \frac{f_{ik}}{l_{ik}} \begin{Bmatrix} x_{ik} + u_{ikx} \\ y_{ik} + u_{iky} \\ z_{ik} + u_{ikz} \end{Bmatrix} + \hat{f}_{iz} \begin{Bmatrix} e_{3x} \\ e_{3y} \\ e_{3z} \end{Bmatrix}. \quad (11.54)$$

Finally, the local moments are transformed to the body coordinates using the relation

$$\begin{Bmatrix} \bar{m}_{ix} \\ \bar{m}_{iy} \\ \bar{m}_{iz} \end{Bmatrix} = \lambda^T \boldsymbol{\mu} \begin{Bmatrix} \hat{m}_{ix} \\ \hat{m}_{iy} \\ \hat{m}_{iz} \end{Bmatrix}. \quad (11.55)$$

12

Fully Integrated Shell (Type 16)

12.1 Introduction

Shell type 16 in LS-DYNA is a fully integrated shell with assumed strain interpolants used to alleviate locking and enhance in-plane bending behavior, see Engelmann, Whirley, and Goudreau [1989]; Simo and Hughes [1986]; Pian and Sumihara [1985]. It uses a local element coordinate system that rotates with the material to account for rigid body motion and automatically satisfies frame invariance of the constitutive relations. The local element coordinate system is similar to the one used for the Belytschko-Tsay element, where the the first two basis vectors are tangent to the shell midsurface at the center of the element, and the third basis vector is in the normal direction of this surface and initially coincident with the fiber vectors.

12.2 Hu-Washizu Three Field Principle

The element is derived starting from the Hu-Washizu three-field principle stated as

$$0 = \delta \Pi(\mathbf{v}, \bar{\mathbf{D}}, \bar{\boldsymbol{\sigma}}) = \int_{\Omega} \delta \bar{\mathbf{D}} : \boldsymbol{\sigma}(\bar{\mathbf{D}}) d\Omega + \int_{\Omega} \delta [\bar{\boldsymbol{\sigma}} : (\mathbf{D}(\mathbf{v}) - \bar{\mathbf{D}})] d\Omega - \delta P_{\text{ext}} + \delta P_{\text{kin}}, \quad (12.1)$$

where \mathbf{v} is the velocity, $\bar{\mathbf{D}}$ is the assumed strain rate, $\bar{\boldsymbol{\sigma}}$ is the assumed stress, $\boldsymbol{\sigma}$ denotes the constitutive update as a function of the assumed strain rate, and \mathbf{D} is the strain rate computed from the velocity field. δP_{kin} and δP_{ext} are the virtual power contributions from the inertial and external forces, respectively, and Ω denotes the domain of the shell element. The contribution from the internal forces can be decomposed in the in-plane and transverse shear parts as

$$0 = \delta P_{\text{int}}^p + \delta P_{\text{int}}^s + H^p + H^s - \delta P_{\text{ext}} + \delta P_{\text{kin}}, \quad (12.2)$$

where

$$\delta P_{\text{int}}^{\text{p}} = \int_{\Omega} \delta \bar{\mathbf{D}}^{\text{p}} : \boldsymbol{\sigma}^{\text{p}}(\bar{\mathbf{D}}) d\Omega, \quad (12.3)$$

$$\delta P_{\text{int}}^{\text{s}} = \kappa \int_{\Omega} \delta \bar{\mathbf{D}}^{\text{s}} : \boldsymbol{\sigma}^{\text{s}}(\bar{\mathbf{D}}) d\Omega, \quad (12.4)$$

$$H^{\text{p}} = \int_{\Omega} \delta [\bar{\boldsymbol{\sigma}}^{\text{p}} : (\mathbf{D}^{\text{p}}(\mathbf{v}) - \bar{\mathbf{D}}^{\text{p}})] d\Omega, \quad (12.5)$$

$$H^{\text{s}} = \kappa \int_{\Omega} \delta [\bar{\boldsymbol{\sigma}}^{\text{s}} : (\mathbf{D}^{\text{s}}(\mathbf{v}) - \bar{\mathbf{D}}^{\text{s}})] d\Omega. \quad (12.6)$$

Here κ is the shear correction factor and the superscripts mean that only the in-plane components (p) or transverse shear (s) components are treated.

12.3 In-plane Assumed Strain Field

Using the standard isoparametric interpolation for the four-node quadrilateral element, the in-plane strain rate can be written

$$\mathbf{D}^{\text{p}} = [\mathbf{B}_{\text{m}} \quad z\mathbf{B}_{\text{b}}] \begin{bmatrix} \mathbf{v}^{\text{p}} \\ \dot{\boldsymbol{\theta}}^{\text{p}} \end{bmatrix}, \quad (12.7)$$

where \mathbf{B}_{m} and \mathbf{B}_{b} are strain-displacement matrices for membrane and bending modes, respectively, z is the through thickness coordinate and \mathbf{v}^{p} and $\dot{\boldsymbol{\theta}}^{\text{p}}$ are the nodal (in-plane) translational and rotational velocities, respectively.

To derive the in-plane assumed strain field, the interpolants for the assumed stress and strain rates are chosen as

$$\bar{\boldsymbol{\sigma}}^{\text{p}} = [\mathbf{S}^{\text{p}} \quad \mathbf{S}^{\text{p}}] \begin{bmatrix} \mathbf{s}_{\text{m}} \\ \mathbf{s}_{\text{b}} \end{bmatrix}, \quad (12.8)$$

$$\bar{\mathbf{D}}^{\text{p}} = \mathbf{C}^{-1} [\mathbf{S}^{\text{p}} \quad \mathbf{S}^{\text{p}}] \begin{bmatrix} \mathbf{e}_{\text{m}} \\ \mathbf{e}_{\text{b}} \end{bmatrix}, \quad (12.9)$$

where

$$\mathbf{S}^{\text{p}} = \begin{bmatrix} 1 & 0 & 0 & a_1^2 \hat{\eta} & b_1^2 \hat{\xi} \\ 0 & 1 & 0 & a_2^2 \hat{\eta} & b_2^2 \hat{\xi} \\ 0 & 0 & 1 & a_1 a_2 \hat{\eta} & b_1 b_2 \hat{\xi} \end{bmatrix}, \quad (12.10)$$

and

$$\hat{\xi} = \xi - \frac{1}{|\Omega|} \int_{\Omega} \xi d\Omega, \quad (12.11)$$

$$\hat{\eta} = \eta - \frac{1}{|\Omega|} \int_{\Omega} \eta d\Omega. \quad (12.12)$$

Furthermore, \mathbf{C} is the plane stress constitutive matrix and ξ and η are the isoparametric coordinates. The coefficients a_i and b_i are defined through

$$\mathbf{J}_0 = \frac{1}{4} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}, \quad (12.13)$$

where \mathbf{J}_0 is the area jacobian matrix from the isoparametric to physical domain computed at the element center.

Inserting the expressions for the strain rate and assumed stress and strain rate into the expression for H^P and requiring $H^P = 0$ for arbitrary \mathbf{s}_m , \mathbf{s}_b , \mathbf{e}_m and \mathbf{e}_b , yields the following expression for the assumed strain rate in terms of the nodal velocities

$$\bar{\mathbf{D}}^P = [\bar{\mathbf{B}}_m \quad z\bar{\mathbf{B}}_b] \begin{bmatrix} \mathbf{v}^P \\ \dot{\theta}^P \end{bmatrix}, \quad (12.14)$$

where

$$\bar{\mathbf{B}}_m = \mathbf{C}^{-1} \mathbf{S}^P \hat{\mathbf{E}} \hat{\mathbf{B}}_m, \quad (12.15)$$

$$\bar{\mathbf{B}}_b = \mathbf{C}^{-1} \mathbf{S}^P \hat{\mathbf{E}} \hat{\mathbf{B}}_b, \quad (12.16)$$

and

$$\hat{\mathbf{E}} = \int_{\Omega} \mathbf{S}^{PT} \mathbf{C}^{-1} \mathbf{S}^P d\Omega, \quad (12.17)$$

$$\hat{\mathbf{B}}_m = \int_{\Omega} \mathbf{S}^{PT} \mathbf{B}_m d\Omega, \quad (12.18)$$

$$\hat{\mathbf{B}}_b = \int_{\Omega} \mathbf{S}^{PT} \mathbf{B}_b d\Omega. \quad (12.19)$$

12.4 Transverse Shear Assumed Strain Field

The transverse shear strain is the Bathe-Dvorkin [1984] assumed natural strain field and is derived as follows. Using the standard isoparametric interpolation for the four-node quadrilateral element, the transverse shear strain rate can be written

$$\mathbf{D}^S = \mathbf{B}_t \begin{bmatrix} \mathbf{v}_z \\ \dot{\theta}^P \end{bmatrix}, \quad (12.20)$$

where \mathbf{B}_t is the corresponding strain-displacement matrix and \mathbf{v}_z and $\dot{\theta}^P$ are the nodal out-of-plane translational and in-plane rotational velocities, respectively.

The assumed strain rate is defined as

$$\bar{\mathbf{D}}^s = \bar{\mathbf{B}}_t \begin{bmatrix} \mathbf{v}_z \\ \dot{\boldsymbol{\theta}}^p \end{bmatrix}, \quad (12.21)$$

where

$$\bar{\mathbf{B}}_t = \mathbf{J}^{-T} \mathbf{E} \int_{\Omega} \mathbf{S}^s \mathbf{J}^T \mathbf{B}_t d\Omega, \quad (12.22)$$

Here \mathbf{J} is the area jacobian matrix from the isoparametric domain to the physical domain,

$$\mathbf{E} = \frac{1}{2} \begin{bmatrix} 1 - \zeta & 1 + \zeta & 0 & 0 \\ 0 & 0 & 1 - \eta & 1 + \eta \end{bmatrix}, \quad (12.23)$$

$$\mathbf{S}^s = \begin{bmatrix} \delta(\eta)\delta(1 + \zeta) & \delta(\eta)\delta(1 - \zeta) & 0 & 0 \\ 0 & 0 & \delta(\zeta)\delta(1 + \eta) & \delta(\zeta)\delta(1 - \eta) \end{bmatrix}, \quad (12.24)$$

and δ is the Dirac delta function. Defining the assumed stress as

$$\bar{\boldsymbol{\sigma}}^p = \mathbf{J} \mathbf{S}^s \mathbf{s}, \quad (12.25)$$

yields $H^s = 0$ regardless of the choice of \mathbf{s} and thus a B-bar expression for the assumed transverse strain rates is obtained as given above. The result is equivalent to defining the isoparametric assumed shear strain rates by interpolating the corresponding strain rates from the mid-side points A, B, C and D shown in [Figure 12.1](#).

12.5 Rigid Body Motion

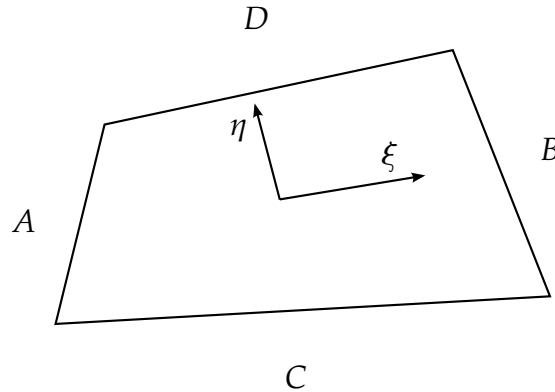


Figure 12.1. Midside locations of isoparametric strain rates

For the in-plane assumed strain field, a rigid body motion may induce a nonzero strain rate. The expression for the in-plane strain rate for a rigid body motion is

$$\bar{\mathbf{D}}^r = \bar{\mathbf{B}}_r \dot{\boldsymbol{\theta}}, \quad (12.26)$$

where

$$\bar{\mathbf{B}}_r = w \bar{\mathbf{B}}_m \mathbf{R}. \quad (12.27)$$

12.6 Belytschko-Leviathan Projection

For warped configurations and since the geometry of the current shell element is flat, extremely flexible behavior can be expected for some modes of deformation. Following [Belytschko and Leviathan 1994], a 7-mode projection matrix \mathbf{P} (3 rigid body rotation modes and 4 nodal drill rotation modes) is constructed used for projecting out these zero energy modes. The explicit formula for the projection matrix is given by

$$\mathbf{P} = \mathbf{I} - \mathbf{R}(\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T, \quad (12.28)$$

where \mathbf{R} is a matrix where each column corresponds to the nodal velocity of a zero energy mode. This projection matrix operates on the nodal velocities prior to computing the strain rates, and also on the resulting internal force vector to maintain invariance of the internal power.

13

Shells with Thickness Stretch

13.1 Introduction

Thickness stretch is of considerable importance in problems involving finite thickness strains, contact and surface loads in nonlinear shell applications. As an example, in sheet metal forming applications, the presence of normal stresses in thickness direction improves the accuracy of the solution and also its response on the double sided contact zone between dies and sheet. It has also been shown that a kinematical representation of a continuous thickness field improves the instability characteristics when compared to experimental results, see [Figure Figure 1313-1](#) and Björklund [2014].

There have been several attempts to account for the through-thickness deformation in the literature, this implementation is inspired by the 7P-CYSE shell introduced in Cardoso and Yoon [2005]. However, the formulation in this paper is rather complicated and involves for instance assumed shear strains (ANS) to alleviate shear locking and complicated setup of internal forces and stiffness matrices by combined analytical and numerical integration to restore rank deficiencies. A direct implementation of this shell following the theory was ruled out for efficiency reasons, and another approach was taken in order to presumably get a useful shell.

The Belytschko-Tsay shell element is one of the fastest elements for thin shell simulations. This, together with its robustness, is the reason why it is popular in finite element codes. The implementation of shell type 25, the reduced integrated shell with thickness stretch, is based on the formulation of the Belytschko-Tsay shell with a relaxation of the thickness variable. This ensures that it will be efficient and hopefully also possess properties useful for applications where through thickness deformation is important. As a fully integrated alternative, shell type 26 is available as an analogue extension of shell type 16 (fully integrated Belytschko-Tsay), and a triangular shell with thickness stretch is available as type 27 mainly to allow for hybrid meshes (quadrilaterals combined with triangles) in this context.

The theory that follows is very similar to that of the Belytschko-Tsay (type 2), Fully Integrated Shell (type 16) and C_0 -shell (type 4), and here we emphasize on the parts involving the amendments to those shell formulations.

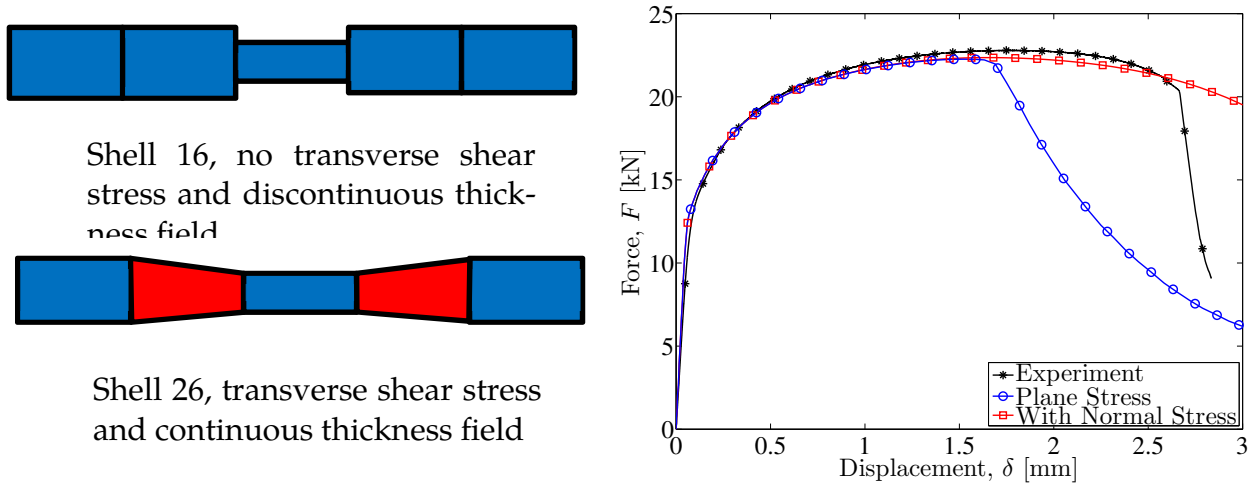


Figure 1313-1 Example of premature localization with shell type 16 while shell type 26 matches experimental result (from Björklund)

13.2 Shell Type 25

13.2.1 Formulation

The kinematics, i.e., position \mathbf{x} and velocity \mathbf{v} , of a material point in the shell with through thickness stretch can be written in the shell element local coordinate system as (sum over nodal indices I)

$$\mathbf{x} = (\mathbf{x}_I + s_I \mathbf{n}) N_I(\xi, \eta) \quad (13.1)$$

$$\mathbf{v} = (\mathbf{v}_I + s_I \boldsymbol{\omega}_I \times \mathbf{n} + \dot{s}_I \mathbf{n}) N_I(\xi, \eta)$$

where we have set

$$s_I = \frac{\zeta}{2} t_I + (1 - \zeta^2) q_I. \quad (13.2)$$

The kinematics is based on the Belytschko-Tsay shell with the additional feature that the thickness is variable, whence the last term in the second of (13.1). The thickness variable is represented by t_I and an additional strain variable q_I to allow for a linear strain through the thickness. The latter represents the location of the mid-surface and is important to avoid “Poisson locking” in bending modes of deformation, so the shell has two additional degrees of freedom compared to the Belytschko-Tsay shell. The other variables and parameters are the nodal coordinates \mathbf{x}_I , nodal velocities \mathbf{v}_I , nodal rotational velocities $\boldsymbol{\omega}_I$, shell normal $\mathbf{n} = (0 \ 0 \ 1)^T$ and bilinear isoparametric shape functions N_I . From this we can determine the local velocity gradient as

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}} = (\mathbf{v}_I + s_I \boldsymbol{\omega}_I \times \mathbf{n} + \dot{s}_I \mathbf{n}) \frac{\partial N_I}{\partial \mathbf{x}} + \left(\boldsymbol{\omega}_I \times \mathbf{n} \frac{\partial s_I}{\partial \mathbf{x}} + \mathbf{n} \frac{\partial \dot{s}_I}{\partial \mathbf{x}} \right) N_I. \quad (13.3)$$

In the local system one can assume a vanishing third component of both $\omega_I \times \mathbf{n}$ and $\frac{\partial N_I}{\partial x}$, and the thickness strain rate is given by

$$\dot{\epsilon}_t = \frac{\partial \dot{s}}{\partial x_3} = \frac{\dot{t} - 4\zeta \dot{q}}{t - 4\zeta q} \quad (13.4)$$

where we used the notation $s = s_I N_I$, $t = t_I N_I$ and $q = q_I N_I$, sum over I . For small strains $q = 0$, and this shows that the thickness strain rate is at most linear.

For evaluating internal forces we define the strain-displacement tensor through (assuming Voigt notation and sum over I)

$$\frac{\partial v}{\partial x} = B_I u_I, \quad (13.5)$$

and the nodal vector u_I is given by

$$u_I = (v_I^T \quad \omega_I^T \quad \dot{t}_I \quad \dot{q}_I)^T, \quad (13.6)$$

indicating the 8 degrees of freedom per node in these elements. The principle of virtual work results in an internal force vector

$$f_I = \int B_I^T \sigma, \quad (13.7)$$

where σ is the Cauchy stress and the integral is over the current element configuration.

13.2.2 Hourglass modes

Shell type 25 is a reduced integration element and in addition to the six hourglass modes present in the original Belytschko-Tsay shell, two more are added by the introduction of the thickness variables. Fortunately these are orthogonal to any rigid body motion and/or any other hourglass mode, and are given by

$$\begin{aligned} t_I &= h_I \\ q_I &= h_I \end{aligned} \quad (13.8)$$

where

$$h_I = (-1)^I / 4 \quad (13.9)$$

and all other displacement components are zero. To restrain these modes we have included generalized strains and stresses according to the following (sum over I)

$$\begin{aligned} \dot{\epsilon}_t^h &= \frac{h_I \dot{t}_I}{t} \\ \dot{\epsilon}_q^h &= -\frac{4h_I \dot{q}_I}{t} \end{aligned} \quad (13.10)$$

The corresponding generalized stresses are obtained through

$$\dot{\sigma}_t = E_H \dot{\epsilon}_t^h \quad (13.11)$$

$$\dot{\sigma}_q = \frac{E_H}{3} \dot{\epsilon}_q^h$$

and the forces are then given by

$$\begin{aligned} f_I^t &= Ah_I \sigma_t \\ f_I^q &= -4Ah_I \sigma_q \end{aligned} \tag{13.12}$$

where A is the area of the element and the value of E_H is taken as $E_H = 0.05E$, i.e., 5% of the Young's modulus.

13.3 Shell Type 26

13.3.1 Modifying shell type 16

When using full integration, care must be taken in order to avoid the well-known shear locking phenomenon. Common techniques developed for this are the ANS (Assumed Natural Strain) and various types of EAS (Enhanced Assumed Strain) techniques. In Cardoso and Yoon [2004] the ANS technique is used in which the transverse shear strains are interpolated from the mid points of the shell element edges. It is reported that this is a successful technique, but when consulting Bischoff and Ramm [1997] it is reported that it is not well suited to avoid membrane locking and reduce mesh distortion sensitivity. They are proposing a combination of ANS and EAS to get a decent shell element formulation. However, elements that use an EAS in general require a nonlinear solution for the assumed strain variables which can be computationally rather expensive. The approach taken here is to suitably modifying an existing LS-DYNA standard assumed strain element, i.e., element type 16, which is a fully integrated extension of the Belytschko-Tsay element. Due to the rather complex setup of assumed strain components, we use the following in order to extend the element to variable thickness stretch without having to bother about all the details of the kinematics.

In the beginning of a time step, the thickness components are reset to the corresponding value in the center of the shell element. That is

$$\tilde{s} = \frac{s_1 + s_2 + s_3 + s_4}{4} \tag{13.13}$$

is used for evaluating the strain increments. However, the rates of the thickness variables are unchanged, so this can be seen as an assumed strain approach in the thickness direction. Using this, the velocity gradient expression is an augmentation of that of element 16

$$\frac{\partial v}{\partial x} = \frac{\partial v}{\partial x} \Big|_{16} + \dot{s}_I \mathbf{n} \frac{\partial N_I}{\partial x} + \mathbf{n} \frac{\partial \dot{s}_I}{\partial x} N_I. \tag{13.14}$$

This should be interpreted that the velocity gradient used in shell element type 16 as a function of the current thickness coordinate \tilde{s} (and its isoparametric derivative) is augmented by thickness component rates. Furthermore, to avoid locking phenomena and

still maintaining a non-singular element we use single point integration of the thickness strain component and an ANS approach for the transverse shear strain components emanating from the rates of thickness components. This approach is inspired by the methodology used for the Belytschko-Tsay shell in which the nodal fiber vectors are reset in the beginning of each time step, and in an analogous way we reset the thickness components in the beginning of each time step in this shell element formulation.

13.4 Shell Type 27

13.4.1 Modifying shell type 4

A triangular element with thickness stretch was added to allow consistent sorting of triangular shell elements, and not necessarily intended to be used for other purposes. The approach taken is very similar to that of shell type 26, the existing LS-DYNA standard C_0 triangular element was modified according to the same principles as described in the previous section. In this case the velocity gradient expression is modified according to

$$\frac{\partial v}{\partial x} = \frac{\partial v}{\partial x}\bigg|_4 + n \frac{\partial \tilde{s}}{\partial x} \quad (13.15)$$

where again the subindex 4 indicates that it's the velocity gradient of element type 4 evaluated with respect to the average thickness value \tilde{s} and its isoparametric derivative. For the triangular element type 27 the thickness is always constant in the element, a decision that was taken for the sake of simplicity. More on thickness distribution in the next section.

13.5 Related Features

13.5.1 Continuous vs decoupled thickness field

A drawback with having a continuous thickness field is that complicated geometries tend to lock the structure. This approach assumes that the geometry is relatively flat and may be suitable in metal forming but not in other situations. To remedy this we added the option to decouple the thickness field so that the thickness is discontinuous between elements, which makes the element suitable for crash analysis. To activate this option the user should put the variable IDOF on *SECTION_SHELL equal to 2, which actually is the default. For shell element type 25, this option will make the thickness variable constant in the element, and the implementation follows the one described above with the restriction $t_I = t$ (constant) and $q_I = q$ (constant). No additional zero energy modes are present with this approach.

Using the same approach for element type 26 turns out to be a poor choice since it leads to locking phenomenon due to full integration. Instead we use the approach where the thickness is bilinear within but discontinuous between elements, which turns out to be

successful. That means that the same code can be executed regardless of the choice of IDOF for this element.

13.5.2 Nodal masses

For dynamic analysis some mass quantity must be associated with the extra degrees of freedom, and although a consistent finite element approach is applicable the mass is here empirically estimated from elaborating with the kinetic energy for compressive in-plane and out-of-plane modes. That is, the kinetic energy of an in-plane uniaxial strain mode should have the same kinetic energy as the equivalent out-of-plane ditto. This assumption leads to a mass of the scalar nodes given by $m = m_T/4$ where m_T is the translational mass. A problem with this approach is that it leads to instabilities due to high eigenfrequencies of the shell. For this reason we have set

$$m = \max(1, \frac{3A}{2t^2})m_T \quad (13.16)$$

where A is the element area and t is the thickness.

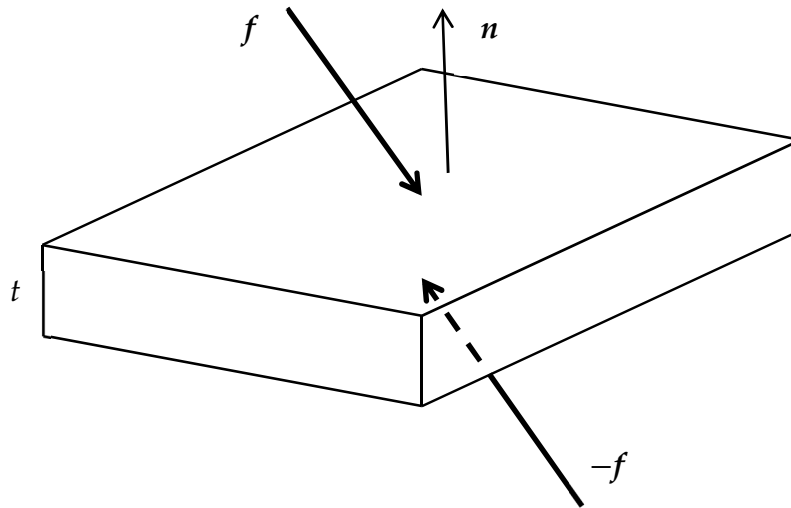


Figure 1313-2 Contact influence on thickness stretch shells.

13.5.3 Transfer of contact forces

The new degrees of freedom allows for a different treatment of how contact forces are transferred from master to slave, for this discussion we refer to [Figure Figure 1313-2](#). The nodal forces in local system acting on the slave from contact on the upper and lower surface can roughly be written

$$f_I^{u/l} = \left(\pm \frac{f^T}{4} \quad \frac{t}{2} \left(n \times \frac{f}{4} \right)^T \quad \frac{n \bullet f}{8} \quad 0 \right)^T \quad (13.17)$$

where we used the same order of the degrees of freedom as in (13.6). Summing these two force contributions give nodal contact forces acting on the slave given by

$$f_I^c = \left(\mathbf{0}^T \quad t \left(\mathbf{n} \times \frac{\mathbf{f}}{4} \right)^T \quad \frac{\mathbf{n} \bullet \mathbf{f}}{4} \quad 0 \right)^T \quad (13.18)$$

We can see that the total force include nodal moments caused by the frictional forces acting at an offset from the midsurface of the shell, but also a pressure acting on the thickness degree of freedom. In conclusion, the relaxation of the thickness in the Belytschko-Tsay shell allows for double sided contact zones, i.e., the shell is affected by contact pressure from both sides even though they are of equal magnitude. This is not possible in traditional shell with zero normal stress, unless a modified option is used.

13.6 Contact Pressure Treatment in Shells 2, 4 and 16

By specifying IDOF = 3 on *SECTION_SHELL for the Belytschko-Tsay (type 2), C₀-element (type 4) and Fully Integrated (type 16) shell, the contact pressure influences the stress and can induce thickness changes. This is a short explanation of the theory behind. The z-stress in a shell element is usually restricted to be zero, but in this case we intend to solve the constitutive update using the constraint

$$\sigma_{zz} = \alpha \sigma_c(z) \quad (13.19)$$

where

$$\sigma_c(z) = -\frac{\sigma_c^b - \sigma_c^t}{4}(z^3 - 3z) - \frac{\sigma_c^b + \sigma_c^t}{2} \quad (13.20)$$

and

σ_c^b = contact pressure at bottom surface of shell

σ_c^t = contact pressure at top surface of shell

z = isoparametric coordinate through the thickness between -1 and 1

α = scaling parameter

The scaling parameter can be set as the 8th parameter on card 3 on *CONTROL_CONTACT.

The constitutive update for an elastic-plastic material can be written

$$\sigma^{n+1} = \sigma^n + K \Delta \varepsilon_{vol} \mathbf{I} + \Delta \mathbf{s}(\mathbf{s}_n, \Delta \varepsilon_{dev}) \quad (13.21)$$

$$\sigma_{zz}^{n+1} = \sigma_c^{n+1} \quad (13.22)$$

where

σ^{n+1} = stress in step $n+1$

σ^n = stress in step n

K = bulk modulus

$\Delta \varepsilon$ = strain increment

$\Delta \varepsilon_{vol} = \Delta \varepsilon$: \mathbf{I} = volumetric strain increment

\mathbf{I} = unit tensor

$\Delta \mathbf{s}$ = deviatoric stress increment

$\mathbf{s}_n = \boldsymbol{\sigma}^n - \frac{\sigma_c^n}{3} \mathbf{I}$ = deviatoric stress in step n

$\Delta \boldsymbol{\varepsilon}_{dev} = \Delta \boldsymbol{\varepsilon} - \frac{\Delta \varepsilon_{zz}}{3} \mathbf{I}$ = deviatoric strain increment

The independent variables in (13.21) and (13.22) are $\sigma_{xx}^{n+1}, \sigma_{yy}^{n+1}, \sigma_{xy}^{n+1}, \sigma_{yz}^{n+1}, \sigma_{xz}^{n+1}$ and $\Delta \varepsilon_{zz}$. Here we assume that the stress response can be decoupled into a volumetric and deviatoric part and the deviatoric stress increment depends only on the deviatoric part of the stress and strain increment as indicated in the formula. We can rewrite (13.21) and (13.22) as

$$\tilde{\boldsymbol{\sigma}}^{n+1} = \tilde{\boldsymbol{\sigma}}^n + K \Delta \tilde{\varepsilon}_{vol} \mathbf{I} + \Delta \mathbf{s}(\tilde{\mathbf{s}}_n, \Delta \tilde{\boldsymbol{\varepsilon}}_{dev}) \quad (13.23)$$

$$\tilde{\sigma}_{zz}^{n+1} = 0 \quad (13.24)$$

by substituting

$$\tilde{\boldsymbol{\sigma}}^n = \boldsymbol{\sigma}^n - \sigma_c^n \mathbf{I} \quad (13.25)$$

$$\tilde{\boldsymbol{\sigma}}^{n+1} = \boldsymbol{\sigma}^{n+1} - \sigma_c^{n+1} \mathbf{I} \quad (13.26)$$

and

$$\Delta \tilde{\boldsymbol{\varepsilon}} = \Delta \boldsymbol{\varepsilon} - \frac{\sigma_c^{n+1} - \sigma_c^n}{3K} \mathbf{I}. \quad (13.27)$$

Since the deviatoric stress and strain increment is not changed, i.e.,

$$\mathbf{s}_n = \tilde{\mathbf{s}}_n \quad (13.28)$$

$$\Delta \boldsymbol{\varepsilon}_{dev} = \Delta \tilde{\boldsymbol{\varepsilon}}_{dev} \quad (13.29)$$

with this substitution it follows that the existing material routines can be used for solving (13.23) and (13.24) in terms of $\tilde{\sigma}_{xx}^{n+1}, \tilde{\sigma}_{yy}^{n+1}, \tilde{\sigma}_{xy}^{n+1}, \tilde{\sigma}_{yz}^{n+1}, \tilde{\sigma}_{xz}^{n+1}$ and $\Delta \tilde{\varepsilon}_{zz}$ and then use the inverse of (13.26) and (13.27) to establish the stress and through thickness strain increment. Thus, the algorithm is as follows

1. Given $\boldsymbol{\sigma}^n, \Delta \boldsymbol{\varepsilon}, \sigma_c^n$ and σ_c^{n+1}
2. Use (13.25) and (13.27) to compute $\tilde{\boldsymbol{\sigma}}^n$ and $\Delta \tilde{\boldsymbol{\varepsilon}}$.
3. Do a constitutive update, (13.23) and (13.24), to get $\tilde{\boldsymbol{\sigma}}^{n+1}$ and $\Delta \tilde{\varepsilon}_{zz}$.
4. Use (13.26) and (13.27) to compute $\boldsymbol{\sigma}^{n+1}$ and $\Delta \varepsilon_{zz}$.

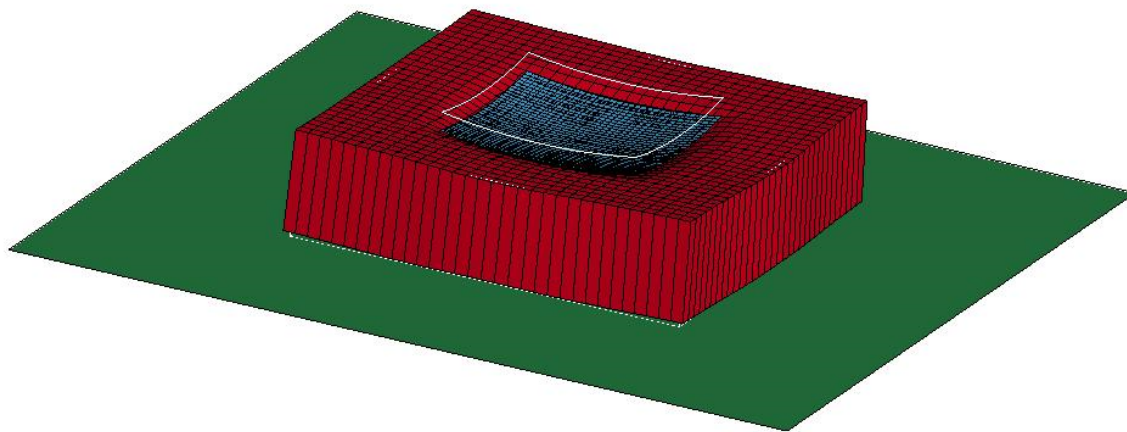


Figure 1313-3 Compression of a shell sheet between two rigid plates using IDOF = 3 on shell 16.

14

Hughes-Liu Shell

The Hughes-Liu shell element formulation ([Hughes and Liu 1981a, b], [Hughes et al., 1981], [Hallquist et al., 1985]) was the first shell element implemented in LS-DYNA. It was selected from among a substantial body of shell element literature because the element formulation has several desirable qualities:

- it is incrementally objective (rigid body rotations do not generate strains), allowing for the treatment of finite strains that occur in many practical applications;
- it is simple, which usually translates into computational efficiency and robustness;
- it is compatible with brick elements, because the element is based on a degenerated brick element formulation. This compatibility allows many of the efficient and effective techniques developed for the DYNA3D brick elements to be used with this shell element;
- it includes finite transverse shear strains;
- a through-the-thickness thinning option (see [Hughes and Carnoy 1981]) is also available when needed in some shell element applications.

The remainder of this section reviews the Hughes-Liu shell element (referred to by Hughes and Liu as the U1 element) which is a four-node shell with uniformly reduced integration, and summarizes the modifications to their theory as it is implemented in LS-DYNA. A detailed discussion of these modifications, as well as those associated with the implementation of the Hughes-Liu shell element in NIKE3D, are presented in an article by Hallquist and Benson [1986].

14.1 Geometry

The Hughes-Liu shell element is based on a degeneration of the standard 8-node brick element formulation, an approach originated by Ahmad et al. [1970]. Recall from the discussion of the solid elements the isoparametric mapping of the biunit cube:

$$\mathbf{x}(\xi, \eta, \zeta) = N_a(\xi, \eta, \zeta) \mathbf{x}_a, \quad (14.1)$$

$$N_a(\xi, \eta, \zeta) = \frac{(1 + \xi_a \xi)(1 + \eta_a \eta)(1 + \zeta_a \zeta)}{8}, \quad (14.2)$$

where \mathbf{x} is an arbitrary point in the element, (ξ, η, ζ) are the parametric coordinates, \mathbf{x}_a are the global nodal coordinates of node a , and N_a are the element shape functions evaluated at node a , i.e., (ξ_a, η_a, ζ_a) are (ξ, η, ζ) evaluated at node a .

In the shell geometry, planes of constant ζ will define the lamina or layers of the shell and fibers are defined by through-the-thickness lines when both ξ and η are constant (usually only defined at the nodes and thus referred to as 'nodal fibers'). To degenerate the 8-node brick geometry into the 4-node shell geometry, the nodal pairs in the ζ direction (through the shell thickness) are combined into a single node, for the translation degrees of freedom, and an inextensible nodal fiber for the rotational degrees of freedom. [Figure 14.1](#) shows a schematic of the bi-unit cube and the shell element.

The mapping of the bi-unit cube into the shell element is separated into two parts

$$\mathbf{x}(\xi, \eta, \zeta) = \bar{\mathbf{x}}(\xi, \eta) + \mathbf{X}(\xi, \eta, \zeta), \quad (14.3)$$

where $\bar{\mathbf{x}}$ denotes a position vector to a point on the reference surface of the shell and \mathbf{X} is a position vector, based at point $\bar{\mathbf{x}}$ on the reference, that defines the fiber direction through that point. In particular, if we consider one of the four nodes which define the reference surface, then

$$\bar{\mathbf{x}}(\xi, \eta) = N_a(\xi, \eta) \bar{\mathbf{x}}_a, \quad (14.4)$$

$$\mathbf{X}(\xi, \eta, \zeta) = N_a(\xi, \eta) \mathbf{X}_a(\zeta). \quad (14.5)$$

With this description, arbitrary points on the reference surface $\bar{\mathbf{x}}$ are interpolated by the two-dimensional shape function $N(\xi, \eta)$ operating on the global position of the four shell nodes that define the reference surfaces, i.e., $\bar{\mathbf{x}}_a$. Points off the reference surface are further interpolated by using a one-dimensional shape function along the fiber direction, i.e., $\mathbf{X}_a(\zeta)$, where

$$\mathbf{X}_a(\zeta) = z_a(\zeta) \hat{\mathbf{X}}_a, \quad (14.6)$$

$$z_a(\zeta) = N_+(\zeta) z_a^+ + N_-(\zeta) z_a^-, \quad (14.7)$$

$$N_+(\zeta) = \frac{(1 + \zeta)}{2}, \quad (14.8)$$

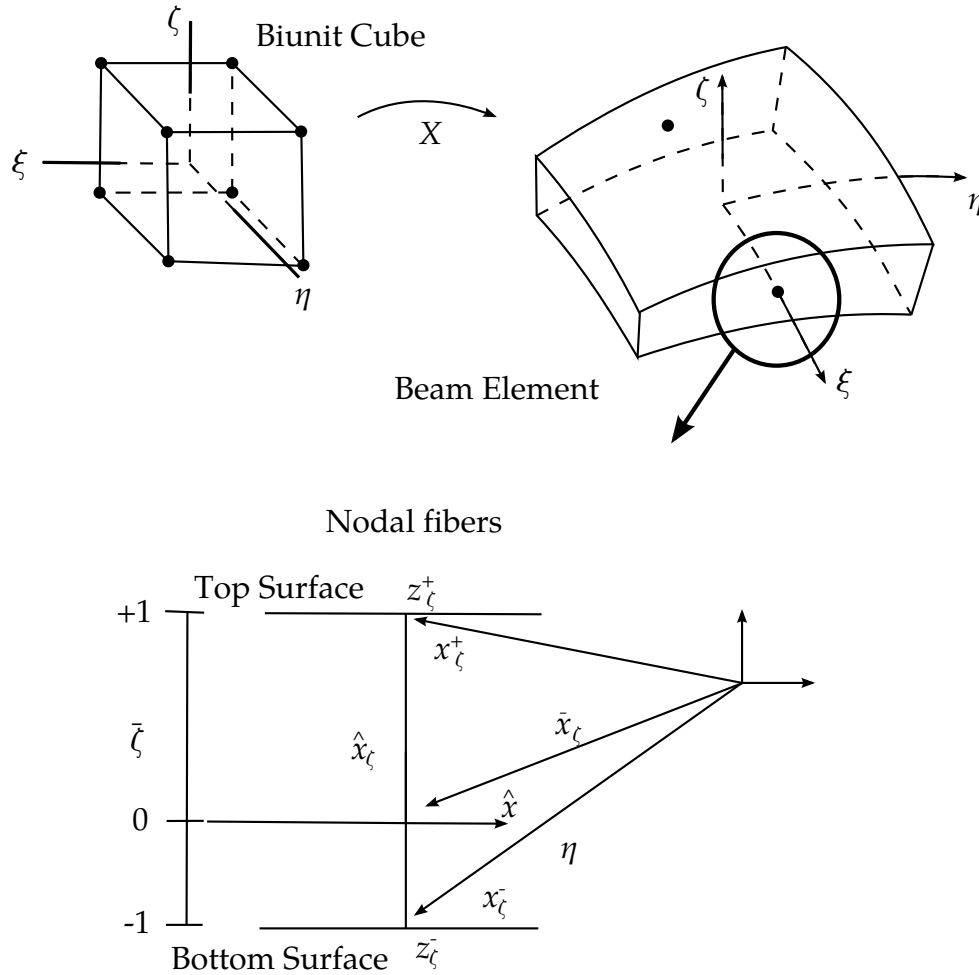


Figure 14.1. Mapping of the biunit cube into the Hughes-Liu shell element and nodal fiber nomenclature.

$$N_-(\zeta) = \frac{(1 - \zeta)}{2} \quad (14.9)$$

As shown in the lower portion of [Figure 14.1](#), $\hat{\mathbf{x}}_\zeta$ is a unit vector in the fiber direction and $z(\zeta)$ is a thickness function. (Thickness changes (see [Hughes and Carnoy 1981]) are accounted for by explicitly adjusting the fiber lengths at the completion of a time step based on the amount of straining in the fiber direction. Updates of the fiber lengths always lag one time step behind other kinematical quantities.)

The reference surface may be located at the mid-surface of the shell or at either of the shell's outer surfaces. This capability is useful in several practical situations involving contact surfaces, connection of shell elements to solid elements, and offsetting elements such as stiffeners in stiffened shells. The reference surface is located within the shell element by specifying the value of the parameter $\bar{\zeta}$ (see lower portion of [Figure 14.1](#)). When $\bar{\zeta} = -1, 0, +1$, the reference surface is located at the bottom, middle, and top surface of the shell, respectively.

The Hughes-Liu formulation uses two position vectors, in addition to $\bar{\zeta}$, to locate the reference surface and define the initial fiber direction. The two position vectors x_a^+ and x_a^- are located on the top and bottom surfaces, respectively, at node a . From these data the following are obtained:

$$\bar{x}_a = \frac{1}{2}(1 - \bar{\zeta})x_a^- + (1 + \bar{\zeta})x_a^+, \quad (14.10)$$

$$\hat{X}_a = \frac{(x_a^+ - x_a^-)}{h_a}, \quad (14.11)$$

$$z_a^+ = \frac{1}{2}(1 - \bar{\zeta})h_a, \quad (14.12)$$

$$z_a^- = -\frac{1}{2}(1 + \bar{\zeta})h_a, \quad (14.13)$$

$$h_a = \|x_a^+ - x_a^-\|, \quad (14.14)$$

where $\|\cdot\|$ is the Euclidean norm.

14.2 Kinematics

The same parametric representation used to describe the geometry of the shell element, i.e., reference surface and fiber vector interpolation, are used to interpolate the shell element displacement, i.e., an isoparametric representation. Again, the displacements are separated into the reference surface displacements and rotations associated with the fiber direction:

$$\mathbf{u}(\xi, \eta, \zeta) = \bar{\mathbf{u}}(\xi, \eta) + \mathbf{U}(\xi, \eta, \zeta), \quad (14.15)$$

$$\bar{\mathbf{u}}(\xi, \eta) = N_a(\xi, \eta)\bar{\mathbf{u}}_a, \quad (14.16)$$

$$\mathbf{U}(\xi, \eta, \zeta) = N_a(\xi, \eta)\mathbf{U}_a(\zeta), \quad (14.17)$$

$$\mathbf{U}_a(\zeta) = z_a(\zeta)\hat{\mathbf{U}}_a, \quad (14.18)$$

where \mathbf{u} is the displacement of a generic point; $\bar{\mathbf{u}}$ is the displacement of a point on the reference surface, and \mathbf{U} is the ‘fiber displacement’ rotations; the motion of the fibers can be interpreted as either displacements or rotations as will be discussed.

Hughes and Liu introduce the notation that follows, and the associated schematic shown in [Figure 14.2](#), to describe the current deformed configuration with respect to the reference configuration:

$$\mathbf{y} = \bar{\mathbf{y}} + \mathbf{Y}, \quad (14.19)$$

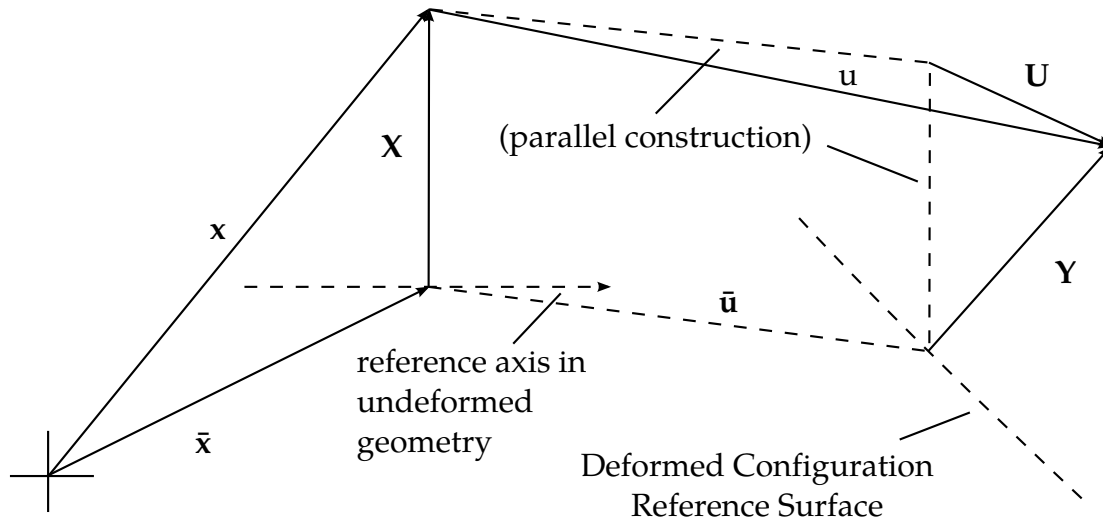


Figure 14.2. Schematic of deformed configuration displacements and position vectors.

$$\bar{\mathbf{y}} = \bar{\mathbf{x}} + \bar{\mathbf{u}}, \quad (14.20)$$

$$\bar{\mathbf{y}}_a = \bar{\mathbf{x}}_a + \bar{\mathbf{u}}_a, \quad (14.21)$$

$$\mathbf{Y} = \mathbf{X} + \mathbf{U}, \quad (14.22)$$

$$\mathbf{Y}_a = \mathbf{X}_a + \mathbf{U}_a, \quad (14.23)$$

$$\hat{\mathbf{Y}}_a = \hat{\mathbf{X}}_a + \hat{\mathbf{U}}_a. \quad (14.24)$$

In the above relations, and in Figure 14.2, the \mathbf{x} quantities refer to the reference configuration, the \mathbf{y} quantities refer to the updated (deformed) configuration and the \mathbf{u} quantities are the displacements. The notation consistently uses a superscript bar ($\bar{\cdot}$) to indicate reference surface quantities, a superscript caret ($\hat{\cdot}$) to indicate unit vector quantities, lower case letters for translational displacements, and upper case letters indicating fiber displacements. To update to the deformed configuration, two vector quantities are needed: the reference surface displacement $\bar{\mathbf{u}}$ and the associated nodal fiber displacement \mathbf{U} . The nodal fiber displacements are defined in the fiber coordinate system, described in the next subsection.

14.2.1 Fiber Coordinate System

For a shell element with four nodes, the known quantities will be the displacements of the reference surface $\bar{\mathbf{u}}$ obtained from the translational equations of motion and some rotational quantities at each node obtained from the rotational equations of motion. To complete the kinematics, we now need a relation between nodal rotations and fiber displacements \mathbf{U} .

At each node a unique local Cartesian coordinate system is constructed that is used as the reference frame for the rotation increments. The relation presented by Hughes and

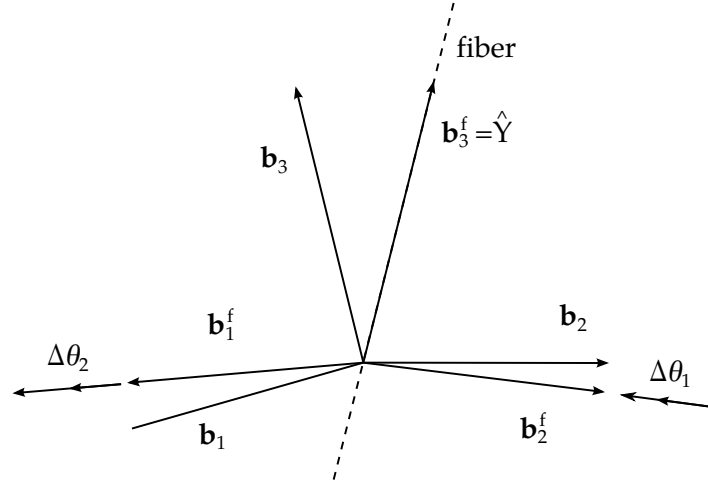


Figure 14.3. Incremental update of fiber vectors using Hughes-Liu incremental rotations.

Liu for the nodal fiber displacements (rotations) is an incremental relation, i.e., it relates the current configuration to the last state, not to the initial configuration. Figure 14.3 shows two triads of unit vectors: $(\mathbf{b}_1^f, \mathbf{b}_2^f, \mathbf{b}_3^f)$ comprising the orthonormal fiber basis in the reference configuration (where the fiber unit vector is now $\hat{\mathbf{Y}} = \mathbf{b}_3^f$) and $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ indicating the incrementally updated current configuration of the fiber vectors. The reference triad is updated by applying the incremental rotations, $\Delta\theta_1$ and $\Delta\theta_2$, obtained from the rotational equations of motion, to the fiber vectors $(\mathbf{b}_1^f$ and $\mathbf{b}_2^f)$ as shown in Figure 14.3. The linearized relationship between the components of $\Delta\hat{\mathbf{U}}$ in the fiber system viz, $\Delta\hat{\mathbf{U}}_1^f, \Delta\hat{\mathbf{U}}_2^f, \Delta\hat{\mathbf{U}}_3^f$, and the incremental rotations is given by

$$\begin{Bmatrix} \Delta\hat{\mathbf{U}}_1^f \\ \Delta\hat{\mathbf{U}}_2^f \\ \Delta\hat{\mathbf{U}}_3^f \end{Bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{Bmatrix}. \quad (14.25)$$

Although the above Hughes-Liu relation for updating the fiber vector enables a reduction in the number of nodal degrees of freedom from six to five, it is not implemented in LS-DYNA because it is not applicable to beam elements.

In LS-DYNA, three rotational increments are used, defined with reference to the global coordinate axes:

$$\begin{Bmatrix} \Delta\hat{\mathbf{U}}_1 \\ \Delta\hat{\mathbf{U}}_2 \\ \Delta\hat{\mathbf{U}}_3 \end{Bmatrix} = \begin{bmatrix} 0 & \hat{\mathbf{Y}}_3 & -\hat{\mathbf{Y}}_2 \\ -\hat{\mathbf{Y}}_3 & 0 & \hat{\mathbf{Y}}_1 \\ \hat{\mathbf{Y}}_2 & -\hat{\mathbf{Y}}_1 & 0 \end{bmatrix} \begin{Bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{Bmatrix}. \quad (14.26)$$

Equation (14.26) is adequate for updating the stiffness matrix, but for finite rotations the error is significant. A more accurate second-order technique is used in LS-DYNA for updating the unit fiber vectors:

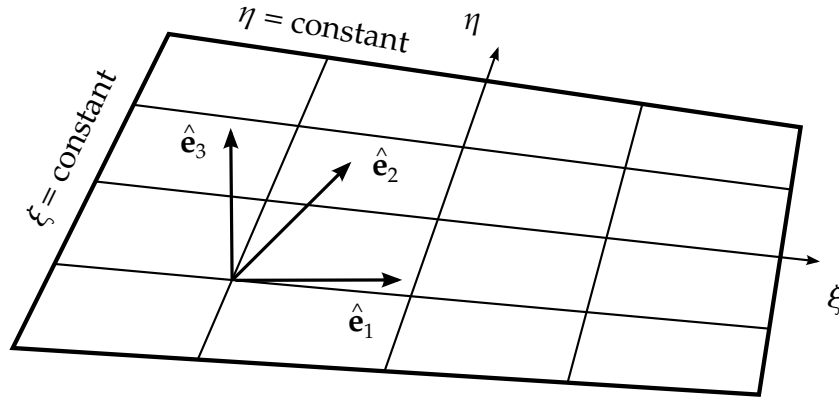


Figure 14.4. Schematic of lamina coordinate unit vectors.

$$\hat{Y}_i^{n+1} = R_{ij}(\Delta\theta) \hat{Y}_i^n, \quad (14.27)$$

$$R_{ij}(\Delta\theta) = \delta_{ij} + \frac{1}{2} \frac{(2\delta_{ij} + \Delta S_{ik}) \Delta S_{ik}}{D}, \quad (14.28)$$

$$\Delta S_{ij} = e_{ikj} \Delta\theta_k, \quad (14.29)$$

$$2D = 2 + \frac{1}{2} (\Delta\theta_1^2 + \Delta\theta_2^2 + \Delta\theta_3^2). \quad (14.30)$$

Here, δ_{ij} is the Kronecker delta and e_{ijk} is the permutation tensor. This rotational update is often referred to as the Hughes-Winget formula [Hughes and Winget 1980]. An exact rotational update using Euler angles or Euler parameters could easily be substituted in Equation (14.27), but it is doubtful that the extra effort would be justified.

14.2.2 Lamina Coordinate System

In addition to the above described fiber coordinate system, a local lamina coordinate system is needed to enforce the zero normal stress condition, i.e., plane stress. Lamina are layers through the thickness of the shell that correspond to the locations and associated thicknesses of the through-the-thickness shell integration points; the analogy is that of lamina in a fibrous composite material. The orthonormal lamina basis (Figure 14.4), with one direction \hat{e}_3 normal to the lamina of the shell, is constructed at every integration point in the shell.

The lamina basis is constructed by forming two unit vectors locally tangent to the lamina:

$$\hat{e}_1 = \frac{\mathbf{y}_{,\xi}}{\|\mathbf{y}_{,\xi}\|}, \quad (14.31)$$

$$\mathbf{e}'_2 = \frac{\mathbf{y}_{,\eta}}{\|\mathbf{y}_{,\eta}\|}, \quad (14.32)$$

where, as before, \mathbf{y} is the position vector in the current configuration. The normal to the lamina at the integration point is constructed from the vector cross product of these local tangents:

$$\hat{\mathbf{e}}_3 = \hat{\mathbf{e}}_1 \times \mathbf{e}'_2, \quad (14.33)$$

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1, \quad (14.34)$$

is defined, because $\hat{\mathbf{e}}_2$, although tangent to both the lamina and lines of constant ξ , may not be normal to $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_3$. The lamina coordinate system rotates rigidly with the element.

The transformation of vectors from the global to lamina coordinate system can now be defined in terms of the lamina basis vectors as

$$\hat{\mathbf{A}} = \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix}^T \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \mathbf{q}\mathbf{A}, \quad (14.35)$$

where e_{ix} , e_{iy} , e_{iz} are the global components of the lamina coordinate unit vectors; $\hat{\mathbf{A}}$ is a vector in the lamina coordinates, and \mathbf{A} is the same vector in the global coordinate system.

14.3 Strains and Stress Update

14.3.1 Incremental Strain and Spin Tensors

The strain and spin increments are calculated from the incremental displacement gradient

$$G_{ij} = \frac{\partial \Delta u_i}{\partial y_j}, \quad (14.36)$$

where Δu_i are the incremental displacements and y_j are the deformed coordinates. The incremental strain and spin tensors are defined as the symmetric and skew-symmetric parts, respectively, of G_{ij} :

$$\Delta \varepsilon_{ij} = \frac{1}{2}(G_{ij} + G_{ji}), \quad (14.37)$$

$$\Delta \omega_{ij} = \frac{1}{2}(G_{ij} - G_{ji}). \quad (14.38)$$

The incremental spin tensor $\Delta\omega_{ij}$ is used as an approximation to the rotational contribution of the Jaumann rate of the stress tensor; LS-DYNA implicit uses the more accurate Hughes-Winget transformation matrix (Equation (14.27)) with the incremental spin tensor for the rotational update. The Jaumann rate update is approximated as:

$$\underline{\sigma}_{ij}^{n+1} = \sigma_{ij}^n + \sigma_{ip}^n \Delta\omega_{pj} + \sigma_{jp}^n \Delta\omega_{pi}, \quad (14.39)$$

where the superscripts on the stress refer to the updated ($n + 1$) and reference (n) configurations. The Jaumann rate update of the stress tensor is applied in the global configuration before the constitutive evaluation is performed. In the Hughes-Liu shell the stresses and history variables are stored in the global coordinate system.

14.3.2 Stress Update

To evaluate the constitutive relation, the stresses and strain increments are rotated from the global to the lamina coordinate system using the transformation defined previously in Equation (14.35), viz.

$$\underline{\sigma}_{ij}^{l^{n+1}} = q_{ik} \underline{\sigma}_{kn}^{n+1} q_{jn}, \quad (14.40)$$

$$\Delta\varepsilon_{ij}^{l^{n+1/2}} = q_{ik} \Delta\varepsilon_{kn}^{n+1/2} q_{jn}, \quad (14.41)$$

where the superscript l indicates components in the lamina (local) coordinate system.

The stress is updated incrementally:

$$\sigma_{ij}^{l^{n+1}} = \underline{\sigma}_{ij}^{l^{n+1}} + \Delta\sigma_{ij}^{l^{n+1/2}}, \quad (14.42)$$

and rotated back to the global system:

$$\sigma_{ij}^{n+1} = q_{ki} \sigma_{kn}^{l^{n+1}} q_{nj}, \quad (14.43)$$

before computing the internal force vector.

14.3.3 Incremental Strain-Displacement Relations

The global stresses are now used to update the internal force vector

$$\mathbf{f}_a^{\text{int}} = \int \mathbf{T}_a^T \mathbf{B}_a^T \boldsymbol{\sigma} dv, \quad (14.44)$$

where $\mathbf{f}_a^{\text{int}}$ are the internal forces at node a , \mathbf{B}_a is the strain-displacement matrix in the lamina coordinate system associated with the displacements at node a , and \mathbf{T}_a is the transformation matrix relating the global and lamina components of the strain-displacement matrix. Because the B matrix relates six strain components to twenty-four displacements (six degrees of freedom at four nodes), it is convenient to partition the B matrix into four groups of six:

$$\mathbf{B} = [\mathbf{B}_1 \quad \mathbf{B}_2 \quad \mathbf{B}_3 \quad \mathbf{B}_4], \quad (14.45)$$

Each \mathbf{B}_a submatrix is further partitioned into a portion due to strain and spin:

$$\mathbf{B}_a = \begin{bmatrix} \mathbf{B}_a^\varepsilon \\ \mathbf{B}_a^\omega \end{bmatrix}, \quad (14.46)$$

$$\mathbf{B}_a^\varepsilon = \begin{bmatrix} B_1 & 0 & 0 & B_4 & 0 & 0 \\ 0 & B_2 & 0 & 0 & B_5 & 0 \\ \bar{B}_2 & \bar{B}_1 & 0 & \bar{B}_5 & \bar{B}_4 & 0 \\ 0 & \bar{B}_3 & \bar{B}_2 & 0 & \bar{B}_6 & \bar{B}_5 \\ \bar{B}_3 & 0 & \bar{B}_1 & \bar{B}_6 & 0 & \bar{B}_4 \end{bmatrix}, \quad (14.47)$$

$$\mathbf{B}_a^\omega = \begin{bmatrix} \bar{B}_2 & -\bar{B}_1 & 0 & \bar{B}_5 & -\bar{B}_4 & 0 \\ 0 & \bar{B}_3 & -\bar{B}_2 & 0 & \bar{B}_6 & -\bar{B}_5 \\ -\bar{B}_3 & 0 & \bar{B}_1 & -\bar{B}_6 & 0 & \bar{B}_4 \end{bmatrix}, \quad (14.48)$$

where

$$B_i = \begin{cases} N_{a,i} = \frac{\partial N_a}{\partial y_i^l} & \text{for } i = 1, 2, 3 \\ (N_a z_a)_{,i-3} = \frac{\partial (N_a z_a)}{\partial y_{i-3}^l} & \text{for } i = 4, 5, 6 \end{cases}. \quad (14.49)$$

Notes on strain-displacement relations:

- The derivatives of the shape functions are taken with respect to the lamina coordinate system, e.g., $y = qy$.
- The superscript bar indicates the B 's are evaluated at the center of the lamina $(0, 0, \zeta)$. The strain-displacement matrix uses the 'B-Bar' (\bar{B}) approach advocated by Hughes [1980]. In the NIKE3D and DYNA3D implementations, this entails replacing certain rows of the B matrix and the strain increments with their counterparts evaluated at the center of the element. In particular, the strain-displacement matrix is modified to produce constant shear and spin increments throughout the lamina.
- The resulting B -matrix is a 8×24 matrix. Although there are six strain and three rotations increments, the B matrix has been modified to account for the fact that σ_{33} will be zero in the integration of [Equation \(14.44\)](#).

14.4 Element Mass Matrix

Hughes, Liu, and Levit [Hughes et al., 1981] describe the procedure used to form the shell element mass matrix in problems involving explicit transient dynamics. Their procedure, which scales the rotary mass terms, is used for all shell elements in LS-DYNA including those formulated by Belytschko and his co-workers. This scaling permits large critical time step sizes without loss of stability.

The consistent mass matrix is defined by

$$\mathbf{M} = \int_{v_m} \rho \mathbf{N}^T \mathbf{N} dv_m, \quad (14.50)$$

but cannot be used effectively in explicit calculations where matrix inversions are not feasible. In LS-DYNA only three and four-node shell elements are used with linear interpolation functions; consequently, we compute the translational masses from the consistent mass matrix by row summing, leading to the following mass at element node a :

$$M_{\text{disp}_a} = \int_v \rho \phi_a dv. \quad (14.51)$$

The rotational masses are computed by scaling the translational mass at the node by the factor α :

$$M_{\text{rot}_a} = \alpha M_{\text{disp}_a}, \quad (14.52)$$

$$\alpha = \max\{\alpha_1, \alpha_2\}, \quad (14.53)$$

$$\alpha_1 = \langle z_a \rangle^2 + \frac{1}{12} [z_a]^2, \quad (14.54)$$

$$\alpha_2 = \frac{V}{8h}, \quad (14.55)$$

$$\langle z_a \rangle = \frac{(z_a^+ + z_a^-)}{2}, \quad (14.56)$$

$$[z_a] = z_a^+ - z_a^-. \quad (14.57)$$

14.5 Accounting for Thickness Changes

Hughes and Carnoy [1981] describe the procedure used to update the shell thickness due to large membrane stretching. Their procedure with any necessary modifications is used across all shell element types in LS-DYNA. One key to updating the thickness is an accurate calculation of the normal strain component $\Delta \varepsilon_{33}$. This strain component is easily obtained for elastic materials but can require an iterative algorithm for nonlinear material behavior. In LS-DYNA we therefore default to an iterative plasticity update to accurately determine $\Delta \varepsilon_{33}$.

Hughes and Carnoy integrate the strain tensor through the thickness of the shell in order to determine a mean value $\Delta \bar{\varepsilon}_{ij}$:

$$\Delta \bar{\varepsilon}_{ij} = \frac{1}{2} \int_{-1}^1 \Delta \varepsilon_{ij} d\zeta, \quad (14.58)$$

and then project it to determine the straining in the fiber direction:

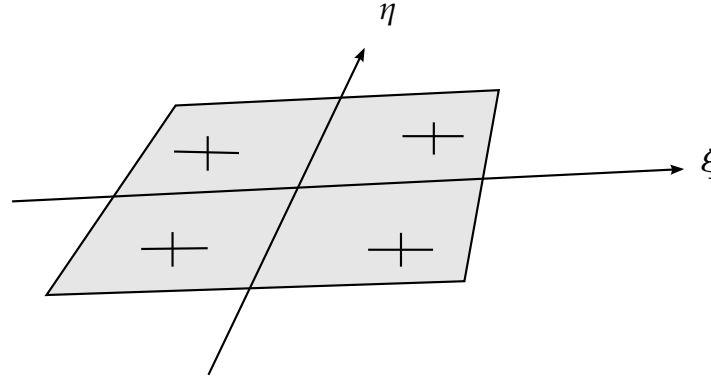


Figure 14.5. Selectively reduced integration rule results in four inplane points being used.

$$\bar{\epsilon}^f = \hat{\mathbf{Y}}^T \Delta \bar{\epsilon} \hat{\mathbf{Y}}. \quad (14.59)$$

Using the interpolation functions through the integration points the strains in the fiber directions are extrapolated to the nodal points if 2×2 selectively reduced integration is employed. The nodal fiber lengths can now be updated:

$$h_a^{n+1} = h_a^n (1 + \bar{\epsilon}_a^f). \quad (14.60)$$

14.6 Fully Integrated Hughes-Liu Shells

It is well known that one-point integration results in zero energy modes that must be resisted. The four-node under integrated shell with six degrees of freedom per node has nine zero energy modes, six rigid body modes, and four unconstrained drilling degrees of freedom. Deformations in the zero energy modes are always troublesome but usually not a serious problem except in regions where boundary conditions such as point loads are active. In areas where the zero energy modes are a problem, it is highly desirable to provide the option of using the original formulation of Hughes-Liu with selectively reduced integration.

The major disadvantages of full integration are two-fold:

1. nearly four times as much data must be stored;
2. the operation count increases three- to fourfold. The level 3 loop is added as shown in [Figure 14.6](#)

However, these disadvantages can be more than offset by the increased reliability and accuracy.

We have implemented two version of the Hughes-Liu shell with selectively reduced integration. The first closely follows the intent of the original paper, and therefore

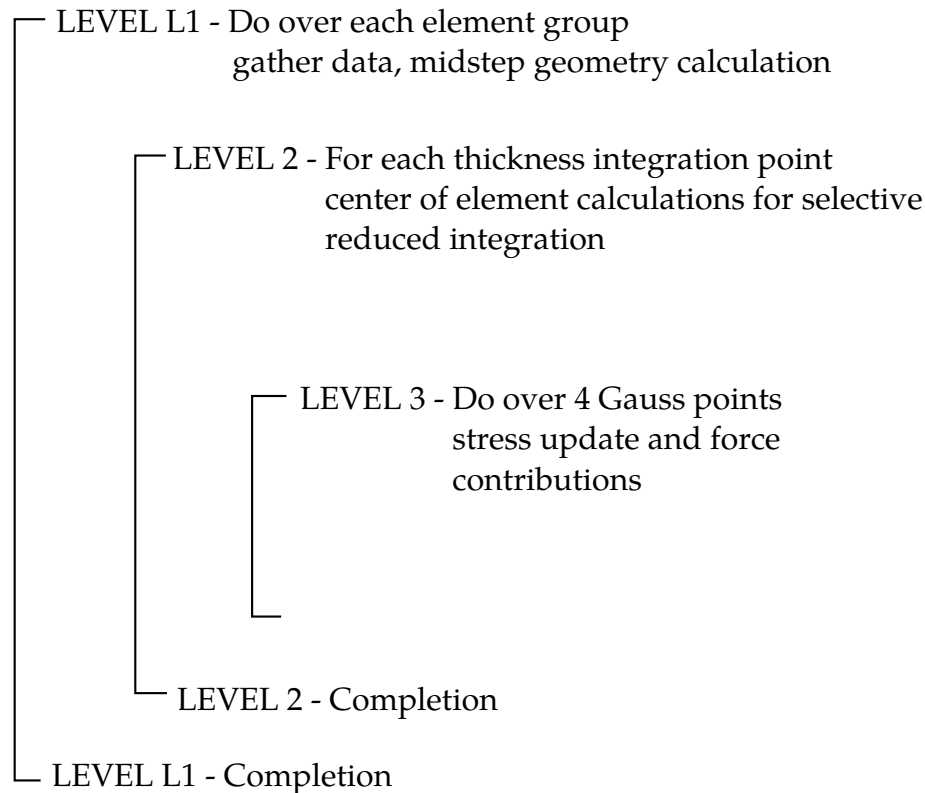


Figure 14.6. An inner loop, LEVEL 3, is added for the Hughes-Liu shell with selectively reduced integration.

no assumptions are made to reduce costs, which are outlined in operation counts in Table 10.1. These operation counts can be compared with those in Table 10.2 for the Hughes-Liu shell with uniformly reduced integration. The second formulation, which reduces the number of operation by more than a factor of two, is referred to as the co-rotational Hughes-Liu shell in the LS-DYNA user's manual. This shell is considerably cheaper due to the following simplifications:

- Strains rates are not centered. The strain displacement matrix is only computed at time $n + 1$ and not at time $n + 1 / 2$.
- The stresses are stored in the local shell system following the Belytschko-Tsay shell. The transformations of the stresses between the local and global coordinate systems are thus avoided.
- The Jaumann rate rotation is not performed, thereby avoiding even more computations. This does not necessarily preclude the use of the shell in large deformations.
- To study the effects of these simplifying assumptions, we can compare results with those obtained with the full Hughes-Liu shell. Thus far, we have been able to get comparable results.

LEVEL L1 - Once per element

Midstep translation geometry, etc.	204
Midstep calculation of \hat{Y}	318
LEVEL L2 - For each integration point through thickness (NT points)	
Strain increment at $(0, 0, \zeta)$	316
Hughes-Winget rotation matrix	33
Square root of Hughes-Winget matrix	47
Rotate strain increments into lamina coordinates	66
Calculate rows 3-8 of B matrix	919
LEVEL L3 - For each integration point in lamina	
Rotate stress to $n+1/2$ configuration	75
Incremental displacement gradient matrix	370
Rotate stress to lamina system	75
Rotate strain increments to lamina system	55
Constitutive model	model dependent
Rotate stress back to global system	69
Rotate stress to $n+1$ configuration	75
Calculate rows 1 and 2 of B matrix	358
Stresses in $n+1$ lamina system	75
Stress divergence	245
<hr/>	
TOTAL	522 + NT {1381 + 4 * 1397}

Table 10.1. Operation counts for the Hughes-Liu shell with selectively reduced integration.

LEVEL L1 - Once per element	
Calculate displacement increments	24
Element areas for time step	53
Calculate \hat{Y}	238
LEVEL L2 and L3 - Integration point through thickness (NT points)	
Incremental displacement gradient matrix	284
Jaumann rotation for stress	33
Rotate stress into lamina coordinates	75
Rotate strain increments into lamina coordinates	81
Constitutive model	model dependent
Rotate stress to $n+1$ global coordinates	69
Stress divergence	125
LEVEL L1 - Cleanup	
Finish stress divergence	60
Hourglass control	356
<hr/>	
TOTAL	731 + NT * 667

Table 10.2. Operation counts for the LS-DYNA implementation of the uniformly reduced Hughes-Liu shell.

15

Transverse Shear Treatment For Layered Shell

The shell element formulations that include the transverse shear strain components are based on the first order shear deformation theory, which yield constant through thickness transverse shear strains. This violates the condition of zero traction on the top and bottom surfaces of the shell. Normally, this is corrected by the use of a shear correction factor. The shear correction factor is 5/6 for isotropic materials; however, this value is incorrect for sandwich and laminated shells. Not accounting for the correct transverse shear strain and stress could yield a very stiff behavior in sandwich and laminated shells. This problem is addressed here by the use of the equilibrium equations without gradient in the y-direction as described by what follows. Consider the stresses in a layered shell:

$$\begin{aligned}\sigma_x^{(i)} &= C_{11}^{(i)}(\varepsilon_x^\circ + z\chi_x) + C_{12}^{(i)}(\varepsilon_y^\circ + z\chi_y) = C_{11}^{(i)}\varepsilon_x^\circ + C_{12}^{(i)}\varepsilon_y^\circ + z(C_{11}^{(i)}\chi_x + C_{12}^{(i)}\chi_y), \\ \sigma_y^{(i)} &= C_{12}^{(i)}\varepsilon_x^\circ + C_{22}^{(i)}\varepsilon_y^\circ + z(C_{12}^{(i)}\chi_x + C_{22}^{(i)}\chi_y), \\ \tau_{xy}^{(i)} &= C_{44}^{(i)}(\varepsilon_{xy}^\circ + z\chi_{xy}).\end{aligned}\quad (15.1)$$

Assume that the bending center \bar{z}_x is known. Then

$$\sigma_x^{(i)} = (z - \bar{z}_x)(C_{11}^{(i)}\chi_x + C_{12}^{(i)}\chi_y) + C_{11}^{(i)}\varepsilon_x(\bar{z}_x) + C_{12}^{(i)}\varepsilon_y(\bar{z}_x). \quad (15.2)$$

The bending moment is given by the following equation:

$$M_{xx} = \chi_x \left(\sum_{i=1}^{NL} C_{11}^{(i)} \int_{z_{i-1}}^{z_i} (z - \bar{z}_x)^2 dz \right) + \chi_y \left(\sum_{i=1}^{NL} C_{12}^{(i)} \int_{z_{i-1}}^{z_i} (z - \bar{z}_x)^2 dz \right) \quad (15.3)$$

or

$$\begin{aligned}M_{xx} &= \frac{1}{3} \left[\chi_x \sum_{i=1}^{NL} C_{11}^{(i)} [(z_i^3 - z_{i-1}^3) - (z_i - z_{i-1})\bar{z}_x^2] \right. \\ &\quad \left. + \chi_y \sum_{i=1}^{NL} C_{12}^{(i)} [(z_i^3 - z_{i-1}^3) - (z_i - z_{i-1})\bar{z}_x^2] \right]\end{aligned}\quad (15.4)$$

where “NL” is the number of layers in the material.

Assume $\varepsilon_y = 0$ and $\sigma_x = E_\chi \varepsilon_\chi$, let

$$(\overline{EI})_x = \sum_{i=1}^{NL} E_x^{(i)} [(z_i^3 - z_{i-1}^3) - (z_i - z_{i-1}) \bar{z}_x^2], \quad (15.5)$$

then

$$\varepsilon_x = \frac{z - \bar{z}_x}{\rho} = (z - \bar{z}_x) \chi_x, \quad (15.6)$$

and

$$M_{xx} = \frac{1}{3} (\chi_x (\overline{EI})_x), \quad (15.7)$$

$$\chi_x = \frac{3M_{xx}}{(\overline{EI})_x}. \quad (15.8)$$

Therefore, the stress becomes

$$\sigma_x^{(i)} = \frac{3M_{xx} E_x^{(i)} (z - \bar{z}_x)}{(\overline{EI})_x}. \quad (15.9)$$

Now considering the first equilibrium equation, one can write the following:

$$\frac{\partial \tau_{xz}}{\partial z} = -\frac{\partial \sigma_x}{\partial x} = -\frac{3Q_{xz} E_x^{(j)} (z - \bar{z}_x)}{(\overline{EI})_x}, \quad (15.10)$$

$$\tau_{xz}^{(j)} = -\frac{3Q_{xz} E_x^{(j)} \left(\frac{z^2}{2} - z \bar{z}_x \right)}{(\overline{EI})_x} + C_j, \quad (15.11)$$

where Q_{xz} is the shear force and C_j is the constant of integration. This constant is obtained from the transverse shear stress continuity requirement at the interface of each layer. Let then

$$C_j = \frac{Q_{xz} E_x^{(i)} \left(\frac{z_{i-1}^2}{2} - z_{i-1} \bar{z}_x \right)}{(\overline{EI})_x} + \tau_{xz}^{i-1}, \quad (15.12)$$

and

$$\tau_{xz}^{(i)} = \tau_{xz}^{(i-1)} + \frac{Q_{xz} E_x^{(i)}}{(\overline{EI})_x} \left[\frac{z_{i-1}^2}{2} - z_{i-1} \bar{z}_x - \frac{z^2}{2} + z \bar{z}_x \right]. \quad (15.13)$$

For the first layer

$$\tau_{xz} = -\frac{3Q_{xz} C_{11}^{(1)}}{(\overline{EI})_x} \left[\frac{z^2 - z_o^2}{2} - \bar{z}_x (z - z_o) \right], \quad (15.14)$$

for subsequent layers

$$\tau_{xz} = \tau_{xz}^{(i-1)} - \frac{3Q_{xz}C_{11}^{(i)}}{(\bar{EI})_x} \left[\frac{z^2 - z_{i-1}^2}{2} - \bar{z}_x(z - z_{i-1}) \right], \quad z_{i-1} \leq z \leq z_i. \quad (15.15)$$

Here $\tau_{xz}^{(i-1)}$ is the stress in previous layer at the interface with the current layer. The shear stress can also be expressed as follows:

$$\tau_{xz} = -\frac{3Q_{xz}C_{11}^{(i)}}{(\bar{EI})_x} \left[f_x^{(i)} + \frac{z^2 - z_{i-1}^2}{2} - \bar{z}_x(z - z_{i-1}) \right], \quad (15.16)$$

where

$$f_x^{(i)} = \frac{1}{C_{11}^{(i)}} \sum_{j=1}^{i-1} C_{11}^{(j)} h_j \left[\frac{z_j + z_{j+1}}{2} - \bar{z}_x \right], \quad (15.17)$$

and

$$h_j = z_j - z_{j-1}. \quad (15.18)$$

To find Q_{xz} , the shear force, assume that the strain energy expressed through average shear modules, \bar{C}_{66} , is equal to the strain energy expressed through the derived expressions as follows:

$$U = \frac{1}{2} \frac{Q_{xz}}{\bar{C}_{66}h} = \frac{1}{2} \int \frac{\tau_{xz}^2}{C_{66}} dz, \quad (15.19)$$

$$\begin{aligned} \frac{1}{\bar{C}_{66}} &= \frac{9h}{(\bar{EI})_x^2} \int \frac{C_{11}^2}{C_{66}} \left[f_x^{(i)} + \frac{(z^2 - z_{i-1}^2)}{2} - \bar{z}_x(z - z_{i-1}) \right]^2 dz \\ &= \frac{9h}{(\bar{EI})_x^2} \sum_{i=1}^{NL} \frac{(C_{11}^{(i)})^2}{C_{66}^{(i)}} \int_{z_{i-1}}^{z_i} \left[f_x^{(i)} + \frac{z^2 - z_{i-1}^2}{2} - \bar{z}_x(z - z_{i-1}) \right]^2 dz \\ &= \frac{1}{60} \frac{9h}{(\bar{EI})_x^2} \sum_i^{NL} \frac{(C_{11}^{(i)})^2 h}{C_{66}^{(i)}} \{ f_x^i [60f_x^i + 20h_i(z_i + 2z_{i-1} - 3\bar{z}_x)] \\ &\quad + \bar{z}_x h_i [20\bar{z}_x h_i + 35z_{i-1}^2 - 10z_{i-1}(z_i + z_{i-1}) - 15z_i^2] + z_i(z_i + z_{i-1})(3z_i^2 \\ &\quad - 7z_{i-1}^2) + 8z_{i-1}^4 \}, \end{aligned} \quad (15.20)$$

then

$$Q_{xz} = \bar{\tau}_{xz} h = \bar{C}_{66} \bar{\gamma}_{xz} h, \quad (15.21)$$

to calculate \bar{z}_x use τ_{xz} for last layer at surface $z = 0$,

$$\sum_{i=1}^{NL} C_{11}^{(i)} \left[\left(\frac{z_i^2 - z_{i-1}^2}{2} \right) - \bar{z}_x(z_i - z_{i-1}) \right] = 0, \quad (15.22)$$

where

$$\bar{z}_x = \frac{\sum_{i=1}^{NL} C_{11}^{(i)} h_i (z_i + z_{i+1})}{2 \sum_{i=1}^{NL} C_{11}^{(i)} h_i}. \quad (15.23)$$

Algorithm:

The following algorithm is used in the implementation of the transverse shear treatment.

1. Calculate \bar{z}_x according to [equation \(15.23\)](#)
2. Calculate f_x^i according to [equation \(15.17\)](#)
3. Calculate $\frac{1}{3} \sum_{i=1}^{NL} C_{11}^{(i)} (z_i^3 - z_{i-1}^3)$
4. Calculate $h \left[\frac{1}{3} \sum_{i=1}^{NL} C_{11}^{(i)} (z_i^3 - z_{i-1}^3) \right]^2$
5. Calculate \bar{C}_{66} according to [equation \(15.20\)](#)
6. Calculate $Q_{xz} = \bar{C}_{66} \bar{\gamma}_{xz} h$
7. Calculate τ_{xz} according to [equation \(15.16\)](#)

Steps 1-5 are performed at the initialization stage. Step 6 is performed in the shell formulation subroutine, and step 7 is performed in the stress calculation inside the constitutive subroutine.

16

Eight-Node Solid Shell Element

The isoparametric eight-node brick element discussed in Section 3 forms the basis for tshell formulation 1, a solid shell element with enhancements based on the Hughes-Liu and the Belytschko-Lin-Tsay shells. Like the eight-node brick, the geometry is interpolated from the nodal point coordinates as:

$$x_i(X_\alpha, t) = x_i(X_\alpha(\xi, \eta, \zeta), t) = \sum_{j=1}^8 \phi_j(\xi, \eta, \zeta) x_i^j(t), \quad (16.1)$$

$$\phi_j = \frac{1}{8} (1 + \xi \xi_j) (1 + \eta \eta_j) (1 + \zeta \zeta_j). \quad (16.2)$$

As with solid elements, \mathbf{N} is the 3×24 rectangular interpolation matrix:

$$\mathbf{N}(\xi, \eta, \zeta) = \begin{bmatrix} \varphi_1 & 0 & 0 & \varphi_2 & 0 & \dots & 0 & 0 \\ 0 & \varphi_1 & 0 & 0 & \varphi_2 & \dots & \varphi_8 & 0 \\ 0 & 0 & \varphi_1 & 0 & 0 & \dots & 0 & \varphi_8 \end{bmatrix}, \quad (16.3)$$

σ is the stress vector:

$$\sigma^T = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}), \quad (16.4)$$

and \mathbf{B} is the 6×24 strain-displacement matrix:

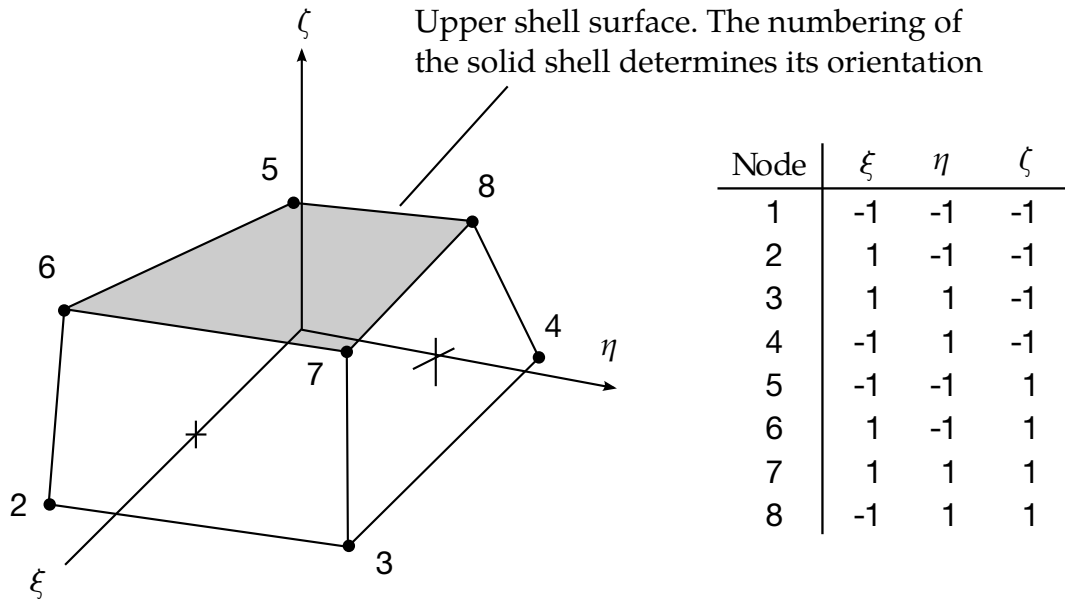


Figure 16.1. Eight node solid shell element

$$\mathbf{B} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \mathbf{N}, \quad (16.5)$$

Terms in the strain-displacement matrix are readily calculated. Note that

$$\begin{aligned} \frac{\partial \varphi_i}{\partial \xi} &= \frac{\partial \varphi_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial \varphi_i}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial \varphi_i}{\partial z} \frac{\partial z}{\partial \xi}, \\ \frac{\partial \varphi_i}{\partial \eta} &= \frac{\partial \varphi_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial \varphi_i}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial \varphi_i}{\partial z} \frac{\partial z}{\partial \eta}, \\ \frac{\partial \varphi_i}{\partial \zeta} &= \frac{\partial \varphi_i}{\partial x} \frac{\partial x}{\partial \zeta} + \frac{\partial \varphi_i}{\partial y} \frac{\partial y}{\partial \zeta} + \frac{\partial \varphi_i}{\partial z} \frac{\partial z}{\partial \zeta}, \end{aligned} \quad (16.6)$$

which can be rewritten as

$$\begin{bmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \\ \frac{\partial \varphi_i}{\partial z} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \\ \frac{\partial \varphi_i}{\partial z} \end{bmatrix}. \quad (16.7)$$

Inverting the Jacobian matrix, \mathbf{J} , we can solve for the desired terms

$$\begin{bmatrix} \frac{\partial \varphi_i}{\partial x} \\ \frac{\partial \varphi_i}{\partial y} \\ \frac{\partial \varphi_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial \varphi_i}{\partial \xi} \\ \frac{\partial \varphi_i}{\partial \eta} \\ \frac{\partial \varphi_i}{\partial \zeta} \end{bmatrix}. \quad (16.8)$$

To obtain shell-like behavior from the solid element, it is necessary to use multiple integration points through the shell thickness along the ζ axis while employing a plane stress constitutive subroutine. Consequently, it is necessary to construct a reference surface within the brick shell. We locate the reference surface midway between the upper and lower surfaces and construct a local coordinate system exactly as was done for the Belytschko-Lin-Tsay shell element. Following the procedure outlined in Section 7, Equations (7.1) – (7.3), the local coordinate system can be constructed as depicted in [Figure 16.2](#). Equation (7.5a) gives the transformation matrix in terms of the local basis:

$$\{\mathbf{A}\} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} \\ e_{1y} & e_{2y} & e_{3y} \\ e_{1z} & e_{2z} & e_{3z} \end{bmatrix} \begin{Bmatrix} \hat{A}_x \\ \hat{A}_y \\ \hat{A}_z \end{Bmatrix} = [\boldsymbol{\mu}] \{\hat{\mathbf{A}}\} = [\mathbf{q}]^T \{\hat{\mathbf{A}}\}. \quad (16.9)$$

As with the Hughes-Liu shell, the next step is to perform the Jaumann rate update:

$$\underline{\sigma}_{ij}^{n+1} = \sigma_{ij}^n + \sigma_{ip}^n \Delta \omega_{pj} + \sigma_{jp}^n \Delta \omega_{pi}, \quad (16.10)$$

to account for the material rotation between time steps n and $n + 1$. The Jaumann rate update of the stress tensor is applied in the global configuration before the constitutive evaluation is performed. In the solid shell, as in the Hughes-Liu shell, the stresses and history variables are stored in the global coordinate system. To evaluate the constitutive relation, the stresses and the strain increments are rotated from the global to the lamina coordinate system using the transformation defined previously:

$$\underline{\sigma}_{ij}^{l^{n+1}} = q_{ik} \underline{\sigma}_{kn}^{n+1} q_{jn}, \quad (16.11)$$

$$\Delta \varepsilon_{ij}^{l^{n+1/2}} = q_{ik} \Delta \varepsilon_{kn}^{n+1/2} q_{jn}, \quad (16.12)$$

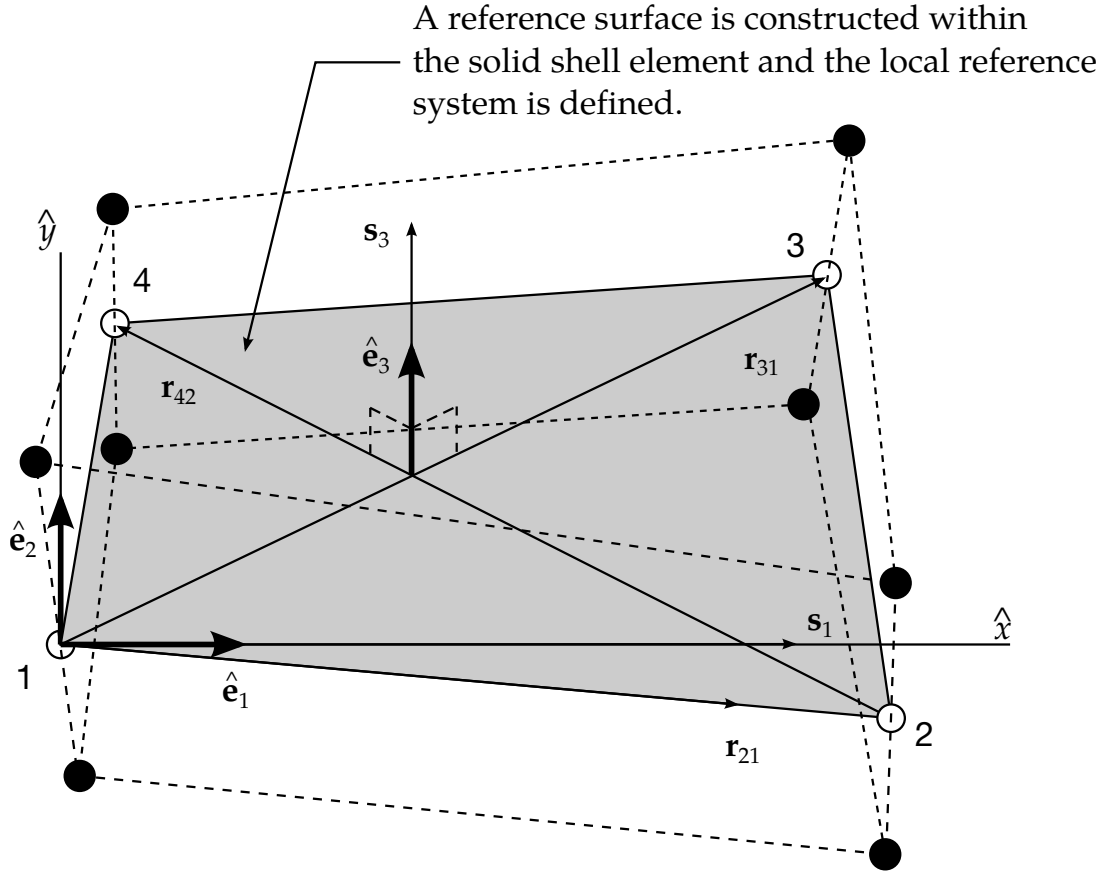


Figure 16.2. Construction of the reference surface in the solid shell element.

where the superscript l indicates components in the lamina (local) coordinate system. The stress is updated incrementally:

$$\sigma_{ij}^{l^{n+1}} = \sigma_{ij}^{l^{n+1}} + \Delta \sigma_{ij}^{l^{n+1/2}}. \quad (16.13)$$

Independently from the constitutive evaluation

$$\sigma_{33}^l = 0, \quad (16.14)$$

which ensures that the plane stress condition is satisfied, we update the normal stress which is used as a penalty to maintain the thickness of the shell:

$$(\sigma_{33}^{\text{penalty}})^{n+1} = (\sigma_{33}^{\text{penalty}})^n + E \Delta \epsilon_{33}^l, \quad (16.15)$$

where E is the elastic Young's modulus for the material. The stress tensor of Equation (16.13) is rotated back to the global system:

$$\sigma_{ij}^{n+1} = q_{ki} (\sigma_{kn}^l)^{n+1} q_{nj}. \quad (16.16)$$

A penalty stress tensor is then formed by transforming the normal penalty stress tensor (a null tensor except for the 33 term) back to the global system:

$$\left(\sigma_{ij}^{\text{penalty}}\right)^{n+1} = q_{ki} \left[\left(\sigma_{ij}^{\text{penalty}}\right)^l \right]^{n+1} q_{nj}, \quad (16.17)$$

before computing the internal force vector. The internal force vector can now be computed:

$$\mathbf{f}^{\text{int}} = \int (\mathbf{B}^{n+1})^T \left[\boldsymbol{\sigma}^{n+1} + \left(\sigma^{\text{penalty}}\right)^{n+1} \right] dv. \quad (16.18)$$

The brick shell exhibits no discernible locking problems with this approach.

The treatment of the hourglass modes is identical to that described for the solid elements in Section 3.

17

Eight-Node Solid Element for Thick Shell Simulations

17.1 Abstract

Tshell formulation 3 is an eight-node hexahedral element incorporated into LS-DYNA to simulate thick shell structures. The element formulation is derived in a co-rotational coordinate system and the strain operator is calculated with a Taylor series expansion about the center of the element. Special treatments are made on the dilatational strain component and shear strain components to eliminate the volumetric and shear locking. The use of consistent tangential stiffness and geometric stiffness greatly improves the convergence rate in implicit analysis.

17.2 Introduction

Large-scale finite element analyses are extensively used in engineering designs and process controls. For example, in automobile crashworthiness, hundreds of thousands of unknowns are involved in the computer simulation models, and in metal forming processing, tests in the design of new dies or new products are done by numerical computations instead of costly experiments. The efficiency of the elements is of crucial importance to speed up the design processes and reduce the computational costs for these problems. Over the past ten years, considerable progress has been achieved in developing fast and reliable elements.

In the simulation of shell structures, Belytschko-Lin-Tsay [Belytschko, 1984a] and Hughes-Liu [Hughes, 1981a and 1981b] shell elements are widely used. However, in some cases thick shell elements are more suitable. For example, in the sheet metal forming with large curvature, traditional thin shell elements cannot give satisfactory results.

Also thin shell elements cannot give us detailed strain information though the thickness. In LS-DYNA, the eight-node solid thick shell element is still based on the Hughes-Liu and Belytschko-Lin-Tsay shells [Hallquist, 1998]. A new eight-node solid element based on Liu, 1985, 1994 and 1998 is incorporated into LS-DYNA, intended for thick shell simulation. The strain operator of this element is derived from a Taylor series expansion and special treatments on strain components are utilized to avoid volumetric and shear locking.

The organization of this paper is as follows. The element formulations are described in the next section. Several numerical problems are studied in the third section, followed by the conclusions.

17.3 Element Formulations

17.3.1 Strain Operator

The new element is based on the eight-node hexahedral element proposed and enhanced by Liu, 1985, 1994, 1998. For an eight-node hexahedral element, the spatial coordinates, x_i , and the velocity components, v_i , in the element are approximated in terms of nodal values, x_{ia} and v_{ia} , by

$$x_i = \sum_{a=1}^8 N_a(\xi, \eta, \zeta) x_{ia}, \quad (17.1)$$

$$v_i = \sum_{a=1}^8 N_a(\xi, \eta, \zeta) v_{ia}, \quad i = 1, 2, 3, \quad (17.2)$$

$$N_a(\xi, \eta, \zeta) = \frac{1}{8} (1 + \xi_a \xi) (1 + \eta_a \eta) (1 + \zeta_a \zeta), \quad (17.3)$$

and the subscripts i and a denote coordinate components ranging from one to three and the element nodal numbers ranging from one to eight, respectively. The referential coordinates ξ , η , and ζ of node a are denoted by ξ_a , η_a , and ζ_a , respectively.

The strain rate (or rate of deformation), $\dot{\epsilon}$, is composed of six components,

$$\dot{\epsilon}^T = [\epsilon_{xx} \ \epsilon_{yy} \ \epsilon_{zz} \ \epsilon_{xy} \ \epsilon_{yz} \ \epsilon_{zx}], \quad (17.4)$$

and is related to the nodal velocities by a strain operator, $\bar{\mathbf{B}}$,

$$\dot{\epsilon} = \bar{\mathbf{B}}(\xi, \eta, \zeta) \mathbf{v}, \quad (17.5)$$

where

$$\mathbf{v}^T = [v_{x1} \ v_{y1} \ v_{z1} \ \cdots \ v_{x8} \ v_{y8} \ v_{z8}], \quad (17.6)$$

$$\bar{\mathbf{B}} = \begin{bmatrix} \bar{B}_{xx} \\ \bar{B}_{yy} \\ \bar{B}_{zz} \\ \bar{B}_{xy} \\ \bar{B}_{yz} \\ \bar{B}_{zx} \end{bmatrix} = \begin{bmatrix} B_1(1) & 0 & 0 & \cdots & B_1(8) & 0 & 0 \\ 0 & B_2(1) & 0 & \cdots & 0 & B_2(8) & 0 \\ 0 & 0 & B_3(1) & \cdots & 0 & 0 & B_3(8) \\ B_2(1) & B_1(1) & 0 & \cdots & B_2(8) & B_1(8) & 0 \\ 0 & B_3(1) & B_2(1) & \cdots & 0 & B_3(8) & B_2(8) \\ B_3(1) & 0 & B_1(1) & \cdots & B_3(8) & 0 & B_1(8) \end{bmatrix}, \quad (17.7)$$

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} N_{,x}(\xi, \eta, \zeta) \\ N_{,y}(\xi, \eta, \zeta) \\ N_{,z}(\xi, \eta, \zeta) \end{bmatrix}. \quad (17.8)$$

Unlike standard solid element where the strain operator is computed by differentiating the shape functions, the strain operator for this new element is expanded in a Taylor series about the element center up to bilinear terms as follows [Liu, 1994, 1998],

$$\begin{aligned} \bar{\mathbf{B}}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}(0) + \bar{\mathbf{B}}_{,\xi}(0)\xi + \bar{\mathbf{B}}_{,\eta}(0)\eta + \bar{\mathbf{B}}_{,\zeta}(0)\zeta \\ & + \bar{\mathbf{B}}_{,\xi\eta}(0)\xi\eta + \bar{\mathbf{B}}_{,\eta\zeta}(0)\eta\zeta + \bar{\mathbf{B}}_{,\zeta\xi}(0)\zeta\xi. \end{aligned} \quad (17.9)$$

Let

$$\mathbf{x}_1^T = \mathbf{x}^T = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8], \quad (17.10)$$

$$\mathbf{x}_2^T = \mathbf{y}^T = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7 \ y_8], \quad (17.11)$$

$$\mathbf{x}_3^T = \mathbf{z}^T = [z_1 \ z_2 \ z_3 \ z_4 \ z_5 \ z_6 \ z_7 \ z_8], \quad (17.12)$$

$$\boldsymbol{\xi}^T = [-1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1], \quad (17.13)$$

$$\boldsymbol{\eta}^T = [-1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1], \quad (17.14)$$

$$\boldsymbol{\zeta}^T = [-1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1], \quad (17.15)$$

the Jacobian matrix at the center of the element can be evaluated as

$$\mathbf{J}(0) = [J_{ij}] = \frac{1}{8} \begin{bmatrix} \boldsymbol{\xi}^T \mathbf{x} & \boldsymbol{\xi}^T \mathbf{y} & \boldsymbol{\xi}^T \mathbf{z} \\ \boldsymbol{\eta}^T \mathbf{x} & \boldsymbol{\eta}^T \mathbf{y} & \boldsymbol{\eta}^T \mathbf{z} \\ \boldsymbol{\zeta}^T \mathbf{x} & \boldsymbol{\zeta}^T \mathbf{y} & \boldsymbol{\zeta}^T \mathbf{z} \end{bmatrix}; \quad (17.16)$$

the determinant of the Jacobian matrix is denoted by j_0 and the inverse matrix of $\mathbf{J}(0)$ is denoted by \mathbf{D}

$$\mathbf{D} = [D_{ij}] = \mathbf{J}^{-1}(0). \quad (17.17)$$

The gradient vectors and their derivatives with respect to the natural coordinates at the center of the element are given as follows,

$$\mathbf{b}_1 = \mathbf{N}_{,x}(0) = \frac{1}{8}[D_{11}\xi + D_{12}\eta + D_{13}\zeta], \quad (17.18)$$

$$\mathbf{b}_2 = \mathbf{N}_{,y}(0) = \frac{1}{8}[D_{21}\xi + D_{22}\eta + D_{23}\zeta], \quad (17.19)$$

$$\mathbf{b}_3 = \mathbf{N}_{,z}(0) = \frac{1}{8}[D_{31}\xi + D_{32}\eta + D_{33}\zeta], \quad (17.20)$$

$$\mathbf{b}_{1,\xi} = \mathbf{N}_{,x\xi}(0) = \frac{1}{8}[D_{12}\gamma_1 + D_{13}\gamma_2], \quad (17.21)$$

$$\mathbf{b}_{2,\xi} = \mathbf{N}_{,y\xi}(0) = \frac{1}{8}[D_{22}\gamma_1 + D_{23}\gamma_2], \quad (17.22)$$

$$\mathbf{b}_{3,\xi} = \mathbf{N}_{,z\xi}(0) = \frac{1}{8}[D_{32}\gamma_1 + D_{33}\gamma_2], \quad (17.23)$$

$$\mathbf{b}_{1,\eta} = \mathbf{N}_{,x\eta}(0) = \frac{1}{8}[D_{11}\gamma_1 + D_{13}\gamma_3], \quad (17.24)$$

$$\mathbf{b}_{2,\eta} = \mathbf{N}_{,y\eta}(0) = \frac{1}{8}[D_{21}\gamma_1 + D_{23}\gamma_3], \quad (17.25)$$

$$\mathbf{b}_{3,\eta} = \mathbf{N}_{,z\eta}(0) = \frac{1}{8}[D_{31}\gamma_1 + D_{33}\gamma_3], \quad (17.26)$$

$$\mathbf{b}_{1,\zeta} = \mathbf{N}_{,x\zeta}(0) = \frac{1}{8}[D_{11}\gamma_2 + D_{12}\gamma_3], \quad (17.27)$$

$$\mathbf{b}_{2,\zeta} = \mathbf{N}_{,y\zeta}(0) = \frac{1}{8}[D_{21}\gamma_2 + D_{22}\gamma_3], \quad (17.28)$$

$$\mathbf{b}_{3,\zeta} = \mathbf{N}_{,z\zeta}(0) = \frac{1}{8}[D_{31}\gamma_2 + D_{32}\gamma_3], \quad (17.29)$$

$$\mathbf{b}_{1,\xi\eta} = \mathbf{N}_{,x\xi\eta}(0) = \frac{1}{8}[D_{13}\gamma_4 - (\mathbf{p}_1^T \mathbf{x}_i) \mathbf{b}_{i,\xi} - (\mathbf{r}_1^T \mathbf{x}_i) \mathbf{b}_{i,\eta}], \quad (17.30)$$

$$\mathbf{b}_{2,\xi\eta} = \mathbf{N}_{,y\xi\eta}(0) = \frac{1}{8}[D_{23}\gamma_4 - (\mathbf{b}_2^T \mathbf{x}_i) \mathbf{b}_{i,\xi} - (\mathbf{r}_2^T \mathbf{x}_i) \mathbf{b}_{i,\eta}], \quad (17.31)$$

$$\mathbf{b}_{3,\xi\eta} = \mathbf{N}_{,z\xi\eta}(0) = \frac{1}{8}[D_{33}\gamma_4 - (\mathbf{p}_3^T \mathbf{x}_i) \mathbf{b}_{i,\xi} - (\mathbf{r}_3^T \mathbf{x}_i) \mathbf{b}_{i,\eta}], \quad (17.32)$$

$$\mathbf{b}_{1,\eta\zeta} = \mathbf{N}_{,x\eta\zeta}(0) = \frac{1}{8}[D_{11}\gamma_4 - (\mathbf{q}_1^T \mathbf{x}_i) \mathbf{b}_{i,\eta} - (\mathbf{p}_1^T \mathbf{x}_i) \mathbf{b}_{i,\zeta}], \quad (17.33)$$

$$\mathbf{b}_{2,\eta\zeta} = \mathbf{N}_{,y\eta\zeta}(0) = \frac{1}{8}[D_{21}\gamma_4 - (\mathbf{q}_2^T \mathbf{x}_i) \mathbf{b}_{i,\eta} - (\mathbf{p}_2^T \mathbf{x}_i) \mathbf{b}_{i,\zeta}], \quad (17.34)$$

$$\mathbf{b}_{3,\eta\zeta} = \mathbf{N}_{,z\eta\zeta}(0) = \frac{1}{8} [D_{31}\gamma_4 - (\mathbf{q}_3^T \mathbf{x}_i) \mathbf{b}_{i,\eta} - (\mathbf{p}_3^T \mathbf{x}_i) \mathbf{b}_{i,\zeta}], \quad (17.35)$$

$$\mathbf{b}_{1,\zeta\zeta} = \mathbf{N}_{,x\zeta\zeta}(0) = \frac{1}{8} [D_{12}\gamma_4 - (\mathbf{r}_1^T \mathbf{x}_i) \mathbf{b}_{i,\zeta} - (\mathbf{q}_1^T \mathbf{x}_i) \mathbf{b}_{i,\zeta}], \quad (17.36)$$

$$\mathbf{b}_{2,\zeta\zeta} = \mathbf{N}_{,y\zeta\zeta}(0) = \frac{1}{8} [D_{22}\gamma_4 - (\mathbf{r}_2^T \mathbf{x}_i) \mathbf{b}_{i,\zeta} - (\mathbf{q}_2^T \mathbf{x}_i) \mathbf{b}_{i,\zeta}], \quad (17.37)$$

$$\mathbf{b}_{3,\zeta\zeta} = \mathbf{N}_{,z\zeta\zeta}(0) = \frac{1}{8} [D_{32}\gamma_4 - (\mathbf{r}_3^T \mathbf{x}_i) \mathbf{b}_{i,\zeta} - (\mathbf{q}_3^T \mathbf{x}_i) \mathbf{b}_{i,\zeta}], \quad (17.38)$$

where

$$\mathbf{p}_i = D_{i1}\mathbf{h}_1 + D_{i3}\mathbf{h}_3, \quad (17.39)$$

$$\mathbf{q}_i = D_{i1}\mathbf{h}_2 + D_{i2}\mathbf{h}_3, \quad (17.40)$$

$$\mathbf{r}_i = D_{i2}\mathbf{h}_1 + D_{i3}\mathbf{h}_2, \quad (17.41)$$

$$\gamma_\alpha = \mathbf{h}_\alpha - (\mathbf{h}_\alpha^T \mathbf{x}_i) \mathbf{b}_i, \quad (17.42)$$

and

$$\mathbf{h}_1^T = [1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad -1], \quad (17.43)$$

$$\mathbf{h}_2^T = [1 \quad -1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1], \quad (17.44)$$

$$\mathbf{h}_3^T = [1 \quad 1 \quad -1 \quad -1 \quad -1 \quad -1 \quad 1 \quad 1], \quad (17.45)$$

$$\mathbf{h}_4^T = [-1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1 \quad 1 \quad -1]. \quad (17.46)$$

In the above equations \mathbf{h}_1 is the $\zeta\eta$ -hourglass vector, \mathbf{h}_2 the $\eta\zeta$ -hourglass vector, \mathbf{h}_3 the $\zeta\zeta$ -hourglass vector and \mathbf{h}_4 the $\zeta\eta\zeta$ -hourglass vector. They are the zero energy-deformation modes associated with the one-point-quadrature element which result in a non-constant strain field in the element [Flanagan, 1981, Belytschko, 1984 and Liu, 1984]. The γ_α in equations (17.21)–(17.38) are the stabilization vectors. They are orthogonal to the linear displacement field and provide a consistent stabilization for the element.

The strain operators, $\bar{\mathbf{B}}(\zeta, \eta, \zeta)$, can be decomposed into two parts, the dilatational part, $\bar{\mathbf{B}}^{\text{dil}}(\zeta, \eta, \zeta)$, and the deviatoric part, $\bar{\mathbf{B}}^{\text{dev}}(\zeta, \eta, \zeta)$, both of which can be expanded about the element center as in Equation (17.9)

$$\begin{aligned} \bar{\mathbf{B}}^{\text{dil}}(\zeta, \eta, \zeta) &= \bar{\mathbf{B}}^{\text{dil}}(0) + \bar{\mathbf{B}}_{,\zeta}^{\text{dil}}(0)\zeta + \bar{\mathbf{B}}_{,\eta}^{\text{dil}}(0)\eta + \bar{\mathbf{B}}_{,\zeta}^{\text{dil}}(0)\zeta \\ &+ \bar{\mathbf{B}}_{,\zeta\eta}^{\text{dil}}(0)\zeta\eta + \bar{\mathbf{B}}_{,\eta\zeta}^{\text{dil}}(0)\eta\zeta + \bar{\mathbf{B}}_{,\zeta\zeta}^{\text{dil}}(0)\zeta\zeta, \end{aligned} \quad (17.47)$$

$$\begin{aligned}\bar{\mathbf{B}}^{\text{dev}}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}^{\text{dev}}(0) + \bar{\mathbf{B}}_{,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.48)$$

To avoid volumetric locking, the dilatational part of the strain operators is evaluated only at one quadrature point, the center of the element, i.e., they are constant terms

$$\bar{\mathbf{B}}^{\text{dil}}(\xi, \eta, \zeta) = \bar{\mathbf{B}}^{\text{dil}}(0). \quad (17.49)$$

To remove shear locking, the deviatoric strain submatrices can be written in an orthogonal co-rotational coordinate system rotating with the element as

$$\begin{aligned}\bar{\mathbf{B}}_{xx}^{\text{dev}}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}_{xx}^{\text{dev}}(0) + \bar{\mathbf{B}}_{xx,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{xx,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{xx,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{xx,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{xx,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{xx,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.50)$$

$$\begin{aligned}\bar{\mathbf{B}}_{yy}^{\text{dev}}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}_{yy}^{\text{dev}}(0) + \bar{\mathbf{B}}_{yy,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{yy,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{yy,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{yy,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{yy,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{yy,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.51)$$

$$\begin{aligned}\bar{\mathbf{B}}_{zz}^{\text{dev}}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}_{zz}^{\text{dev}}(0) + \bar{\mathbf{B}}_{zz,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{zz,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{zz,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{zz,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{zz,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{zz,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.52)$$

$$\bar{\mathbf{B}}_{xy}^{\text{dev}}(\xi, \eta, \zeta) = \bar{\mathbf{B}}_{xy}^{\text{dev}}(0) + \bar{\mathbf{B}}_{xy,\zeta}^{\text{dev}}(0)\zeta, \quad (17.53)$$

$$\bar{\mathbf{B}}_{yz}^{\text{dev}}(\xi, \eta, \zeta) = \bar{\mathbf{B}}_{yz}^{\text{dev}}(0) + \bar{\mathbf{B}}_{yz,\xi}^{\text{dev}}(0)\xi, \quad (17.54)$$

$$\bar{\mathbf{B}}_{zx}^{\text{dev}}(\xi, \eta, \zeta) = \bar{\mathbf{B}}_{zx}^{\text{dev}}(0) + \bar{\mathbf{B}}_{zx,\eta}^{\text{dev}}(0)\eta. \quad (17.55)$$

Here, only one linear term is left for shear strain components such that the modes causing shear locking are removed. The normal strain components keep all non-constant terms given in [equation \(17.48\)](#).

Summation of [equation \(17.49\)](#) and [equations \(17.50\)–\(17.55\)](#) yields the following strain submatrices which can eliminate the shear and volumetric locking:

$$\begin{aligned}\bar{\mathbf{B}}_{xx}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}_{xx}(0) + \bar{\mathbf{B}}_{xx,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{xx,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{xx,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{xx,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{xx,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{xx,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.56)$$

$$\begin{aligned}\bar{\mathbf{B}}_{yy}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}_{yy}(0) + \bar{\mathbf{B}}_{yy,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{yy,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{yy,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{yy,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{yy,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{yy,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.57)$$

$$\begin{aligned}\bar{\mathbf{B}}_{zz}(\xi, \eta, \zeta) = & \bar{\mathbf{B}}_{zz}(0) + \bar{\mathbf{B}}_{zz,\xi}^{\text{dev}}(0)\xi + \bar{\mathbf{B}}_{zz,\eta}^{\text{dev}}(0)\eta + \bar{\mathbf{B}}_{zz,\zeta}^{\text{dev}}(0)\zeta \\ & + \bar{\mathbf{B}}_{zz,\xi\eta}^{\text{dev}}(0)\xi\eta + \bar{\mathbf{B}}_{zz,\eta\zeta}^{\text{dev}}(0)\eta\zeta + \bar{\mathbf{B}}_{zz,\zeta\xi}^{\text{dev}}(0)\zeta\xi,\end{aligned}\quad (17.58)$$

$$\bar{B}_{xy}(\xi, \eta, \zeta) = \bar{B}_{xy}(0) + \bar{B}_{xy,\zeta}^{\text{dev}}(0)\zeta, \quad (17.59)$$

$$\bar{B}_{yz}(\xi, \eta, \zeta) = \bar{B}_{yz}(0) + \bar{B}_{yz,\xi}^{\text{dev}}(0)\xi, \quad (17.60)$$

$$\bar{B}_{zx}(\xi, \eta, \zeta) = \bar{B}_{zx}(0) + \bar{B}_{zx,\eta}^{\text{dev}}(0)\eta. \quad (17.61)$$

It is noted that the elements developed above cannot pass the patch test if the elements are skewed. To remedy this drawback, the gradient vectors defined in (17.18)–(17.20) are replaced by the uniform gradient matrices, proposed by Flanagan [1981],

$$\begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \end{bmatrix} = \frac{1}{V_e} \int_{\Omega_e} \begin{bmatrix} B_1(\xi, \eta, \zeta) \\ B_2(\xi, \eta, \zeta) \\ B_3(\xi, \eta, \zeta) \end{bmatrix} dV. \quad (17.62)$$

Where V_e is the element volume and the stabilization vector are redefined as

$$\tilde{\gamma}_\alpha = \mathbf{h}_\alpha - (\mathbf{h}_\alpha^T \mathbf{x}_i) \tilde{\mathbf{b}}_i. \quad (17.63)$$

The element using the strain submatrices (17.56)–(17.61) and uniform gradient matrices (17.62) with four-point quadrature scheme is called HEXDS element.

17.3.2 Co-rotational Coordinate System

In elements for shell/plate structure simulations, the elimination of the shear locking depends on the proper treatment of the shear strain. It is necessary to attach a local

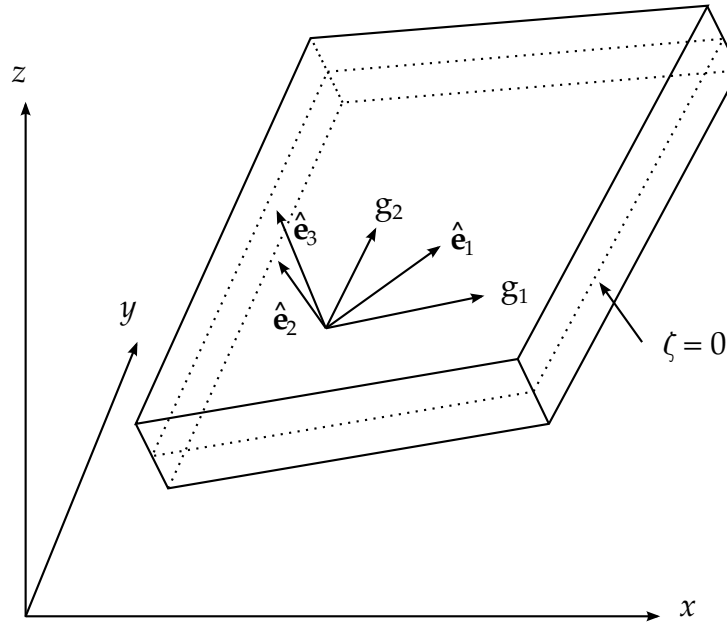


Figure 17.1. Definition of co-rotational coordinate system

coordinate system to the element so that the strain tensor in this local system is relevant

for the treatment. The co-rotational coordinate system determined here is one of the most convenient ways to define such a local system.

A co-rotational coordinate system is defined as a Cartesian coordinate system which rotates with the element. Let $\{x_a, y_a, z_a\}$ denote the current nodal spatial coordinates in the global system. For each quadrature point with natural coordinates (ξ, η, ζ) , we can have two tangent directions on the mid-surface ($\zeta = 0$) within the element (see Fig. 17.1)

$$\mathbf{g}_1 = \frac{\partial \mathbf{x}}{\partial \xi} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \end{bmatrix} = [N_{a,\xi} x_a \quad N_{a,\xi} y_a \quad N_{a,\xi} z_a]_{(\xi,\eta,0)}, \quad (17.64)$$

$$\mathbf{g}_2 = \frac{\partial \mathbf{x}}{\partial \eta} = \begin{bmatrix} \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{bmatrix} = [N_{a,\eta} x_a \quad N_{a,\eta} y_a \quad N_{a,\eta} z_a]_{(\xi,\eta,0)}. \quad (17.65)$$

The unit vector $\hat{\mathbf{e}}_1$ of the co-rotational coordinate system is defined as the bisector of the angle intersected by these two tangent vectors \mathbf{g}_1 and \mathbf{g}_2 ; the unit vector $\hat{\mathbf{e}}_3$ is perpendicular to the mid-surface and the other unit vector is determined by $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_3$, i.e.,

$$\hat{\mathbf{e}}_1 = \left(\frac{\mathbf{g}_1}{|\mathbf{g}_1|} + \frac{\mathbf{g}_2}{|\mathbf{g}_2|} \right) / \left(\left| \frac{\mathbf{g}_1}{|\mathbf{g}_1|} + \frac{\mathbf{g}_2}{|\mathbf{g}_2|} \right| \right), \quad (17.66)$$

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{|\mathbf{g}_1 \times \mathbf{g}_2|}, \quad (17.67)$$

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1, \quad (17.68)$$

which lead to the transformation matrix

$$\mathbf{R} = \begin{bmatrix} \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \\ \hat{\mathbf{e}}_3 \end{bmatrix}. \quad (17.69)$$

17.3.3 Stress and Strain Measures

Since the co-rotational coordinate system rotates with the configuration, the stress defined in this co-rotational system does not change with the rotation or translation of the material body and is thus objective. Therefore, we use the Cauchy stress in the co-rotational coordinate system, called the co-rotational Cauchy stress, as our stress measure.

The rate of deformation (or velocity strain tensor), also defined in the co-rotational coordinate system, is used as the measure of the strain rate,

$$\dot{\mathbf{e}} = \hat{\mathbf{d}} = \frac{1}{2} \left[\frac{\partial \hat{\mathbf{v}}^{\text{def}}}{\partial \hat{\mathbf{x}}} + \left(\frac{\partial \hat{\mathbf{v}}^{\text{def}}}{\partial \hat{\mathbf{x}}} \right)^T \right], \quad (17.70)$$

where $\hat{\mathbf{v}}^{\text{def}}$ is the deformation part of the velocity in the co-rotational system $\hat{\mathbf{x}}$. If the initial strain $\hat{\boldsymbol{\varepsilon}}(\mathbf{X}, 0)$ is given, the strain tensor can be expressed as,

$$\hat{\boldsymbol{\varepsilon}}(\mathbf{X}, t) = \hat{\boldsymbol{\varepsilon}}(\mathbf{X}, 0) + \int_0^t \hat{\mathbf{d}}(\mathbf{X}, \tau) d\tau. \quad (17.71)$$

The strain increment is then given by the mid-point integration of the velocity strain tensor,

$$\Delta \hat{\boldsymbol{\varepsilon}} = \int_{t_n}^{t_{n+1}} \hat{\mathbf{d}} d\tau \doteq \frac{1}{2} \left[\frac{\partial \Delta \hat{\mathbf{u}}^{\text{def}}}{\partial \hat{\mathbf{x}}_{n+\frac{1}{2}}} + \left(\frac{\partial \Delta \hat{\mathbf{u}}^{\text{def}}}{\partial \hat{\mathbf{x}}_{n+\frac{1}{2}}} \right)^T \right], \quad (17.72)$$

where $\Delta \hat{\mathbf{u}}^{\text{def}}$ is the deformation part of the displacement increment in the co-rotational system $\hat{\mathbf{x}}_{n+\frac{1}{2}}$ referred to the mid-point configuration.

17.3.4 Co-rotational Stress and Strain Updates

For stress and strain updates, we assume that all variables at the previous time step t_n are known. Since the stress and strain measures defined in the earlier section are objective in the co-rotational system, we only need to calculate the strain increment from the displacement field within the time increment $[t_n, t_{n+1}]$. The stress is then updated by using the radial return algorithm.

All the kinematical quantities must be computed from the last time step configuration, Ω_n , at $t = t_n$ and the current configuration, Ω_{n+1} at $t = t_{n+1}$ since these are the only available data. Denoting the spatial coordinates of these two configurations as \mathbf{x}_n and \mathbf{x}_{n+1} in the fixed global Cartesian coordinate system $O\mathbf{x}$, as shown in [Figure 17.2](#), the coordinates in the corresponding co-rotational Cartesian coordinate systems, $O\hat{\mathbf{x}}_n$ and $O\hat{\mathbf{x}}_{n+1}$, can be obtained by the following transformation rules:

$$\hat{\mathbf{x}}_n = \mathbf{R}_n \mathbf{x}_n, \quad (17.73)$$

$$\hat{\mathbf{x}}_{n+1} = \mathbf{R}_{n+1} \mathbf{x}_{n+1}, \quad (17.74)$$

where \mathbf{R}_n and \mathbf{R}_{n+1} are the orthogonal transformation matrices which rotate the global coordinate system to the corresponding co-rotational coordinate systems, respectively.

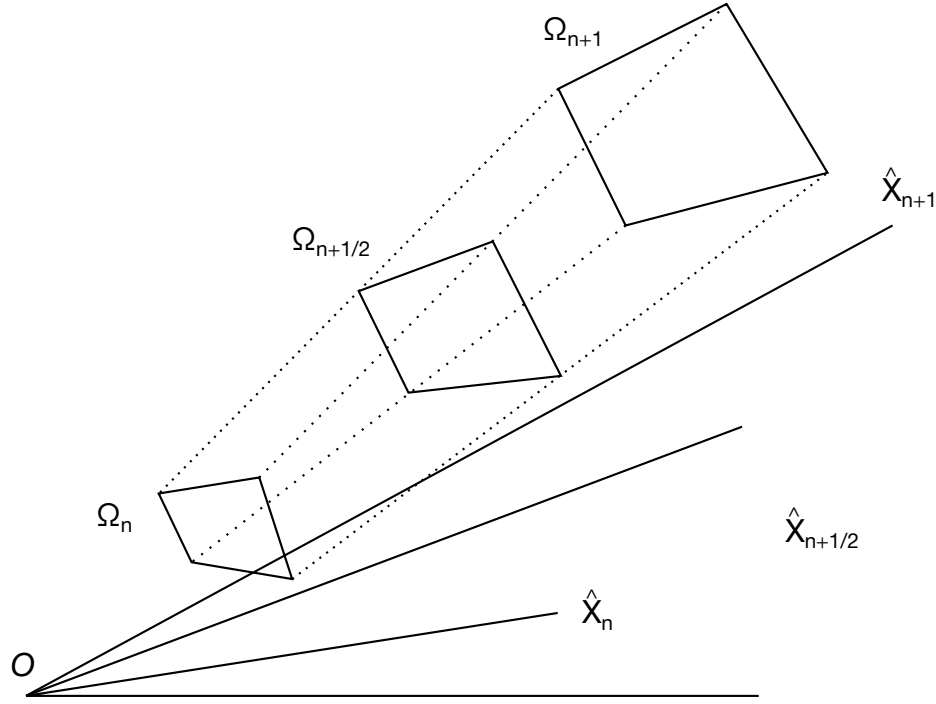


Figure 17.2. Configurations at times t_n , $t_{n+\frac{1}{2}}$, and t_{n+1} ,

Since the strain increment is referred to the configuration at $t = t_{n+\frac{1}{2}}$, by assuming the velocities within the time increment $[t_n, t_{n+1}]$ are constant, we have

$$\mathbf{x}_{n+\frac{1}{2}} = \frac{1}{2}(\mathbf{x}_n + \mathbf{x}_{n+1}), \quad (17.75)$$

and the transformation to the co-rotational system associated with this mid-point configuration, $\Omega_{n+\frac{1}{2}}$, is given by

$$\hat{\mathbf{x}}_{n+\frac{1}{2}} = \mathbf{R}_{n+\frac{1}{2}} \mathbf{x}_{n+\frac{1}{2}}. \quad (17.76)$$

Similar to polar decomposition, an incremental deformation can be separated into the summation of a pure deformation and a pure rotation [Belytschko, 1973]. Letting $\Delta \mathbf{u}$ indicate the displacement increment within the time increment $[t_n, t_{n+\frac{1}{2}}]$, we write

$$\Delta \mathbf{u} = \Delta \mathbf{u}^{\text{def}} + \Delta \mathbf{u}^{\text{rot}}, \quad (17.77)$$

where $\Delta \mathbf{u}^{\text{def}}$ and $\Delta \mathbf{u}^{\text{rot}}$ are, respectively, the deformation part and the pure rotation part of the displacement increment in the global coordinate system. The deformation part also includes the translation displacements which cause no strains.

In order to obtain the deformation part of the displacement increment referred to the configuration at $t = t_{n+\frac{1}{2}}$, we need to find the rigid rotation from Ω_n to Ω_{n+1}

provided that the mid-point configuration, $\Omega_{n+\frac{1}{2}}$, is held still. Defining two virtual configurations, Ω'_n and Ω'_{n+1} , by rotating the element bodies Ω_n and Ω_{n+1} into the co-rotational system $O\hat{\mathbf{x}}_{n+\frac{1}{2}}$ (Fig. 13.3) and denoting $\hat{\mathbf{x}}'_{n+1}$ as the coordinates of Ω'_n and Ω'_{n+1} in the co-rotational system $O\hat{\mathbf{x}}_{n+\frac{1}{2}}$, we have

$$\hat{\mathbf{x}}'_n = \hat{\mathbf{x}}_n, \quad \hat{\mathbf{x}}'_{n+1} = \hat{\mathbf{x}}_{n+1}. \quad (17.78)$$

We can see that from Ω_n to Ω'_n and from Ω'_{n+1} to Ω_{n+1} , the body experiences two rigid rotations and the rotation displacements are given by

$$\Delta \mathbf{u}_1^{\text{rot}} = \mathbf{x}'_n - \mathbf{x}_n = \mathbf{R}_{n+\frac{1}{2}}^T \hat{\mathbf{x}}'_n - \mathbf{x}_n = \mathbf{R}_{n+\frac{1}{2}}^T \hat{\mathbf{x}}_n - \mathbf{x}_n, \quad (17.79)$$

$$\Delta \mathbf{u}_2^{\text{rot}} = \mathbf{x}_{n+1} - \mathbf{x}'_{n+1} = \mathbf{x}_{n+1} - \mathbf{R}_{n+\frac{1}{2}}^T \hat{\mathbf{x}}'_{n+1} = \mathbf{x}_{n+1} - \mathbf{R}_{n+\frac{1}{2}}^T \hat{\mathbf{x}}_{n+1}. \quad (17.80)$$

Thus the total rotation displacement increment can be expressed as

$$\begin{aligned} \Delta \mathbf{u}^{\text{rot}} &= \Delta \mathbf{u}_1^{\text{rot}} + \Delta \mathbf{u}_2^{\text{rot}} = \mathbf{x}_{n+1} - \mathbf{x}_n - \mathbf{R}_{n+\frac{1}{2}}^T (\hat{\mathbf{x}}_{n+1} - \hat{\mathbf{x}}_n) \\ &= \Delta \mathbf{u} - \mathbf{R}_{n+\frac{1}{2}}^T (\hat{\mathbf{x}}_{n+1} - \hat{\mathbf{x}}_n). \end{aligned} \quad (17.81)$$

Then the deformation part of the displacement increment referred to the configuration $\Omega_{n+\frac{1}{2}}$ is

$$\Delta \mathbf{u}^{\text{def}} = \Delta \mathbf{u} - \Delta \mathbf{u}^{\text{rot}} = \mathbf{R}_{n+\frac{1}{2}}^T (\hat{\mathbf{x}}_{n+1} - \hat{\mathbf{x}}_n). \quad (17.82)$$

Therefore, the deformation displacement increment in the co-rotational coordinate system $O\hat{\mathbf{x}}_{n+\frac{1}{2}}$ is obtained as

$$\Delta \hat{\mathbf{u}}^{\text{def}} = \mathbf{R}_{n+\frac{1}{2}} \Delta \mathbf{u}^{\text{def}} = \hat{\mathbf{x}}_{n+1} - \hat{\mathbf{x}}_n. \quad (17.83)$$

Once the strain increment is obtained by [equation \(17.72\)](#), the stress increment, also referred to the mid-point Configuration, can be calculated with the radial return algorithm. The total strain and stress can then be updated as

$$\hat{\boldsymbol{\varepsilon}}_{n+1} = \hat{\boldsymbol{\varepsilon}}_n + \Delta \hat{\boldsymbol{\varepsilon}}, \quad (17.84)$$

$$\hat{\boldsymbol{\sigma}}_{n+1} = \hat{\boldsymbol{\sigma}}_n + \Delta \hat{\boldsymbol{\sigma}}. \quad (17.85)$$

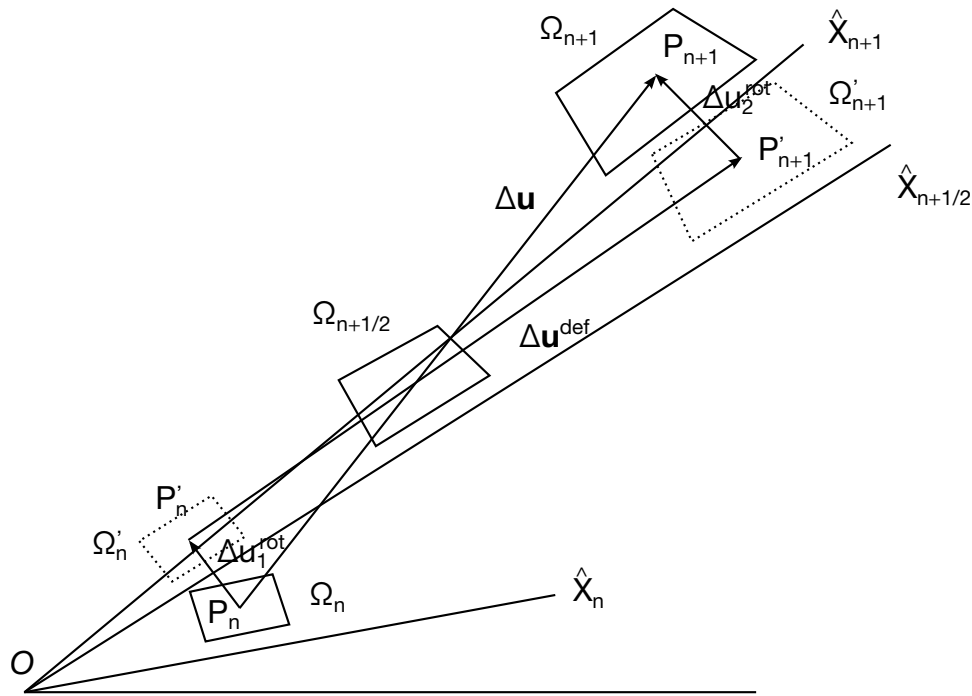


Figure 17.3. Separation of the displacement increment

Note that the resultant stress and strain tensors are both referred to the current configuration and defined in the current co-rotational coordinate system. By using the tensor transformation rule we can have the strain and stress components in the global coordinate system.

Tangent Stiffness Matrix and Nodal Force Vectors

From the Hu-Washizu variational principle, at both v th and $(v + 1)$ th iteration, we have

$$\int_{\hat{\Omega}^v} \delta \hat{\varepsilon}_{ij}^v \hat{\sigma}_{ij}^v dV = \delta \hat{\pi}_{\text{ext}}^v, \quad (17.86)$$

$$\int_{\hat{\Omega}^{v+1}} \delta \hat{\varepsilon}_{ij}^{v+1} \hat{\sigma}_{ij}^{v+1} dV = \delta \hat{\pi}_{\text{ext}}^{v+1}, \quad (17.87)$$

where $\delta \hat{\pi}_{\text{ext}}$ is the virtual work done by the external forces. Note that both equations are written in the co-rotational coordinate system defined in the v th iterative configuration given by x_{n+1}^v . The variables in this section are within the time step $[t_n, t_{n+\frac{1}{2}}]$ and superscripts indicate the number of iterations.

Assuming that all external forces are deformation-independent, linearization of Equation (17.87) gives [Liu, 1992]

$$\int_{\Omega^v} \delta \hat{u}_{i,j}^v \hat{C}_{ijkl}^v \Delta \hat{u}_{k,l} dV + \int_{\Omega^v} \delta \hat{u}_{i,j}^v \hat{T}_{ijkl}^v \Delta \hat{u}_{k,l} dV = \delta \hat{\pi}_{\text{ext}}^{v+1} - \delta \hat{\pi}_{\text{ext}}^v, \quad (17.88)$$

where the Green-Naghdi rate of Cauchy stress tensor is used, i.e.,

$$\hat{T}_{ijkl}^v = \delta_{ik} \hat{\sigma}_{jl}^v. \quad (17.89)$$

The first term on the left hand side of (17.88) denotes the material response since it is due to pure deformation or stretching; the second term is an initial stress part resulting from finite deformation effect.

Taking account of the residual of the previous iteration, Equation (17.87) can be approximated as

$$\int_{\Omega^v} \delta \hat{u}_{i,j}^v (\hat{C}_{ijkl}^v + \hat{T}_{ijkl}^v) \Delta \hat{u}_{k,l} dV = \delta \hat{\pi}_{\text{ext}}^{v+1} - \int_{\Omega^v} \delta \hat{\varepsilon}_{ij}^v \hat{\sigma}_{ij}^v dV. \quad (17.90)$$

If the strain and stress vectors are defined as

$$\boldsymbol{\varepsilon}^T = [\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad 2\varepsilon_{xy} \quad 2\varepsilon_{yz} \quad 2\varepsilon_{zx} \quad 2\omega_{xy} \quad 2\omega_{yz} \quad 2\omega_{zx}], \quad (17.91)$$

$$\boldsymbol{\sigma}^T = [\sigma_x \quad \sigma_y \quad \sigma_z \quad \sigma_{xy} \quad \sigma_{yz} \quad \sigma_{zx}], \quad (17.92)$$

We can rewrite equation (17.90) as

$$\int_{\Omega^v} \delta \hat{\varepsilon}_i^v (\hat{C}_{ij}^v + \hat{T}_{ij}^v) \delta \hat{\varepsilon}_j dV = \delta \hat{\pi}_{\text{ext}}^{v+1} - \int_{\Omega^v} \delta \hat{\varepsilon}_i^v \hat{\sigma}_j^v dV, \quad (17.93)$$

where \hat{C}_{ij}^v is the consistent tangent modulus tensor corresponding to pure deformation (see Section 3.2.3) but expanded to a 9 by 9 matrix; \hat{T}_{ij}^v is the geometric stiffness matrix which is given as follows [Liu 1992]:

$$\begin{aligned}
\mathbf{T} = & \begin{bmatrix}
\sigma_1 & 0 & 0 & \frac{\sigma_4}{2} & 0 & \frac{\sigma_6}{2} & \frac{\sigma_4}{2} & 0 & -\frac{\sigma_6}{2} \\
& \sigma_2 & 0 & \frac{\sigma_4}{2} & \frac{\sigma_5}{2} & 0 & -\frac{\sigma_4}{2} & \frac{\sigma_5}{2} & 0 \\
& & \sigma_3 & 0 & \frac{\sigma_5}{2} & \frac{\sigma_6}{2} & 0 & -\frac{\sigma_5}{2} & \frac{\sigma_6}{2} \\
& & & \frac{\sigma_1 + \sigma_2}{4} & \frac{\sigma_6}{4} & \frac{\sigma_5}{4} & \frac{\sigma_2 - \sigma_1}{4} & \frac{\sigma_6}{4} & -\frac{\sigma_5}{4} \\
& & & & \frac{\sigma_2 + \sigma_3}{4} & \frac{\sigma_4}{4} & -\frac{\sigma_6}{4} & \frac{\sigma_3 - \sigma_2}{4} & \frac{\sigma_4}{4} \\
& & & & & \frac{\sigma_1 + \sigma_3}{4} & \frac{\sigma_5}{4} & -\frac{\sigma_4}{4} & \frac{\sigma_1 - \sigma_3}{4} \\
& & \text{symm.} & & & & \frac{\sigma_1 + \sigma_2}{4} & -\frac{\sigma_6}{4} & -\frac{\sigma_5}{4} \\
& & & & & & & \frac{\sigma_2 + \sigma_3}{4} & -\frac{\sigma_4}{4} \\
& & & & & & & & \frac{\sigma_3 + \sigma_1}{4}
\end{bmatrix}. \quad (17.94)
\end{aligned}$$

By interpolation

$$\Delta \mathbf{u} = \mathbf{N} \Delta \mathbf{d}, \quad \delta \mathbf{u} = \mathbf{N} \delta \mathbf{d}; \quad (17.95)$$

$$\Delta \boldsymbol{\varepsilon} = \bar{\mathbf{B}} \Delta \mathbf{d}, \quad \delta \boldsymbol{\varepsilon} = \bar{\mathbf{B}} \delta \mathbf{d}, \quad (17.96)$$

where \mathbf{N} and $\bar{\mathbf{B}}$ are, respectively, the shape functions and strain operators defined in Section 2. This leads to a set of equations

$$\hat{\mathbf{K}}^v \Delta \hat{\mathbf{d}} = \hat{\mathbf{r}}^{v+1} = \hat{\mathbf{f}}_{\text{ext}}^{v+1} - \hat{\mathbf{f}}_{\text{int}}^v, \quad (17.97)$$

where the tangent stiffness matrix, $\hat{\mathbf{K}}^v$, and the internal nodal force vector, $\hat{\mathbf{f}}_{\text{int}}^v$, are

$$\hat{\mathbf{K}}^v = \int_{\hat{\Omega}^v} \hat{\mathbf{B}}^T (\hat{\mathbf{C}}^v + \hat{\mathbf{T}}^v) \hat{\mathbf{B}} dV, \quad (17.98)$$

$$\hat{\mathbf{f}}_{\text{int}}^v = \int_{\hat{\Omega}^v} \hat{\mathbf{B}}^T \hat{\boldsymbol{\sigma}}^v dV. \quad (17.99)$$

The tangent stiffness and nodal force are transformed into the global coordinate system tensorially as

$$\mathbf{K}^v = \mathbf{R}^{vT} \hat{\mathbf{K}}^v \mathbf{R}^v, \quad (17.100)$$

$$\mathbf{r}^{v+1} = \mathbf{R}^{vT} \hat{\mathbf{f}}_{\text{int}}^v, \quad (17.101)$$

where \mathbf{R}^v is the transformation matrix of the co-rotational system defined by \mathbf{x}_{n+1}^v . Finally, we get a set of linear algebraic equations

$$\mathbf{K}^v \Delta \mathbf{d}^{v+1} = \mathbf{r}^{v+1}. \quad (17.102)$$

17.4 Numerical Examples

To investigate the performance of the element introduced in this paper, a variety of problems including linear elastic and nonlinear elastic-plastic/large deformation problems are studied. Since the element is developed to avoid locking, the applicability to problems of thin structures is studied by solving the standard test problems including pinched cylinder and Scordelis-Lo roof, which are proposed by MacNeal, 1985 and Belytschko, 1984b. Also a sheet metal forming problem is solved to test and demonstrate the effectiveness and efficiency of this element.

17.4.1 Timoshenko Cantilever Beam

The first problem is a linear, elastic cantilever beam with a load at its end as shown in Fig. 17.4, where M and P at the left end of the cantilever are reactions at the support. The analytical solution from Timoshenko, 1970 is

$$u_x(x, y) = \frac{-Py}{6\bar{E}I} \left[(6L - 3x)x + (2 + \bar{\nu}) \left(y^2 - \frac{1}{4}D^2 \right) \right], \quad (17.103)$$

$$u_y(x, y) = \frac{P}{6\bar{E}I} \left[3\bar{\nu}y^2(L - x) + \frac{1}{4}(4 + 5\bar{\nu})D^2x + (3L - x)x^2 \right], \quad (17.104)$$

where

$$I = \frac{1}{12}D^3, \quad (17.105)$$

$$\bar{E} = \begin{cases} E, \\ E/(1 - \nu^2), \end{cases}, \quad \bar{\nu} = \begin{cases} \nu & \text{for plane stress} \\ \frac{\nu}{1 - \nu} & \text{for plane strain} \end{cases} \quad (17.106)$$

The displacements at the support end, $x = 0$, $-\frac{1}{2}D \leq y \leq \frac{1}{2}D$ are nonzero except at the top, bottom and midline (as shown in Fig. 13.5). Reaction forces are applied at the support based on the stresses corresponding to the displacement field at $x = 0$, which are

$$\sigma_{xx} = -\frac{Py}{I}(L - x), \quad \sigma_{yy} = 0, \quad \sigma_{xy} = \frac{P}{2I} \left(\frac{1}{4}D^2 - y^2 \right). \quad (17.107)$$

The distribution of the applied load to the nodes at $x = L$ is also obtained from the closed-form stress fields.

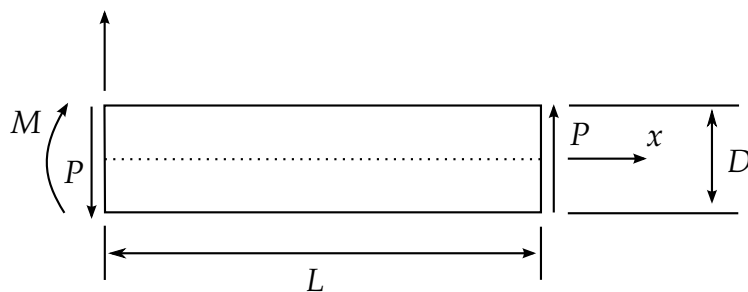
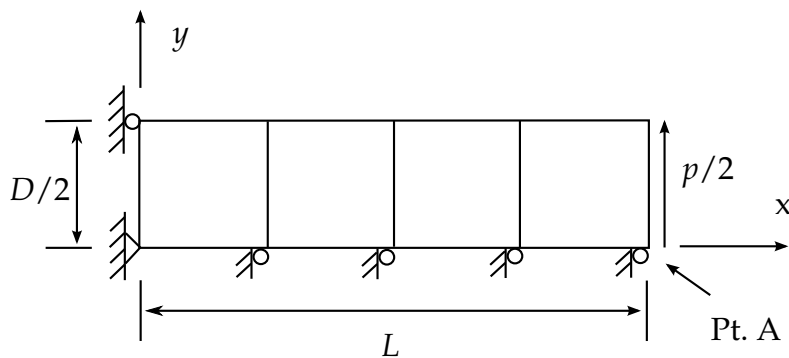
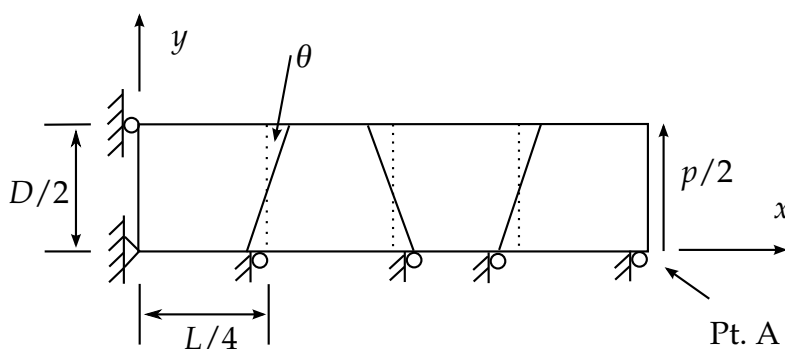


Figure 17.4. Timoshenko cantilever beam.



(a) Regular mesh



(b) Skewed mesh

Figure 17.5. Top half of anti-symmetric beam mesh

The parameters for the cantilever beam are: $L = 1.0$, $D = 0.02$, $P = 2.0$, $E = 1 \times 10^7$; and two values of Poisson's ratio: (1) $\nu = 0.25$, (2) $\nu = 0.4999$.

Since the problem is anti-symmetric, only the top half of the beam is modeled. Plane strain conditions are assumed in the z-direction and only one layer of elements is used in this direction. Both regular mesh and skewed mesh are tested for this problem.

Normalized vertical displacements at point A for each case are given in Table 13.1. Tables 13.1a and 13.1b show the normalized displacement at point A for the regular

mesh. There is no shear or volumetric locking for this element. For the skewed mesh, with the skewed angle increased, we need more elements to get more accurate solution (Table 13.1c).

(a) $\nu = 0.25$, regular mesh

Analytical solution $w_A = 9.3777 \times 10^{-2}$

Mesh	$4 \times 1 \times 1$	$8 \times 1 \times 1$	$8 \times 2 \times 1$
HEXDS	1.132	1.142	1.029

(b) $\nu = 0.4999$, regular mesh

Analytical solution $w_A = 7.5044 \times 10^{-2}$

Mesh	$4 \times 1 \times 1$	$8 \times 1 \times 1$	$8 \times 2 \times 1$
HEXDS	1.182	1.197	1.039

(c) $\nu = 0.25$, skewed mesh

	1°	5°	10°
$4 \times 1 \times 1$	1.078	0.580	0.317
$8 \times 1 \times 1$	1.136	0.996	0.737
$16 \times 1 \times 1$	1.142	1.090	.955

Table 13.1. Normalized displacement at point A of cantilever beam.

17.4.2 Pinched Cylinder

Figure 17.6 shows a pinched cylinder subjected to a pair of concentrated loads. Two cases are studied in this example. In the first case, both ends of the cylinder are assumed to be free. In the second case, both ends of the cylinder are covered with rigid diaphragms so that only the displacement in the axial direction is allowed at the ends. The parameters for the first case (without diaphragms) are

$$E = 1.05 \times 10^6, v = 0.3125, L = 10.35, R = 1.0, t = 0.094, P = 100.0; \quad (17.108)$$

while for the second case (with diaphragms), the parameters are set to be

$$E = 3 \times 10^6, v = 0.3, L = 600.0, R = 300.0, t = 3.0, P = 1.0. \quad (17.109)$$

Due to symmetry only one octant of the cylinder is modeled. The computed displacements at the loading point are compared to the analytic solutions in Table 13.2. HEXDS element works well in both cases, indicating that this element can avoid not only shear locking but also membrane locking; this is not unexpected since membrane locking occurs primarily in curved elements [Stolarski, 1983].

17.4.3 Scordelis-Lo Roof

Scordelis-Lo roof subjected to its own weight is shown in Figure 17.7. Both ends of the roof are assumed to be covered with rigid diaphragms. The parameters are selected to be: $E = 4.32 \times 10^8$, $v = 0.0$, $L = 50.0$, $R = 25.0$, $t = 0.25$, $\theta = 40^\circ$, and the gravity is 360.0 per volume.

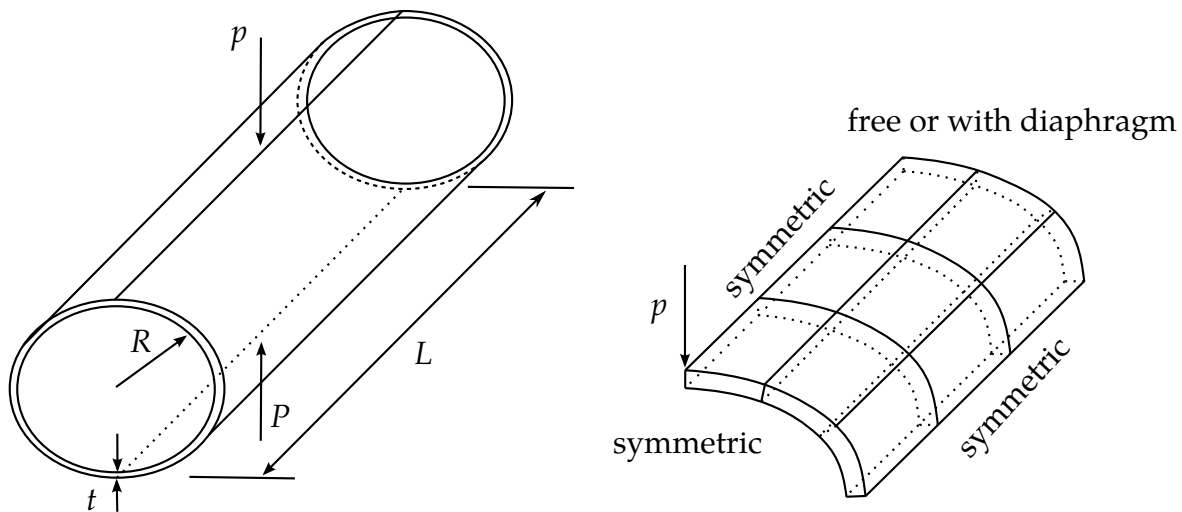
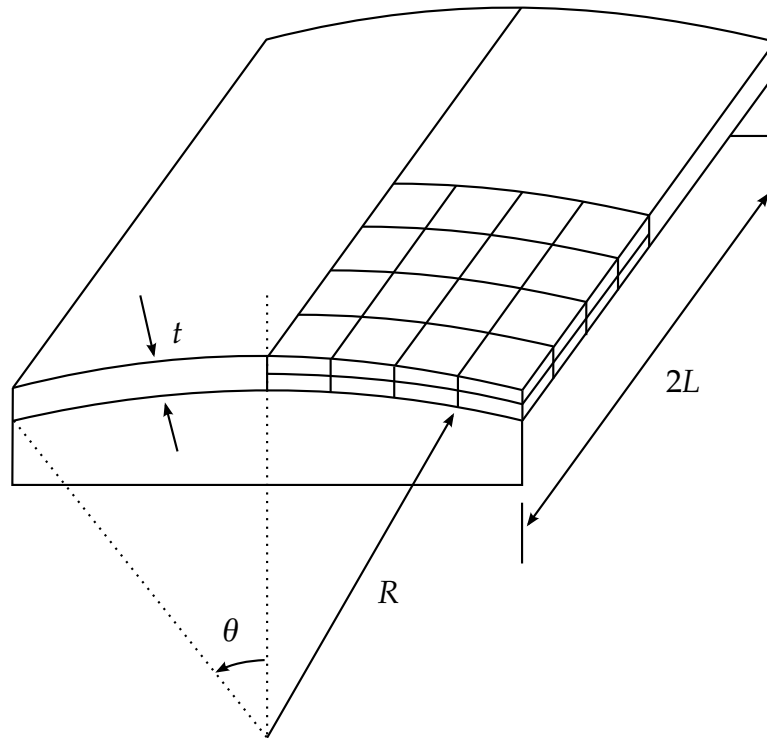


Figure 17.6. Pinched cylinder and the element model

**Figure 17.7.** Scordelis-Lo roof under self-weight

(a) First case without diaphragms

Analytical solution $w_{\max} = 0.1137$

Mesh	$10 \times 10 \times 2$	$16 \times 16 \times 4$	$20 \times 20 \times 4$
HEXDS	1.106	1.054	1.067

(b) Second case with diaphragms

Analytical solution $w_{\max} = 1.8248 \times 10^{-5}$

Mesh	$10 \times 10 \times 2$	$16 \times 16 \times 4$	$20 \times 20 \times 4$
HEXDS	0.801	0.945	.978

Table 13.2. Normalized displacement at loading point of pinched cylinder

Due to symmetry only one quarter of the roof is modeled. The computed displacement at the midpoint of the edge is compared to the analytic solution in Table 13.3. In this example the HEXDS element can get good result with 100×2 elements.

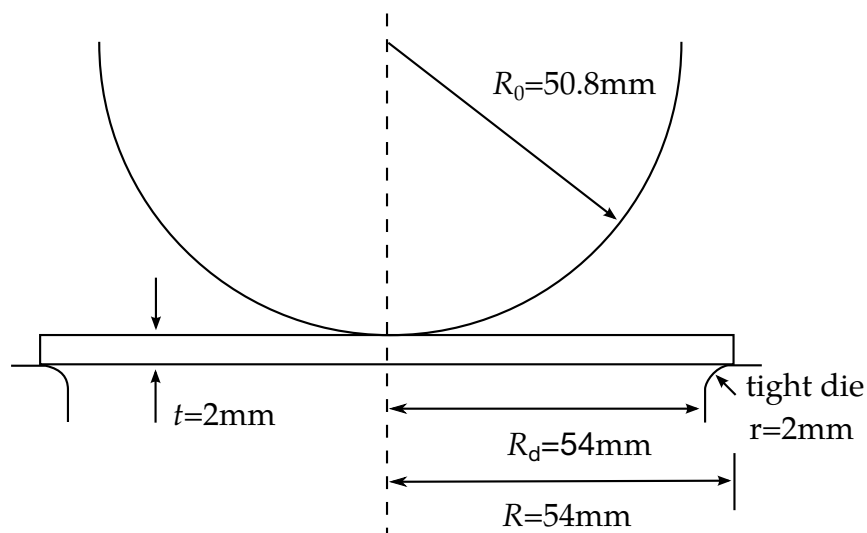


Figure 17.8. Circular sheet stretched with a tight die

Analytical solution $w_{\max} = 0.3024$

Mesh	$8 \times 8 \times 1$	$16 \times 16 \times 1$	$32 \times 32 \times 1$	$10 \times 10 \times 2$
HEXDS	1.157	1.137	1.132	1.045

17.4.4 Circular Sheet Stretched with a Tight Die

A circular sheet is stretched under a hemisphere punch and a tight die with a small corner radius (Fig. 17.8). The material is elastoplastic with nonlinear hardening rule. The elastic material constants are: $E = 206 \text{ GPa}$ and $\nu = 0.3$. In the plastic range, the uniaxial stress-strain curve is given by

$$\sigma = K\varepsilon^n, \quad (17.110)$$

where $K = 509.8 \text{ MPa}$, $n = 0.21$, σ is Cauchy stress and ε is natural strain (logarithmic strain). The initial yield stress is obtained to be $\sigma_0 = 103.405 \text{ MPa}$ and the tangent modulus at the initial yield point is $E_t = 0.4326 \times 10^5 \text{ MPa}$.

Because of the small corner radius of the die, the same difficulties as in the problem of sheet stretch under the rigid cylinders lead the shell elements to failure in this problem. Three-dimensional solid elements are needed and fine meshes should be put in the areas near the center and the edge of the sheet.

One quarter of the sheet is modeled with 1400×2 HEXDS elements due to the double symmetries. The mesh is shown in Fig. 17.9. Two layers of elements are used in the thickness. Around the center and near the circular edge of the sheet, fine mesh is used. The nodes on the edge are fixed in x- and y-directions and the bottom nodes on the edge are prescribed in three directions. No friction is considered in this simulation. For

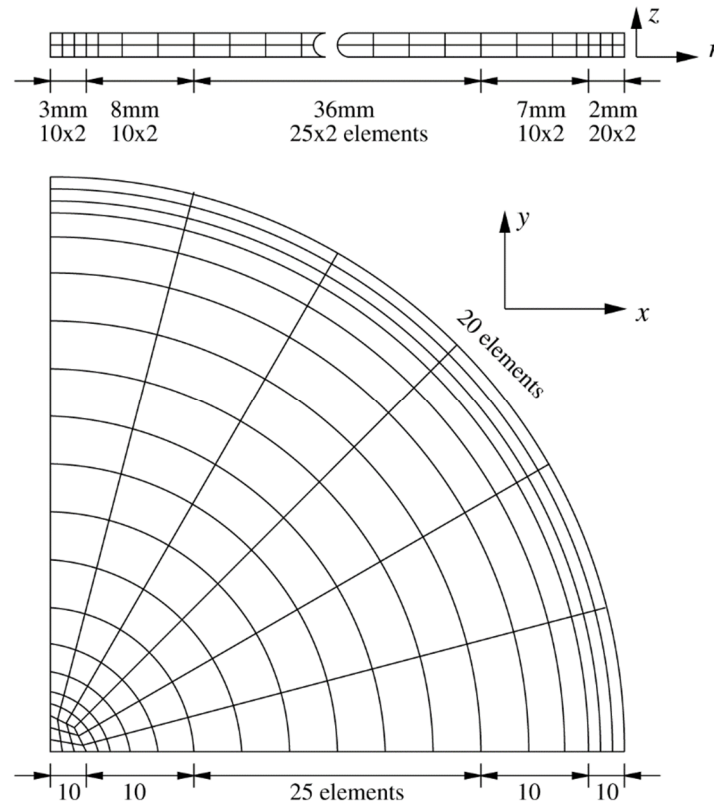


Figure 17.9. Mesh for circular sheet stretching

comparison, the axisymmetric four-node element with reduced integration (CAX4R) is also used and the mesh for this element is the same as shown in the top of Figure 13.9.

The results presented here are after the punch has traveled down 50 mm. The profile of the circular sheet is shown in [Figure 17.10](#) where we can see that the sheet under the punch experiences most of the stretching and the thickness of the sheet above the die

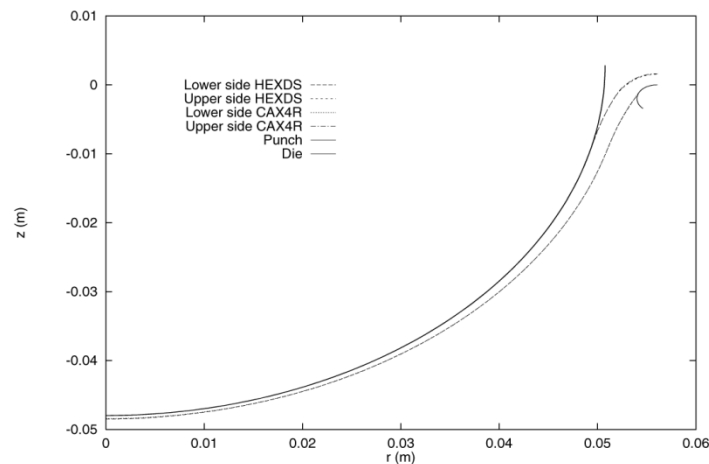


Figure 17.10. Deformed shape of a circular sheet with punch travel 50 mm

changes a lot. The deformation between the punch and the die is small. However, the

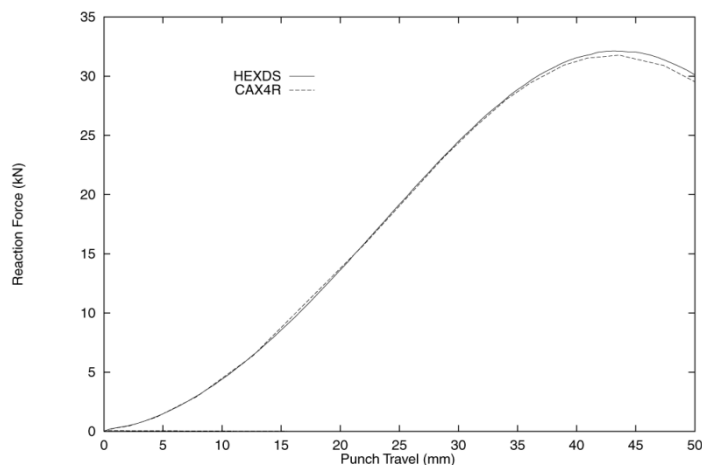
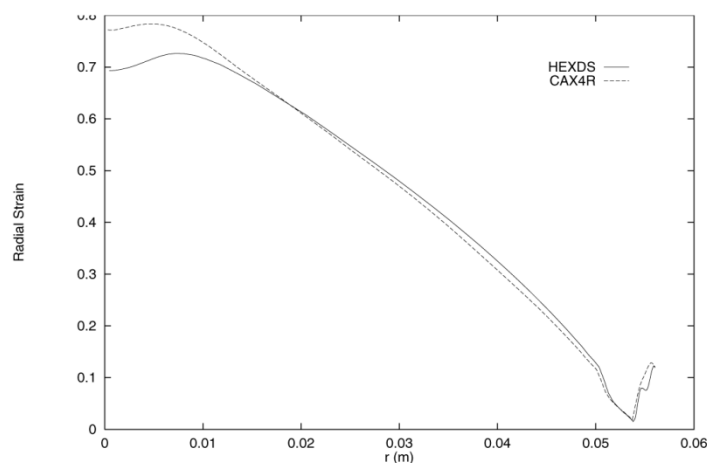
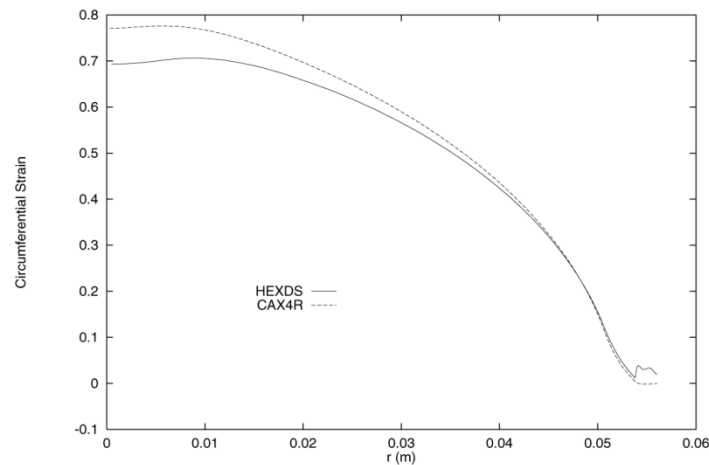


Figure 17.11. Reaction force vs. punch travel for the circular sheet

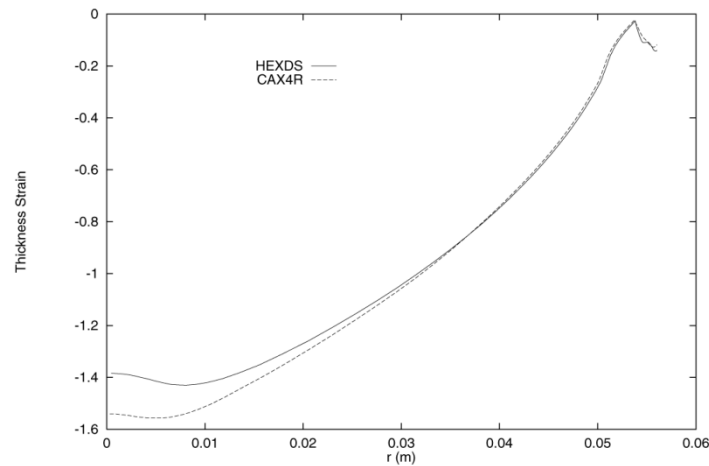
sheet thickness obtained by the CAX4R element is less than that by the HEXDS element and there is slight difference above the die. These observations can be verified by the strain distributions in the sheet along the radial direction (Figure 13.12). The direction of the radial strain is the tangent of the mid-surface of the element in the rz plane and the thickness strain is in the direction perpendicular to the mid-surface of the element. The unit vector of the circumferential strain is defined as the cross-product of the directional cosine vectors of the radial strain and the thickness strain. We can see that the CAX4R element yields larger strain components in the area under the punch than the HEXDS element. The main difference of the strain distributions in the region above the die is that the CAX4R element gives zero circumferential strain in this area but the HEXDS element yields non-zero strain. The value of the reaction force shown in the Figure 13.11 is only one quarter of the total punch reaction force since only one quarter of the sheet is modeled. From this figure we can see that the sheet begins softening after the punch travels down about 45 mm, indicating that the sheet may have necking though this cannot be seen clearly from Figure 17.10.



(a) Radial strain distribution



(b) Circumferential strain distribution



(c) Thickness strain distribution

Figure 13.12. Strain distributions in circular sheet with punch travel 50 mm

17.5 Conclusions

A new eight-node hexahedral element is implemented for the large deformation elastic-plastic analysis. Formulated in the co-rotational coordinate system, this element is shown to be effective and efficient and can achieve fast convergence in solving a wide variety of nonlinear problems.

By using a co-rotational system which rotates with the element, the locking phenomena can be suppressed by omitting certain terms in the generalized strain operators. In addition, the integration of the constitutive equation in the co-rotational system takes

the same simple form as small deformation theory since the stress and strain tensors defined in this co-rotational system are objective.

Radial return algorithm is used to integrate the rate-independent elastoplastic constitutive equation. The tangent stiffness matrix consistently derived from this integration scheme is crucial to preserve the second order convergence rate of the Newton's iteration method for the nonlinear static analyses.

Test problems studied in this paper demonstrate that the element is suitable to continuum and structural numerical simulations. In metal sheet forming analysis, this element has advantages over shell elements for certain problems where through the thickness deformation and strains are significant.

18

Truss Element

One of the simplest elements is the pin-jointed truss element shown in [Figure 18.1](#). This element has three degrees of freedom at each node and carries an axial force. The displacements and velocities measured in the local system are interpolated along the axis according to

$$u = u_1 + \frac{x}{L}(u_2 - u_1), \quad (18.1)$$

$$\dot{u} = \dot{u}_1 + \frac{x}{L}(\dot{u}_2 - \dot{u}_1), \quad (18.2)$$

where at $x = 0$, $u = u_1$ and at $x = L$, $u = u_2$. Incremental strains are found from

$$\Delta \varepsilon = \frac{(\dot{u}_2 - \dot{u}_1)}{L} \Delta t \quad (18.3)$$

and are computed in LS-DYNA using

$$\Delta \varepsilon^{n+1/2} = \frac{2 \left(\dot{u}_2^{n+1/2} - \dot{u}_1^{n+1/2} \right)}{L^n + L^{n+1}} \Delta t^{n+1/2} \quad (18.4)$$

The normal force N is then incrementally updated using a tangent modulus E^t according to

$$N^{n+1} = N^n A E^t + \Delta \varepsilon^{n+1/2} \quad (18.5)$$

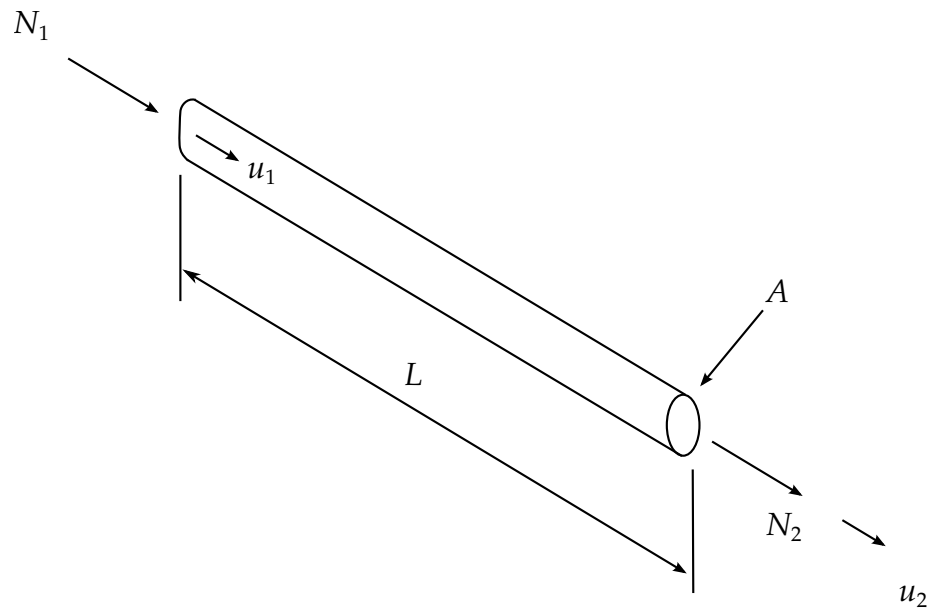


Figure 18.1. Truss element.

Two constitutive models are implemented for the truss element: elastic and elastic-plastic with kinematic hardening.

19

Membrane Element

The Belytschko-Lin-Tsay shell element {Belytschko and Tsay [1981], Belytschko et al., [1984a]} is the basis for this very efficient membrane element. In this section we briefly outline the theory employed which, like the shell on which it is based, uses a combined co-rotational and velocity-strain formulation. The efficiency of the element is obtained from the mathematical simplifications that result from these two kinematical assumptions. The co-rotational portion of the formulation avoids the complexities of nonlinear mechanics by embedding a coordinate system in the element. The choice of velocity strain or rate of deformation in the formulation facilitates the constitutive evaluation, since the conjugate stress is the more familiar Cauchy stress.

In membrane elements the rotational degrees of freedom at the nodal points may be constrained, so that only the translational degrees-of-freedom contribute to the straining of the membrane. A triangular membrane element may be obtained by collapsing adjacent nodes of the quadrilateral.

19.1 Co-rotational Coordinates

The mid-surface of the quadrilateral membrane element is defined by the location of the element's four corner nodes. An embedded element coordinate system (Figure 7.1) that deforms with the element is defined in terms of these nodal coordinates. The co-rotational coordinate system follows the development in Section 7, Equations (7.1)—(7.3).

19.2 Velocity-Strain Displacement Relations

The co-rotational components of the velocity strain (rate of deformation) are given by:

$$\hat{d}_{ij} = \frac{1}{2} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right), \quad (19.1)$$

The above velocity-strain relations are evaluated only at the center of the shell. Standard bilinear nodal interpolation is used to define the mid-surface velocity, angular velocity, and the element's coordinates (isoparametric representation). These interpolation relations are given by

$$v^m = N_I(\xi, \eta) v_I, \quad (19.2)$$

$$x^m = N_I(\xi, \eta) x_I, \quad (19.3)$$

where the subscript I is summed over all the element's nodes and the nodal velocities are obtained by differentiating the nodal coordinates with respect to time, i.e., $v_I = \dot{x}_I$. The bilinear shape functions are defined in Equations (7.10).

The velocity strains at the center of the element, i.e., at $\xi = 0$, and $\eta = 0$, are obtained as in Section 7 giving:

$$\hat{d}_x = B_{1I} \hat{v}_{xI}, \quad (19.4)$$

$$\hat{d}_y = B_{2I} \hat{v}_{yI}, \quad (19.5)$$

$$2\hat{d}_{xy} = B_{2I} \hat{v}_{xI} + B_{1I} \hat{v}_{yI}, \quad (19.6)$$

where

$$B_{1I} = \frac{\partial N_I}{\partial \hat{x}}, \quad (19.7)$$

$$B_{2I} = \frac{\partial N_I}{\partial \hat{y}}. \quad (19.8)$$

19.3 Stress Resultants and Nodal Forces

After suitable constitutive evaluations using the above velocity strains, the resulting stresses are multiplied by the thickness of the membrane, h , to obtain local resultant forces. Therefore,

$$\hat{f}_{\alpha\beta}^R = h\hat{\sigma}_{\alpha\beta}, \quad (19.9)$$

where the superscript R indicates a resultant force and the Greek subscripts emphasize the limited range of the indices for plane stress plasticity.

The above element centered force resultants are related to the local nodal forces by

$$\hat{f}_{xI} = A(B_{1I}\hat{f}_{xx}^R + B_{2I}\hat{f}_{xy}^R), \quad (19.10)$$

$$\hat{f}_{yI} = A(B_{2I}\hat{f}_{yy}^R + B_{1I}\hat{f}_{xy}^R), \quad (19.11)$$

where A is the area of the element.

The above local nodal forces are then transformed to the global coordinate system using the transformation relations given in Equation (7.5a).

19.4 Membrane Hourglass Control

Hourglass deformations need to be resisted for the membrane element. The hourglass control for this element is discussed in Section 7.4.

20

Discrete Elements and Masses

The discrete elements and masses in LS-DYNA provide a capability for modeling simple spring-mass systems as well as the response of more complicated mechanisms. Occasionally, the response of complicated mechanisms or materials needs to be included in LS-DYNA models, e.g., energy absorbers used in passenger vehicle bumpers. These mechanisms are often experimentally characterized in terms of force-displacement curves. LS-DYNA provides a selection of discrete elements that can be used individually or in combination to model complex force-displacement relations.

The discrete elements are assumed to be massless. However, to solve the equations of motion at unconstrained discrete element nodes or nodes joining multiple discrete elements, nodal masses must be specified at these nodes. LS-DYNA provides a direct method for specifying these nodal masses in the model input.

All of the discrete elements are two-node elements, i.e., three-dimensional springs or trusses. A discrete element may be attached to any of the other LS-DYNA continuum, structural, or rigid body element. The force update for the discrete elements may be written as

$$\hat{\mathbf{f}}^{i+1} = \hat{\mathbf{f}}^i + \Delta \hat{\mathbf{f}}, \quad (20.1)$$

where the superscript $i + 1$ indicates the time increment and the superposed caret ($\hat{\cdot}$) indicates the force in the local element coordinates, i.e., along the axis of the element. In the default case, i.e., no orientation vector is used; the global components of the discrete element force are obtained by using the element's direction cosines:

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \frac{\hat{f}}{l} \begin{Bmatrix} \Delta l_x \\ \Delta l_y \\ \Delta l_z \end{Bmatrix} = \hat{f} \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = \hat{f} \mathbf{\tilde{n}}, \quad (20.2)$$

where

$$\Delta \mathbf{l} = \begin{Bmatrix} \Delta l_x \\ \Delta l_y \\ \Delta l_z \end{Bmatrix} = \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{Bmatrix}. \quad (20.3)$$

l is the length

$$l = \sqrt{\Delta l_x^2 + \Delta l_y^2 + \Delta l_z^2}, \quad (20.4)$$

and (x_i, y_i, z_i) are the global coordinates of the nodes of the spring element. The forces in Equation (20.2) are added to the first node and subtracted from the second node.

For a node tied to ground we use the same approach but for the (x_2, y_2, z_2) coordinates in Equation (20.2) the initial coordinates of node 1, i.e., (x_0, y_0, z_0) are used instead; therefore,

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \frac{\hat{f}}{l} \begin{Bmatrix} x_0 - x_1 \\ y_0 - y_1 \\ z_0 - z_1 \end{Bmatrix} = \hat{f} \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix}. \quad (20.5)$$

The increment in the element force is determined from the user specified force-displacement relation. Currently, nine types of force-displacement/velocity relationships may be specified:

1. linear elastic;
2. linear viscous;
3. nonlinear elastic;
4. nonlinear viscous;
5. elasto-plastic with isotropic hardening;
6. general nonlinear;
7. linear viscoelastic.
8. inelastic tension and compression only.
9. muscle model.

The force-displacement relations for these models are discussed in the following later.

20.1 Orientation Vectors

An orientation vector,

$$\mathbf{m} = \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix}, \quad (20.6)$$

can be defined to control the direction the spring acts. If orientation vectors are used, it is strongly recommended that the nodes of the discrete element be coincident and remain

approximately so throughout the calculation. If the spring or damper is of finite length, rotational constraints will appear in the model that can substantially affect the results. If finite length springs are needed with directional vectors, then the discrete beam elements, the type 6 beam, should be used with the coordinate system flagged for the finite length case.

We will first consider the portion of the displacement that lies in the direction of the vector. The displacement of the spring is updated based on the change of length given by

$$\Delta I = I - I_0, \quad (20.7)$$

where I_0 is the initial length in the direction of the vector and I is the current length given for a node to node spring by

$$I = m_1(x_2 - x_1) + m_2(y_2 - y_1) + m_3(z_2 - z_1), \quad (20.8)$$

and for a node to ground spring by

$$I = m_1(x_0 - x_1) + m_2(y_0 - y_1) + m_3(z_0 - z_1), \quad (20.9)$$

The latter case is not intuitively obvious and can affect the sign of the force in unexpected ways if the user is not familiar with the relevant equations. The nodal forces are then given by

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \hat{f} \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix}. \quad (20.10)$$

The orientation vector can be either permanently fixed in space as defined in the input or acting in a direction determined by two moving nodes which must not be coincident but may be independent of the nodes of the spring. In the latter case, we recompute the direction every cycle according to:

$$\begin{Bmatrix} m_1 \\ m_2 \\ m_3 \end{Bmatrix} = \frac{1}{l^n} \begin{Bmatrix} x_2^n - x_1^n \\ y_2^n - y_1^n \\ z_2^n - z_1^n \end{Bmatrix}. \quad (20.11)$$

In Equation (20.9) the superscript, n , refers to the orientation nodes.

For the case where we consider motion in the plane perpendicular to the orientation vector we consider only the displacements in the plane, $\underline{\Delta}^p$, given by,

$$\Delta \mathbf{l}^p = \Delta \mathbf{l} - \mathbf{m}(\mathbf{m} \cdot \Delta \mathbf{l}). \quad (20.12)$$

We update the displacement of the spring based on the change of length in the plane given by

$$\Delta l^p = l^p - l_0^p, \quad (20.13)$$

where l_0^p is the initial length in the direction of the vector and l is the current length given for a node to node spring by

$$l^p = m_1^p(x_2 - x_1) + m_2^p(y_2 - y_1) + m_3^p(z_2 - z_1), \quad (20.14)$$

and for a node to ground spring by

$$l^p = m_1^p(x_0 - x_1) + m_2^p(y_0 - y_1) + m_3^p(z_0 - z_1), \quad (20.15)$$

where

$$\begin{Bmatrix} m_1^p \\ m_2^p \\ m_3^p \end{Bmatrix} = \frac{1}{\sqrt{\Delta l_x^{p^2} + \Delta l_y^{p^2} + \Delta l_z^{p^2}}} \begin{Bmatrix} \Delta l_x^p \\ \Delta l_y^p \\ \Delta l_z^p \end{Bmatrix}. \quad (20.16)$$

After computing the displacements, the nodal forces are then given by

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \hat{f} \begin{Bmatrix} m_1^p \\ m_2^p \\ m_3^p \end{Bmatrix}. \quad (20.17)$$

20.2 Dynamic Magnification “Strain Rate” Effects

To account for “strain rate” effects, we have a simple method of scaling the forces based on the relative velocities that applies to all springs. The forces computed from the spring elements are assumed to be the static values and are scaled by an amplification factor to obtain the dynamic value:

$$F_{\text{dynamic}} = \left(1. + k_d \frac{V}{V_0}\right) F_{\text{static}}, \quad (20.18)$$

where

k_d = is a user defined input value
 V = absolute relative velocity
 V_0 = dynamic test velocity

For example, if it is known that a component shows a dynamic crush force at 15 m/s equal to 2.5 times the static crush force, use $k_d = 1.5$ and $V_0 = 15$.

20.3 Deflection Limits in Tension and Compression

The deflection limit in compression and tension is restricted in its application to no more than one spring per node subject to this limit, and to deformable bodies only. For example in the former case, if three spring are in series either the center spring or the two end springs may be subject to a limit but not all three. When the limiting deflection

is reached momentum conservation calculations are performed and a common acceleration is computed:

$$\hat{a}_{\text{common}} = \frac{\hat{f}_1 + \hat{f}_2}{m_1 + m_2}. \quad (20.19)$$

An error termination will occur if a rigid body node is used in a spring definition where compression is limited.

20.4 Linear Elastic or Linear Viscous

These discrete elements have the simplest force-displacement relations. The linear elastic discrete element has a force-displacement relation of the form

$$\hat{f} = K\Delta l, \quad (20.20)$$

where K is the element's stiffness and Δl is the change in length of the element. The linear viscous element has a similar force-velocity (rate of displacement) relation:

$$\hat{f} = C \frac{\Delta l}{\Delta t}. \quad (20.21)$$

where C is a viscous damping parameter and Δt is the time step increment.

20.5 Nonlinear Elastic or Nonlinear Viscous

These discrete elements use piecewise force-displacement or force-relative velocity relations. The nonlinear elastic discrete element has a tabulated force-displacement relation of the form

$$\hat{f} = K\Delta l, \quad (20.22)$$

where $K(\Delta l)$ is the tabulated force that depends on the total change in the length of the element ([Figure 20.1](#)) The nonlinear viscous element has a similar tabulated force-relative velocity relation:

$$\hat{f} = C \frac{\Delta l}{\Delta t}, \quad (20.23)$$

where $C(\frac{\Delta l}{\Delta t})$ is the viscous damping force that depends on the rate of change of the element's length. Nonlinear discrete elements unload along the loading paths.

If the spring element is initially of zero length and if no orientation vectors are used then only the tensile part of the stress strain curve needs to be defined. However, if

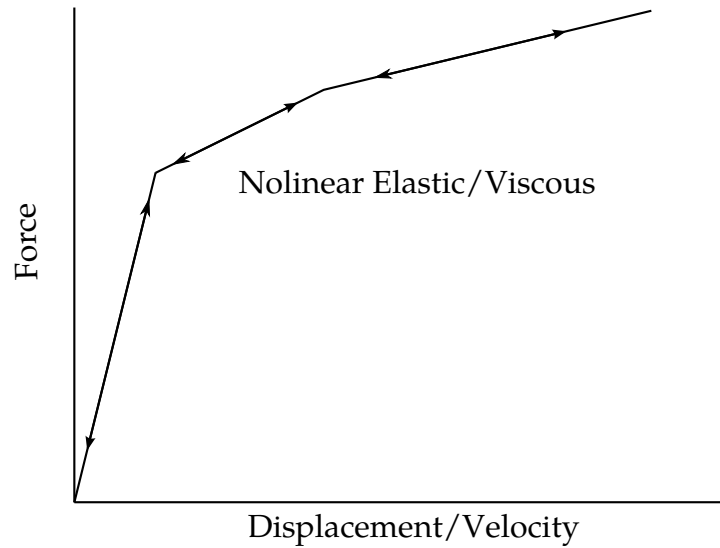


Figure 20.1. Piecewise linear force-displacement curve for nonlinear elastic discrete element.

the spring element is initially of finite length then the curve must be defined in both the positive and negative quadrants.

20.6 Elasto-Plastic with Isotropic Hardening

The elasto-plastic discrete element has a bilinear force-displacement relationship that is specified by the elastic stiffness, a tangent stiffness and a yield force (Figure 20.2). This discrete element uses the elastic stiffness model for unloading until the yield force is exceeded during unloading. The yield force is updated to track its maximum value which is equivalent to an isotropic hardening model. The force-displacement relation during loading may be written as

$$\hat{f} = F_y \left(1 - \frac{K_t}{K} \right) + K_t \Delta l, \quad (20.24)$$

where F_y is the yield force and K_t is the tangent stiffness.

20.7 General Nonlinear

The general nonlinear discrete element allows the user to specify independent and nonsymmetrical piecewise linear loading and unloading paths (Figure 20.3(a)).

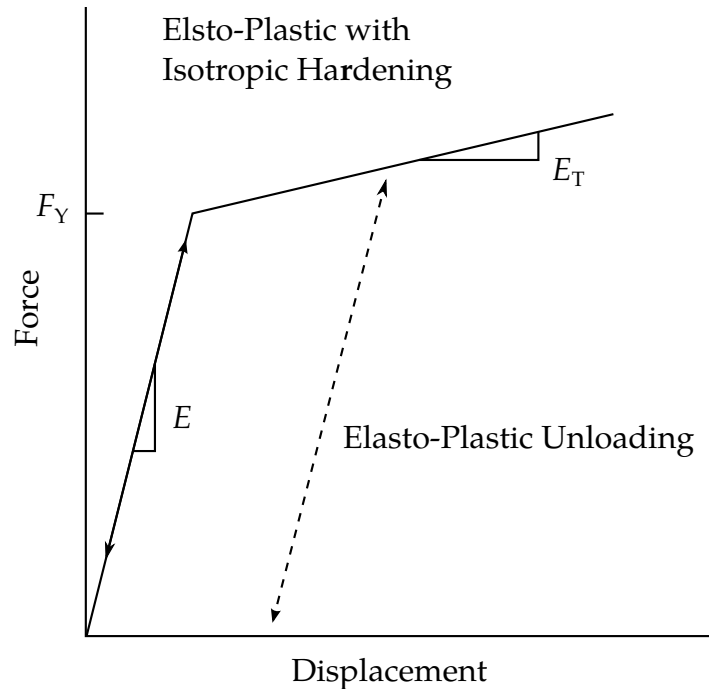


Figure 20.2. Loading and unloading force-displacement curves for elasto-plastic discrete element.

This element combines the features of the above-described nonlinear elastic and elasto-plastic discrete elements by allowing the user to specify independent initial yield forces in tension (FYT) and in compression (FYC). If the discrete element force remains between these initial yield values, the element unloads along the loading path (Figure 20.3(b)). This corresponds to the nonlinear elastic discrete element.

However, if the discrete element force exceeds either of these initial yield values, the specified unloading curve is used for subsequent unloading. Additionally, the initial loading and unloading curves are allowed to move in the force-displacement space by specifying a mixed hardening parameter β , where $\beta = 0$ corresponds to kinematic hardening (Figure 20.3(c)) and $\beta = 0 \rightarrow \beta = 1$ corresponds to isotropic hardening (Figure 20.3(d)).

20.8 Linear Visco-Elastic

The linear viscoelastic discrete element [Schwer, Cheva, and Hallquist 1991] allows the user to model discrete components that are characterized by force relaxation or displacement creep behavior. The element's variable stiffness is defined by three parameters and has the form

$$K(t) = K_{\infty} + (K_0 - K_{\infty})e^{-\beta t}, \quad (20.25)$$

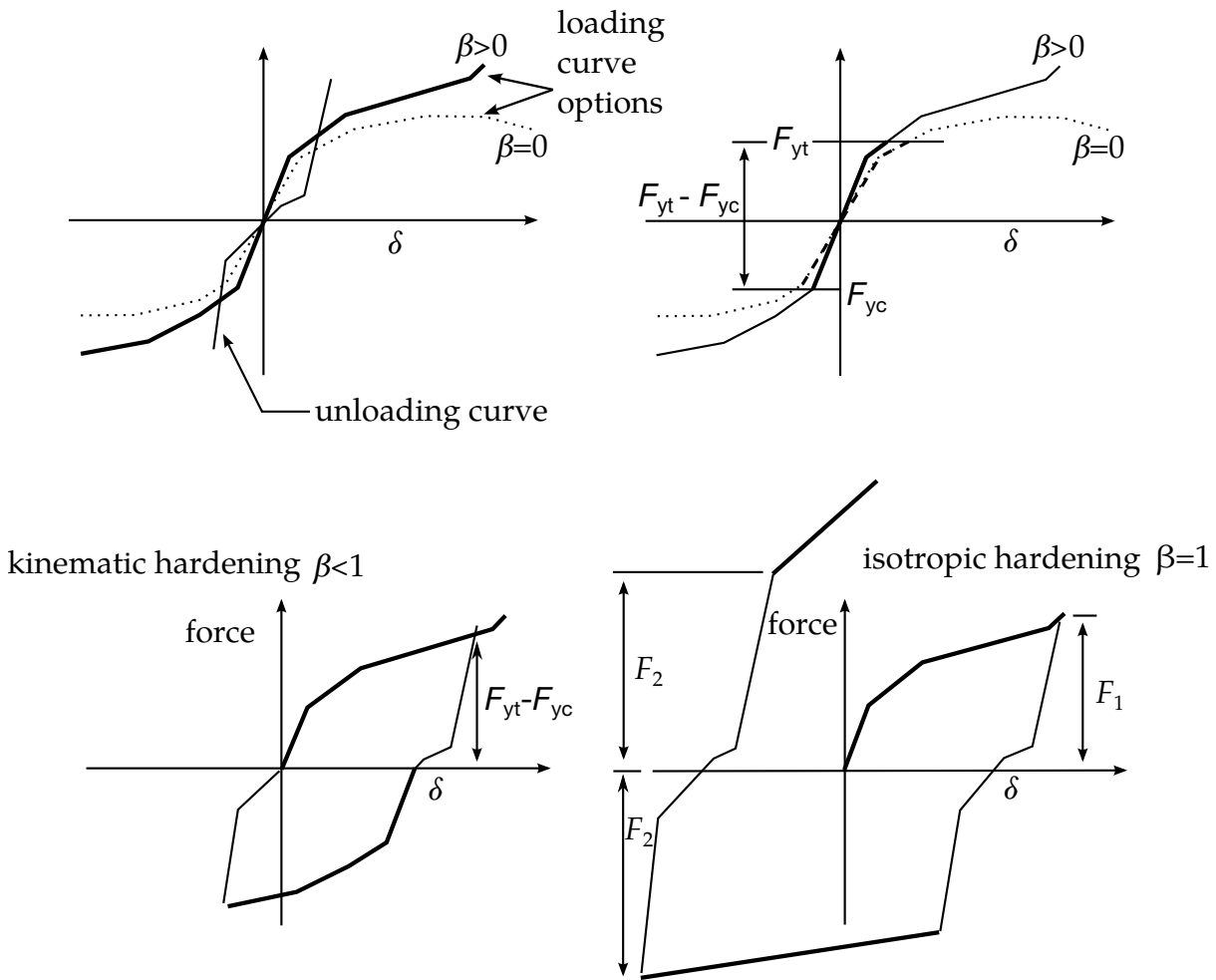


Figure 20.3. Loading and unloading force displacement curves for general non-linear discrete element.

where K_∞ is the long duration stiffness, K_0 is the short time stiffness, and β is a decay parameter that controls the rate at which the stiffness transitions between the short and long duration stiffness (Figure 16.4).

This model corresponds to a three-parameter Maxwell model (see insert in [Figure 20.4](#)) which consists of a spring and damper in series connected to another spring in parallel. Although this discrete element behavior could be built up using the above-described linear elastic and linear viscous discrete elements, such a model would also require the user to specify the nodal mass at the connection of the series spring and damper. This mass introduces a fourth parameter which would further complicate fitting the model to experimental data.

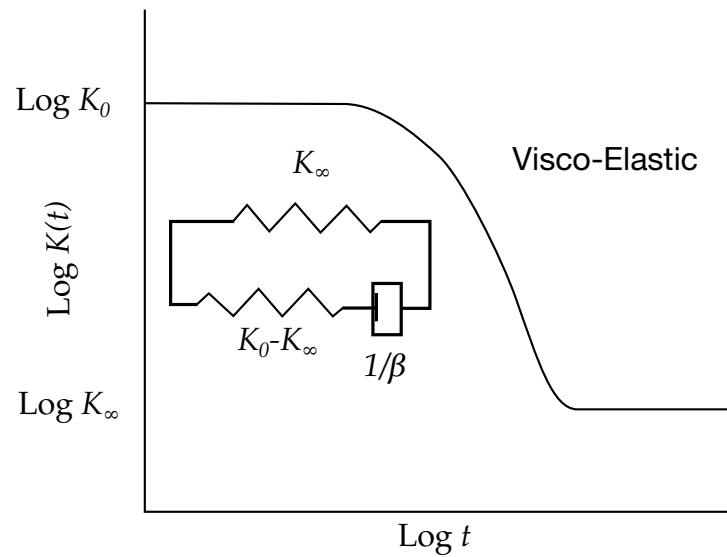


Figure 20.4. Typical stiffness relaxation curve used for the viscoelastic discrete element.

20.9 Muscle Model

This is Material Type 15 for discrete springs and dampers. This material is a Hill-type muscle model with activation. It is for use with discrete elements. The LS-DYNA implementation is due to Dr. J.A. Weiss.

- L0 Initial muscle length, L_0 .
- VMAX Maximum CE shortening velocity, V_{max} .
- SV Scale factor, S_v , for V_{max} vs. active state.
 - LT.0: absolute value gives load curve ID
 - GE.0: constant value of 1.0 is used
- A Activation level vs. time function.
 - LT.0: absolute value gives load curve ID
 - GE.0: constant value of A is used
- FMAXPeak isometric force, F_{max} .
- TL Active tension vs. length function.
 - LT.0: absolute value gives load curve ID
 - GE.0: constant value of 1.0 is used

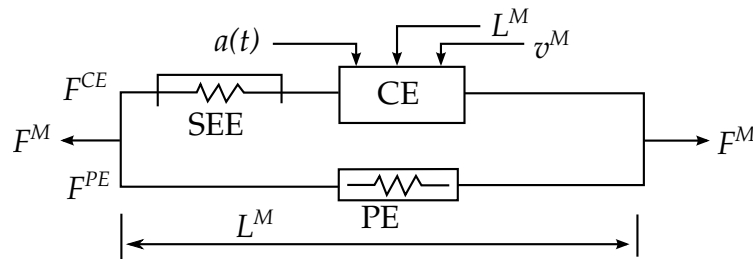


Figure 20.5. Discrete model for muscle contraction dynamics, based on a Hill-type representation. The total force is the sum of passive force F^{PE} and active force F^{CE} . The passive element (PE) represents energy storage from muscle elasticity, while the contractile element (CE) represents force generation by the muscle. The series elastic element (SEE), shown in dashed lines, is often neglected when a series tendon compliance is included. Here, $a(t)$ is the activation level, L^M is the length of the muscle, and v^M is the shortening velocity of the muscle.

TV Active tension vs. velocity function.

LT.0: absolute value gives load curve ID

GE.0: constant value of 1.0 is used

FPE Force vs. length function, F_{pe} , for parallel elastic element.

LT.0: absolute value gives load curve ID

EQ.0: exponential function is used (see below)

GT.0: constant value of 0.0 is used

LMAX Relative length when F_{pe} reaches F_{max} . Required if $F_{pe} = 0$ above.

KSH Constant, K_{sh} , governing the exponential rise of F_{pe} . Required if $F_{pe} = 0$ above.

The material behavior of the muscle model is adapted from the original model proposed by Hill (1938). Reviews of this model and extensions can be found in Winters (1990) and Zajac (1989). The most basic Hill-type muscle model consists of a contractile element (CE) and a parallel elastic element (PE) (Figure 20.5). An additional series elastic element (SEE) can be added to represent tendon compliance. The main assumptions of the Hill model are that the contractile element is entirely stress free and freely distensible in the resting state, and is described exactly by Hill's equation (or some variation). When the muscle is activated, the series and parallel elements are elastic, and the whole muscle is a simple combination of identical sarcomeres in series and parallel. The main criticism of Hill's model is that the division of forces between the parallel elements and the division of extensions between the series elements is arbitrary, and cannot be made without introducing auxiliary hypotheses. However, these criticisms apply to any discrete element model. Despite these limitations, the Hill model has become extremely useful for modeling musculoskeletal dynamics, as illustrated by its widespread use today.

When the contractile element (CE) of the Hill model is inactive, the entire resistance to elongation is provided by the PE element and the tendon load-elongation behavior. As activation is increased, force then passes through the CE side of the parallel Hill model, providing the contractile dynamics. The original Hill model accommodated

only full activation - this limitation is circumvented in the present implementation by using the modification suggested by Winters (1990). The main features of his approach were to realize that the CE force-velocity input force equals the CE tension-length output force. This yields a three-dimensional curve to describe the force-velocity-length relationship of the CE. If the force-velocity y-intercept scales with activation, then given the activation, length and velocity, the CE force can be determined.

Without the SEE, the total force in the muscle FM is the sum of the force in the CE and the PE because they are in parallel:

$$F^M = F^{PE} + F^{CE}. \quad (20.26)$$

The relationships defining the force generated by the CE and PE as a function of LM, VM and $a(t)$ are often scaled by F_{\max} , the peak isometric force (p. 80, Winters 1990), L_0 , the initial length of the muscle (p. 81, Winters 1990), and V_{\max} , the maximum unloaded CE shortening velocity (p. 80, Winters 1990). From these, dimensionless length and velocity can be defined:

$$L = \frac{L^M}{L_0}, \quad V = \frac{V^M}{V_{\max} * S_V(a(t))}. \quad (20.27)$$

Here, S_V scales the maximum CE shortening velocity V_{\max} and changes with activation level $a(t)$. This has been suggested by several researchers, i.e. Winters and Stark [1985]. The activation level specifies the level of muscle stimulation as a function of time. Both have values between 0 and 1. The functions $S_V(a(t))$ and $a(t)$ are specified via load curves in LS-DYNA, or default values of $S_V = 1$ and $a(t) = 0$ are used. Note that L is always positive and that V is positive for lengthening and negative for shortening.

The relationship between F^{CE} , V and L was proposed by Bahler et al. [1967]. A three-dimensional relationship between these quantities is now considered standard for computer implementations of Hill-type muscle models [i.e., eqn 5.16, p. 81, Winters 1990]. It can be written in dimensionless form as:

$$F^{CE} = a(t) * F_{\max} * f_{TL}(L) * f_{TV}(V), \quad (20.28)$$

The force in the parallel elastic element F^{PE} is determined directly from the current length of the muscle using an exponential relationship [eqn 5.5, p. 73, Winters 1990]:

$$f_{PE} = \frac{F^{PE}}{F_{MAX}} = 0 \quad L \leq 1$$

$$f_{PE} = \frac{F^{PE}}{F_{MAX}} = \frac{1}{\exp(K_{sh}) - 1} \left[\exp\left(\frac{K_{sh}}{L_{max}}(L - 1)\right) - 1 \right] \quad L > 1 \quad (20.29)$$

For computation of the total force developed in the muscle FM, the functions for the tension-length f_{TL} and force-velocity f_{TV} relationships used in the Hill element must

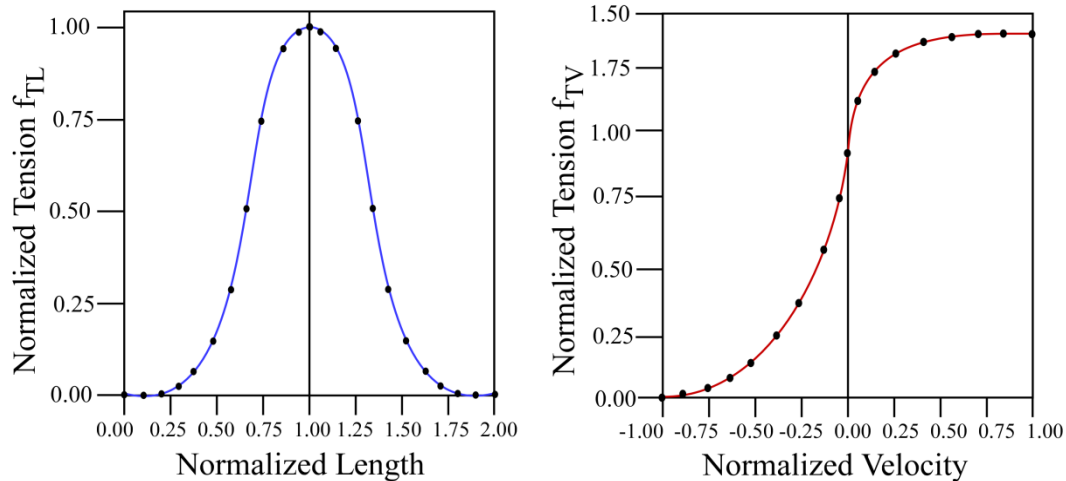


Figure 20.6. Typical normalized tension-length (TL) and tension-velocity (TV) curves for skeletal muscle.

be defined. These relationships have been available for over 50 years, but have been refined to allow for behavior such as active lengthening. The active tension-length curve f_{TL} describes the fact that isometric muscle force development is a function of length, with the maximum force occurring at an optimal length. According to Winters, this optimal length is typically around $L = 1.05$, and the force drops off for shorter or longer lengths, approaching zero force for $L = 0.4$ and $L = 1.5$. Thus the curve has a bell-shape. Because of the variability in this curve between muscles, the user must specify the function f_{TL} via a load curve, specifying pairs of points representing the normalized force (with values between 0 and 1) and normalized length L (Figure 20.6).

The active tension-velocity relationship f_{TV} used in the muscle model is mainly due to the original work of Hill. Note that the dimensionless velocity V is used. When $V = 0$, the normalized tension is typically chosen to have a value of 1.0. When V is greater than or equal to 0, muscle lengthening occurs. As V increases, the function is typically designed so that the force increases from a value of 1.0 and asymptotes towards a value near 1.4. When V is less than zero, muscle shortening occurs and the classic Hill equation hyperbola is used to drop the normalized tension to 0 (Figure 16.6). The user must specify the function f_{TV} via a load curve, specifying pairs of points representing the normalized tension (with values between 0 and 1) and normalized velocity V .

20.10 Seat Belt Material

The seat belt capability reported here was developed by Walker and co-workers [Walker and Dallard 1991, Strut, Walker, et al., 1991] and this section excerpted from their documentation. Each belt material defines stretch characteristics and mass properties for

a set of belt elements. The user enters a load curve for loading, the points of which are (Strain, Force). Strain is defined as engineering strain, i.e.

$$\text{Strain} = \frac{\text{current length}}{\text{initial length}} - 1. \quad (20.30)$$

Another similar curve is entered to describe the unloading behavior. Both loadcurves should start at the origin (0,0) and contain positive force and strain values only. The belt material is tension only with zero forces being generated whenever the strain becomes negative. The first non-zero point on the loading curve defines the initial yield point of the material. On unloading, the unloading curve is shifted along the strain axis until it crosses the loading curve at the 'yield' point from which unloading commences. If the initial yield has not yet been exceeded or if the origin of the (shifted) unloading curve is at negative strain, the original loading curves will be used for both loading and unloading. If the strain is less than the strain at the origin of the unloading curve, the belt is slack and no force is generated. Otherwise, forces will then be determined by the unloading curve for unloading and reloading until the strain again exceeds yield after which the loading curves will again be used.

A small amount of damping is automatically included. This reduces high frequency oscillation, but, with realistic force-strain input characteristics and loading rates, does not significantly alter the overall forces-strain performance. The damping forced opposes the relative motion of the nodes and is limited by stability:

$$D = \frac{.1 \times \text{mass} \times \text{relative velocity}}{\text{timestep size}}. \quad (20.31)$$

In addition, the magnitude of the damping forces is limited to one tenth of the force calculated from the forces-strain relationship and is zero when the belt is slack. Damping forces are not applied to elements attached to slings and retractors.

The user inputs a mass per unit length that is used to calculate nodal masses on initialization.

A 'minimum length' is also input. This controls the shortest length allowed in any element and determines when an element passes through slings or are absorbed into the retractors. One tenth of a typical initial element length is usually a good choice.

20.11 Seat Belt Elements

Belt elements are single degree of freedom elements connecting two nodes and are treated in a manner similar to the spring elements. When the strain in an element is

positive (i.e., the current length is greater than the unstretched length), a tension force is calculated from the material characteristics and is applied along the current axis of the element to oppose further stretching. The unstretched length of the belt is taken as the initial distance between the two nodes defining the position of the element plus the initial slack length. At the beginning of the calculation the seatbelt elements can be obtained within a retractor.

20.12 Sliprings

Sliprings are defined in the LS-DYNA input by giving a slipring ID and element ID's for two elements who share a node which is coincident with the slipring node. The slipring node may not be attached to any belt elements.

Sliprings allow continuous sliding of a belt through a sharp change of angle. Two elements (1 and 2 in [Figure 20.7](#)) meet at the slipring. Node B in the belt material remains attached to the slipring node, but belt material (in the form of unstretched length) is passed from element 1 to element 2 to achieve slip. The amount of slip at each timestep is calculated from the ratio of forces in elements 1 and 2. The ratio of forces is determined by the relative angle between elements 1 and 2 and the coefficient of friction, μ . The tension in the belts is taken as T_1 and T_2 , where T_2 is on the high-tension side and T_1 is the force on the low-tension side. Thus if T_2 is sufficiently close to T_1 no slip occurs; otherwise, slip is just sufficient to reduce the ratio T_2/T_1 to $e^{\mu\theta}$. No slip occurs if both elements are slack. The out-of-balance force at node B is reacted on the slipring node; the motion of node B follows that of slipring node.

If, due to slip through the slipring, the unstretched length of an element becomes less than the minimum length (as entered on the belt material card), the belt is remeshed locally: the short element passes through the slipring and reappears on the other side (see [Figure 20.7](#)). The new unstretched length of e_1 is $1.1 \times \text{minimum length}$. Force and strain in e_2 and e_3 are unchanged; force and strain in e_1 are now equal to those in e_2 . Subsequent slip will pass material from e_3 to e_1 . This process can continue with several elements passing in turn through the slipring.

To define a slipring, the user identifies the two belt elements which meet at the slipring, the friction coefficient, and the slipring node. The two elements must have a common node coincident with the slipring node. No attempt should be made to restrain or constrain the common node for its motion will automatically be constrained to follow the slipring node. Typically, the slipring node is part of the vehicle body structure and, therefore, belt elements should not be connected to this node directly, but any other feature can be attached, including rigid bodies.

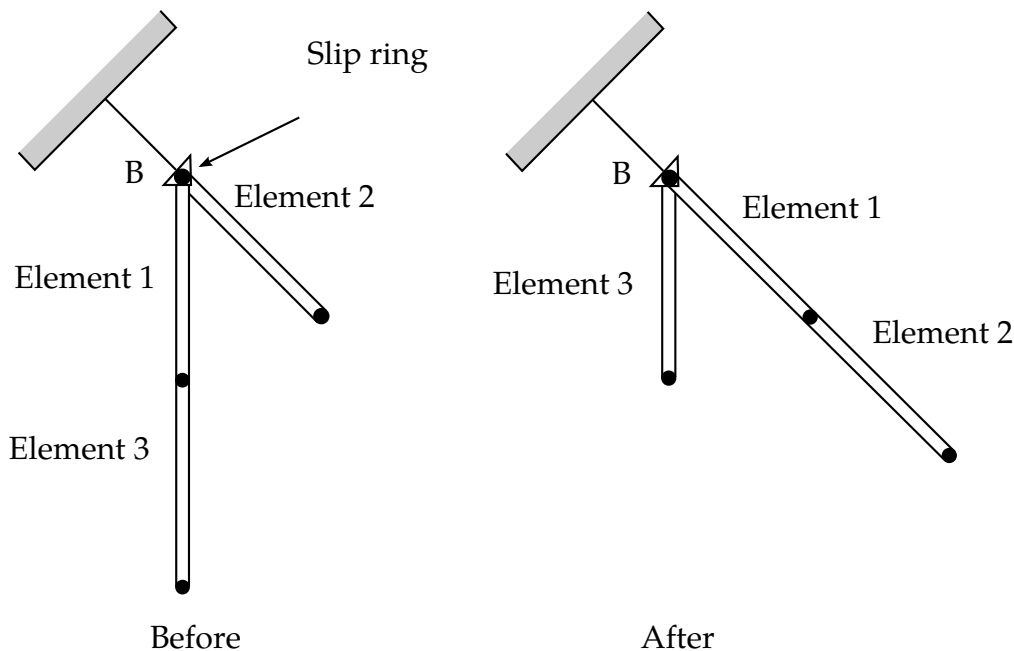


Figure 20.7. Elements passing through slipping.

20.13 Retractors

Retractors are defined by giving a node, the “retractor node” and an element ID of an element outside the retractor but with one node that is coincident with the retractor node. Also sensor ID’s must be defined for up to four sensors which can activate the seatbelt.

Retractors allow belt material to be paid out into a belt element, and they operate in one of two regimes: unlocked when the belt material is paid out or reeled in under constant tension and locked when a user defined force-pullout relationship applies.

The retractor is initially unlocked, and the following sequence of events must occur for it to become locked:

- Any one of up to four sensors must be triggered. (The sensors are described below).
- Then a user-defined time delay occurs.
- Then a user-defined length of belt must be paid out (optional).
- Then the retractor locks.

and once locked, it remains locked.

In the unlocked regime, the retractor attempts to apply a constant tension to the belt. This feature allows an initial tightening of the belt, and takes up any slack whenever

it occurs. The tension value is taken from the first point on the force-pullout load curve. The maximum rate of pull out or pull in is given by $0.01 \times \text{fed length}$ per time step. Because of this, the constant tension value is not always achieved.

In the locked regime, a user-defined curve describes the relationship between the force in the attached element and the amount of belt material paid out. If the tension in the belt subsequently relaxes, a different user-defined curve applies for unloading. The unloading curve is followed until the minimum tension is reached.

The curves are defined in terms of initial length of belt. For example, if a belt is marked at 10mm intervals and then wound onto a retractor, and the force required to make each mark emerge from the (locked) retractor is recorded, the curves used for input would be as follows:

```
0 Minimum tension (should be > zero)
10 mm Force to emergence of first mark
20 mm Force to emergence of second mark
..
..
..
```

Pyrotechnic pretensions may be defined which cause the retractor to pull in the belt at a predetermined rate. This overrides the retractor force-pullout relationship from the moment when the pretensioner activates.

If desired, belt elements may be defined which are initially inside the retractor. These will emerge as belt material is paid out, and may return into the retractor if sufficient material is reeled in during unloading.

Elements e2, e3 and e4 are initially inside the retractor, which is paying out material into element e1. When the retractor has fed L_{crit} into e1, where:

$$L_{\text{crit}} = \text{fed length} - 1.1 \times \text{minimum length} \quad (20.32)$$

Here, minimum length is defined on belt material input, and fed length is defined on retractor input.

element e2 emerges with an unstretched length of $1.1 \times \text{minimum length}$; the unstretched length of element e1 is reduced by the same amount. The force and strain in e1 are unchanged; in e2, they are set equal to those in e1. The retractor now pays out material into e2.

If no elements are inside the retractor, e2 can continue to extend as more material is fed into it.

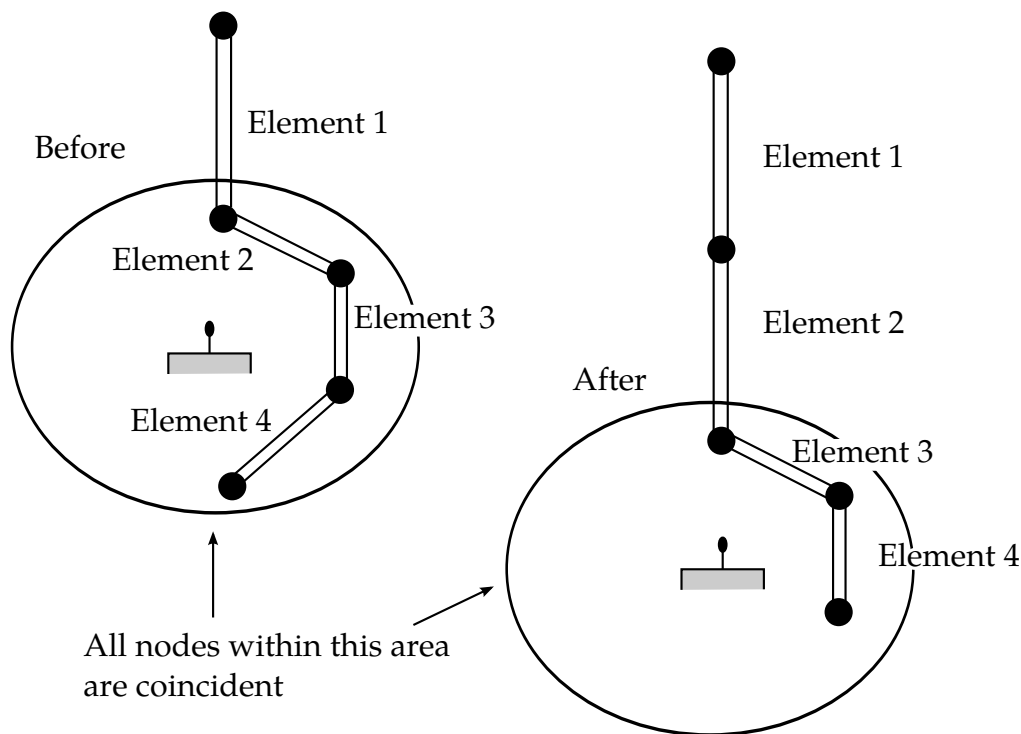


Figure 20.8. Elements in a retractor.

As the retractor pulls in the belt (for example, during initial tightening), if the unstretched length of the mouth element becomes less than the minimum length, the element is taken into the retractor.

To define a retractor, the user enters the retractor node, the 'mouth' element (into which belt material will be fed, e1 in [Figure 20.8](#), up to 4 sensors which can trigger unlocking, a time delay, a payout delay (optional), load and unload curve numbers, and the fed length. The retractor node is typically part of the vehicle structure; belt elements should not be connected to this node directly, but any other feature can be attached including rigid bodies. The mouth element should have a node coincident with the retractor but should not be inside the retractor. The fed length would typically be set either to a typical element initial length, for the distance between painted marks on a real belt for comparisons with high-speed film. The fed length should be at least three times the minimum length.

If there are elements initially inside the retractor (e2, e3 and e4 in the Figure) they should not be referred to on the retractor input, but the retractor should be identified on the element input for these elements. Their nodes should all be coincident with the retractor node and should not be restrained or constrained. Initial slack will automatically be set to $1.1 \times \text{minimum length}$ for these elements; this overrides any user-defined value.

Weblockers can be included within the retractor representation simply by entering a 'locking up' characteristic in the force pullout curve, see [Figure 20.9](#). The final section can be very steep (but must have a finite slope).

20.14 Sensors

Sensors are used to trigger locking of retractors and activate pretensioners. Four types of sensor are available which trigger according to the following criteria:

Type 1—When the magnitude of x-, y-, or z- acceleration of a given node has remained above a given level continuously for a given time, the sensor triggers. This does not work with nodes on rigid bodies.

Type 2—When the rate of belt payout from a given retractor has remained above a given level continuously for a given time, the sensor triggers.

Type 3—The sensor triggers at a given time.

Type 4—The sensor triggers when the distance between two nodes exceeds a given maximum or becomes less than a given minimum. This type of sensor is intended for use with an explicit mass/spring representation of the sensor mechanism.

By default, the sensors are inactive during dynamic relaxation. This allows initial tightening of the belt and positioning of the occupant on the seat without locking the retractor or firing any pretensioners. However, a flag can be set in the sensor input to make the sensors active during the dynamic relaxation phase.

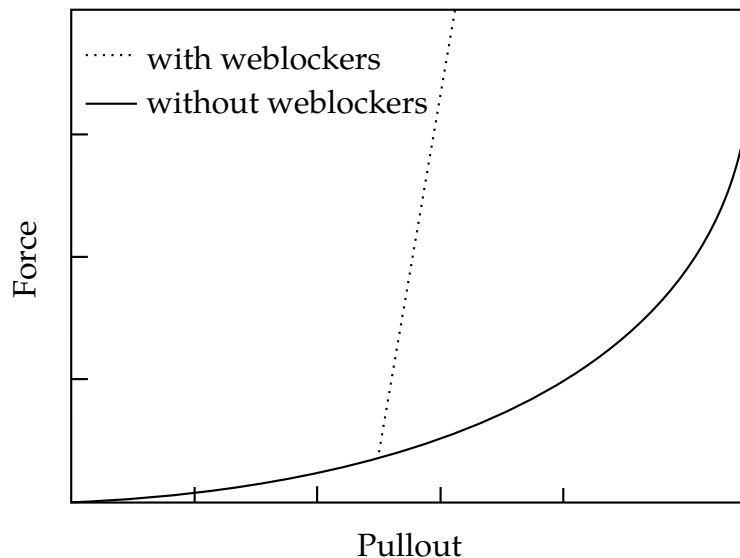


Figure 20.9. Retractor force pull characteristics.

20.15 Pretensioners

Pretensioners allow modeling of three types of active devices which tighten the belt during the initial stages of a crash. The first type represents a pyrotechnic device which spins the spool of a retractor, causing the belt to be reeled in. The user defines a pull-in versus time curve which applies once the pretensioner activates. The remaining types represent preloaded springs or torsion bars which move the buckle when released. The pretensioner is associated with any type of spring element including rotational. Note that the preloaded spring, locking spring and any restraints on the motion of the associated nodes are defined in the normal way; the action of the pretensioner is merely to cancel the force in one spring until (or after) it fires. With the second type, the force in the spring element is cancelled out until the pretensioner is activated. In this case the spring in question is normally a stiff, linear spring which acts as a locking mechanism, preventing motion of the seat belt buckle relative to the vehicle. A preloaded spring is defined in parallel with the locking spring. This type avoids the problem of the buckle being free to 'drift' before the pretensioner is activated.

To activate the pretensioner the following sequence of events must occur:

1. Any one of up to four sensors must be triggered.
2. Then a user-defined time delay occurs.
3. Then the pretensioner acts.

20.16 Accelerometers

The accelerometer is defined by three nodes in a rigid body which defines a triad to measure the accelerations in a local system. The presence of the accelerometer means that the accelerations and velocities of node 1 will be output to all output files in local instead of global coordinates.

The local coordinate system is defined by the three nodes as follows:

- local x from node 1 to node 2
- local z perpendicular to the plane containing nodes, 1, 2, and 3 ($z = x \times a$), where a is from node 1 to node 3).
- local $y = x \times z$

The three nodes should all be part of the same rigid body. The local axis then rotates with the body.

21

Simplified Arbitrary Lagrangian-Eulerian

Arbitrary Lagrangian-Eulerian (ALE) formulations may be thought of as algorithms that perform automatic rezoning. Users perform manual rezoning by

1. Stopping the calculation when the mesh is distorted,
2. Smoothing the mesh,
3. Remapping the solution from the distorted mesh to the smooth mesh.

An ALE formulation consists of a Lagrangian time step followed by a “remap” or “advection” step. The advection step performs an incremental rezone, where “incremental” refers to the fact that the positions of the nodes are moved only a small fraction of the characteristic lengths of the surrounding elements. Unlike a manual rezone, the topology of the mesh is fixed in an ALE calculation. An ALE calculation can be interrupted like an ordinary Lagrangian calculation and a manual rezone can be performed if an entirely new mesh is necessary to continue the calculation.

The accuracy of an ALE calculation is often superior to the accuracy of a manually rezoned calculation because the algorithm used to remap the solution from the distorted to the undistorted mesh is second order accurate for the ALE formulation while the algorithm for the manual rezone is only first order accurate.

In theory, an ALE formulation contains the Eulerian formulation as a subset. Eulerian codes can have more than one material in each element, but most ALE implementations are simplified ALE formulations which permit only a single material in each element. The primary advantage of a simplified formulation is its reduced cost per time step. When elements with more than one material are permitted, the number and types of materials present in an element can change dynamically. Additional data is necessary

to specify the materials in each element and the data must be updated by the remap algorithms.

The range of problems that can be solved with an ALE formulation is a direct function of the sophistication of the algorithms for smoothing the mesh. Early ALE codes were not very successful largely because of their primitive algorithms for smoothing the mesh. In simplified ALE formulations, most of the difficulties with the mesh are associated with the nodes on the material boundaries. If the material boundaries are purely Lagrangian, i.e., the boundary nodes move with the material at all times, no smooth mesh maybe possible and the calculation will terminate. The algorithms for maintaining a smooth boundary mesh are therefore as important to the robustness of the calculations as the algorithms for the mesh interior.

The cost of the advection step per element is usually much larger than the cost of the Lagrangian step. Most of the time in the advection step is spent in calculating the material transported between the adjacent elements, and only a small part of it is spent on calculating how and where the mesh should be adjusted. Second order accurate monotonic advection algorithms are used in LS-DYNA despite their high cost per element because their superior coarse mesh accuracy which allows the calculation to be performed with far fewer elements than would be possible with a cheaper first order accurate algorithm.

The second order transport accuracy is important since errors in the transport calculations generally smooth out the solution and reduce the peak values in the history variables. Monotonic advection algorithms are constructed to prevent the transport calculations from creating new minimum or maximum values for the solution variables. They were first developed for the solution of the Navier Stokes equations to eliminate the spurious oscillations that appeared around the shock fronts. Although monotonic algorithms are more diffusive than algorithms that are not monotonic, they must be used for stability in general purpose codes. Many constitutive models have history variables that have limited ranges, and if their values are allowed to fall outside of their allowable ranges, the constitutive models are undefined. Examples include explosive models, which require the burn fraction to be between zero and one, and many elastoplasticity models, such as those with power law hardening, which require a non-negative plastic strain.

The overall flow of an ALE time step is:

1. Perform a Lagrangian time step.
2. Perform an advection step.
 - a) Decide which nodes to move.
 - b) Move the boundary nodes.

- c) Move the interior nodes.
- d) Calculate the transport of the element-centered variables.
- e) Calculate the momentum transport and update the velocity.

Each element solution variable must be transported. The total number of solution variables, including the velocity, is at least six and depends on the material models. For elements that are modeled with an equation of state, only the density, the internal energy, and the shock viscosity are transported. When the elements have strength, the six components of the stress tensor and the plastic strain must also be advected, for a total of ten solution variables. Kinematic hardening, if it is used, introduces another five solution variables, for a total of fifteen.

The nodal velocities add an extra three solution variables that must be transported, and they must be advected separately from the other solution variables because they are centered at the nodes and not in the elements. In addition, the momentum must be conserved, and it is a product of the node-centered velocity and the element-centered density. This imposes a constraint on how the momentum transport is performed that is unique to the velocity field. A detailed consideration of the difficulties associated with the transport of momentum is deferred until later.

Perhaps the simplest strategy for minimizing the cost of the ALE calculations is to perform them only every few time steps. The cost of an advection step is typically two to five times the cost of the Lagrangian time step. By performing the advection step only every ten steps, the cost of an ALE calculation can often be reduced by a factor of three without adversely affecting the time step size. In general, it is not worthwhile to advect an element unless at least twenty percent of its volume will be transported because the gain in the time step size will not offset the cost of the advection calculations.

21.1 Mesh Smoothing Algorithms

The algorithms for moving the mesh relative to the material control the range of the problems that can be solved by an ALE formulation. The equipotential method which is used in LS-DYNA was developed by Winslow [1990] and is also used in the DYNA2D ALE code [Winslow 1963]. It, and its extensions, have proven to be very successful in a wide variety of problems. The following is extracted from reports prepared by Alan Winslow for LSTC.

21.1.1 Equipotential Smoothing of Interior Nodes

“Equipotential” zoning [Winslow, 1963] is a method of making a structured mesh for finite difference or finite element calculations by using the solutions of Laplace

equations (later extended to Poisson equations) as the mesh lines. The same method can be used to smooth selected points in an unstructured three-dimensional mesh provided that it is at least locally structured. This chapter presents a derivation of the three-dimensional equipotential zoning equations, taken from the references, and gives their finite difference equivalents in a form ready to be used for smoothing interior points. We begin by reviewing the well-known two-dimensional zoning equations, and then discuss their extension to three dimensions.

In two dimensions we define curvilinear coordinates ξ, η which satisfy Laplace's equation:

$$\nabla^2 \xi = 0, \quad (21.1.1a)$$

$$\nabla^2 \eta = 0. \quad (21.1.1b)$$

We solve [Equations \(21.1.1b\)](#) for the coordinates $x(\xi, \eta)$ and $y(\xi, \eta)$ of the mesh lines: that is, we invert them so that the geometric coordinates x, y become the dependent variables and the curvilinear coordinates ξ, η the independent variables. By the usual methods of changing variables we obtain

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0, \quad (21.1.2a)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0, \quad (21.1.2b)$$

where

$$\alpha \equiv x_\eta^2 + y_\eta^2, \quad \beta \equiv x_\xi x_\eta + y_\xi y_\eta, \quad \gamma \equiv x_\xi^2 + y_\xi^2. \quad (21.1.3)$$

Equations (21.1.2) can be written in vector form:

$$\alpha \mathbf{r}_{\xi\xi} - 2\beta \mathbf{r}_{\xi\eta} + \gamma \mathbf{r}_{\eta\eta} = \mathbf{0}, \quad (21.1.4)$$

where

$$\mathbf{r} \equiv x\mathbf{i} + y\mathbf{j}. \quad (21.1.5)$$

We differentiate [Equations \(21.1.4\)](#) and solve them numerically by an iterative method, since they are nonlinear. In (ξ, η) space, we use a mesh whose curvilinear coordinates are straight lines which take on integer values corresponding to the usual numbering in a two-dimensional mesh. The numerical solution then gives us the location of the "equipotential" mesh lines.

In three dimensions x, y, z , we add a third curvilinear coordinate ζ and a third Laplace equation

$$\nabla^2 \zeta = 0. \quad (21.1.1c)$$

Inversion of the system of three equations (17.1.1) by change of variable is rather complicated. It is easier, as well as more illuminating, to use the methods of tensor analysis pioneered by Warsi [1982]. Let the curvilinear coordinates be represented by ζ^i ($i = 1, 2, 3$). For a scalar function $A(x, y, z)$, Warsi shows that the transformation of its Laplacian from rectangular Cartesian to curvilinear coordinates is given by

$$\nabla^2 A = \sum_{i,j=1}^3 g^{ij} A_{\zeta^i \zeta^j} + \sum_{k=1}^3 (\nabla^2 \zeta^k) A_{\zeta^k}, \quad (21.1.6)$$

where a variable subscript indicates differentiation with respect to that variable. Since the curvilinear coordinates are each assumed to satisfy Laplace's equation, the second summation in Equation (21.1.6) vanishes and we have

$$\nabla^2 A = \sum_{i,j=1}^3 g^{ij} A_{\zeta^i \zeta^j}. \quad (21.1.7)$$

If now we let $A = x, y$, and z successively, the left-hand side of (21.1.7) vanishes in each case and we get three equations which we can write in vector form

$$\sum_{i,j=1}^3 g^{ij} \mathbf{r}_{\zeta^i \zeta^j} = 0. \quad (21.1.8)$$

Equation (21.1.8) is the three-dimensional generalization of Equations (21.1.4), and it only remains to determine the components of the contravariant metric tensor g_{ij} in three dimensions. These are defined to be

$$g^{ij} \equiv \mathbf{a}^i \cdot \mathbf{a}^j, \quad (21.1.9)$$

where the contravariant base vectors of the transformation from (x, y, z) to $(\zeta^1, \zeta^2, \zeta^3)$ are given by

$$\mathbf{a}^i \equiv \nabla \zeta^i = \frac{\mathbf{a}_j \times \mathbf{a}_k}{\sqrt{g}}, \quad (21.1.10)$$

(i, j, k cyclic). Here the covariant base vectors, the coordinate derivatives, are given by

$$\mathbf{a}_i \equiv \mathbf{r}_{\zeta^i}, \quad (21.1.11)$$

where

$$\mathbf{r} \equiv x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}. \quad (21.1.12)$$

Also,

$$g \equiv \det(g_{ij}) = [\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)]^2 = J^2, \quad (21.1.13)$$

where g_{ij} is the covariant metric tensor given by

$$g_{ij} \equiv \mathbf{a}_i \cdot \mathbf{a}_j, \quad (21.1.14)$$

and J is the Jacobian of the transformation.

Substituting (21.1.10) into (21.1.9), and using the vector identity

$$(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) \equiv (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c}), \quad (21.1.15)$$

we get

$$gg^{ii} = \mathbf{a}_j^2 \mathbf{a}_k^2 - (\mathbf{a}_j \cdot \mathbf{a}_k)^2 = g_{jj}g_{kk} - (g_{jk})^2, \quad (21.1.16)$$

$$gg^{ij} = (\mathbf{a}_i \cdot \mathbf{a}_k)(\mathbf{a}_j \cdot \mathbf{a}_k) - (\mathbf{a}_i \cdot \mathbf{a}_j)\mathbf{a}_k^2 = g_{ik}g_{jk} - g_{ij}g_{kk}. \quad (21.1.17)$$

Before substituting (21.1.11) into (17.1.13a, b), we return to our original notation:

$$\xi + \xi^1, \eta + \xi^2, \zeta + \xi^3. \quad (21.1.18)$$

Then, using (21.1.11), we get

$$gg^{11} = \mathbf{r}_\eta^2 \mathbf{r}_\xi^2 - (\mathbf{r}_\eta \cdot \mathbf{r}_\xi)^2, \quad (21.1.19)$$

$$gg^{22} = \mathbf{r}_\xi^2 \mathbf{r}_\zeta^2 - (\mathbf{r}_\xi \cdot \mathbf{r}_\zeta)^2, \quad (21.1.20)$$

$$gg^{33} = \mathbf{r}_\xi^2 \mathbf{r}_\eta^2 - (\mathbf{r}_\xi \cdot \mathbf{r}_\eta)^2, \quad (21.1.21)$$

for the three diagonal components, and

$$gg^{12} = (\mathbf{r}_\xi \cdot \mathbf{r}_\zeta)(\mathbf{r}_\eta \cdot \mathbf{r}_\zeta) - (\mathbf{r}_\xi \cdot \mathbf{r}_\eta)\mathbf{r}_\zeta^2, \quad (21.1.22)$$

$$gg^{23} = (\mathbf{r}_\eta \cdot \mathbf{r}_\xi)(\mathbf{r}_\zeta \cdot \mathbf{r}_\xi) - (\mathbf{r}_\eta \cdot \mathbf{r}_\zeta)\mathbf{r}_\xi^2, \quad (21.1.23)$$

$$gg^{31} = (\mathbf{r}_\zeta \cdot \mathbf{r}_\eta)(\mathbf{r}_\xi \cdot \mathbf{r}_\eta) - (\mathbf{r}_\zeta \cdot \mathbf{r}_\xi)\mathbf{r}_\eta^2, \quad (21.1.24)$$

for the three off-diagonal components of this symmetric tensor.

When we express Equations (17.1.15) in terms of the Cartesian coordinates, some cancellation takes place and we can write them in the form

$$gg^{11} = (x_\eta y_\zeta - x_\zeta y_\eta)^2 + (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (y_\eta z_\zeta - y_\zeta z_\eta)^2, \quad (21.1.25)$$

$$gg^{22} = (x_\zeta y_\xi - x_\xi y_\zeta)^2 + (x_\zeta z_\xi - x_\xi z_\zeta)^2 + (y_\zeta z_\xi - y_\xi z_\zeta)^2, \quad (21.1.26)$$

$$gg^{33} = (x_\xi y_\eta - x_\eta y_\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2 + (y_\xi z_\eta - y_\eta z_\xi)^2, \quad (21.1.27)$$

guaranteeing positivity as required by Equations (21.1.9). Writing out Equations (17.1.16) we get

$$\begin{aligned} gg^{12} &= (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)(x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta) \\ &\quad - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2), \end{aligned} \quad (21.1.28)$$

$$gg^{23} = (x_\eta x_{\bar{\zeta}} + y_\eta y_{\bar{\zeta}} + z_\eta z_{\bar{\zeta}})(x_{\bar{\zeta}} x_{\bar{\zeta}} + y_{\bar{\zeta}} y_{\bar{\zeta}} + z_{\bar{\zeta}} z_{\bar{\zeta}}) - (x_\eta x_{\bar{\zeta}} + y_\eta y_{\bar{\zeta}} + z_\eta z_{\bar{\zeta}})(x_{\bar{\zeta}}^2 + y_{\bar{\zeta}}^2 + z_{\bar{\zeta}}^2), \quad (21.1.29)$$

$$gg^{31} = (x_{\bar{\zeta}} x_\eta + y_{\bar{\zeta}} y_\eta + z_{\bar{\zeta}} z_\eta)(x_{\bar{\zeta}} x_\eta + y_{\bar{\zeta}} y_\eta + z_{\bar{\zeta}} z_\eta) - (x_{\bar{\zeta}} x_{\bar{\zeta}} + y_{\bar{\zeta}} y_{\bar{\zeta}} + z_{\bar{\zeta}} z_{\bar{\zeta}})(x_\eta^2 + y_\eta^2 + z_\eta^2). \quad (21.1.30)$$

Hence, we finally write [Equations \(21.1.8\)](#) in the form

$$g(g^{11}\mathbf{r}_{\bar{\zeta}\bar{\zeta}} + g^{22}\mathbf{r}_{\eta\eta} + g^{33}\mathbf{r}_{\bar{\zeta}\bar{\zeta}} + 2g^{12}\mathbf{r}_{\bar{\zeta}\eta} + 2g^{23}\mathbf{r}_{\eta\bar{\zeta}} + 2g^{31}\mathbf{r}_{\bar{\zeta}\zeta}) = 0, \quad (21.1.31)$$

where the gg^{ij} are given by [Equations \(17.1.17\)](#) and [\(17.1.18\)](#). Because [Equations \(21.1.8\)](#) are homogeneous, we can use gg^{ij} in place of g^{ij} as long as g is positive, as it must be for a nonsingular transformation. We can test for positivity at each mesh point by using [Equation \(1.1.32\)](#):

$$\sqrt{g} = J = \begin{vmatrix} x_{\bar{\zeta}} & y_{\bar{\zeta}} & z_{\bar{\zeta}} \\ x_\eta & y_\eta & z_\eta \\ x_{\bar{\zeta}} & y_{\bar{\zeta}} & z_{\bar{\zeta}} \end{vmatrix}, \quad (21.1.32)$$

and requiring that $J > 0$.

To check that these equations reproduce the two-dimensional equations when there is no variation in one-dimension, we take $\bar{\zeta}$ as the invariant direction, thus reducing [\(17.1.19\)](#) to

$$gg^{11}\mathbf{r}_{\bar{\zeta}\bar{\zeta}} + 2gg^{12}\mathbf{r}_{\bar{\zeta}\eta} + gg^{22}\mathbf{r}_{\eta\eta} = 0. \quad (21.1.33)$$

If we let $\bar{\zeta} = z$, then the covariant base vectors become

$$\mathbf{a}_1 = x_{\bar{\zeta}}\mathbf{i} + y_{\bar{\zeta}}\mathbf{j}, \quad (21.1.34)$$

$$\mathbf{a}_2 = x_\eta\mathbf{i} + y_\eta\mathbf{j}, \quad (21.1.35)$$

$$\mathbf{a}_3 = \mathbf{k}. \quad (21.1.36)$$

From [\(17.1.22\)](#), using [\(17.1.13\)](#), we get

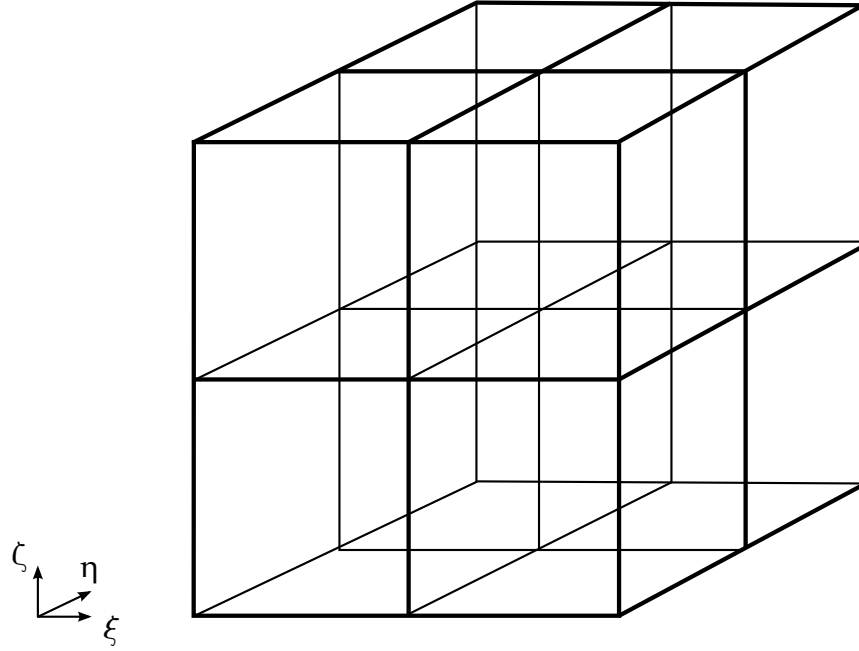
$$gg^{11} = x_\eta^2 + y_\eta^2, \quad (21.1.37)$$

$$gg^{22} = x_{\bar{\zeta}}^2 + y_{\bar{\zeta}}^2, \quad (21.1.38)$$

$$gg^{12} = -(x_{\bar{\zeta}}x_\eta + y_{\bar{\zeta}}y_\eta). \quad (21.1.39)$$

Substituting [\(17.1.23\)](#) into [\(21.1.33\)](#) yields the two-dimensional equipotential zoning [Equations \(17.1.2\)](#).

Before differencing [Equations \(17.1.19\)](#) we simplify the notation and write them in the form

**Figure 21.1.** Example Caption

$$\alpha_1 \mathbf{r}_{\xi\xi} + \alpha_2 \mathbf{r}_{\eta\eta} + \alpha_3 \mathbf{r}_{\zeta\zeta} + 2\beta_1 \mathbf{r}_{\xi\eta} + 2\beta_2 \mathbf{r}_{\eta\zeta} + 2\beta_3 \mathbf{r}_{\zeta\xi} = 0, \quad (21.1.40)$$

where

$$\alpha_1 = (x_\eta y_\xi - x_\xi y_\eta)^2 + (x_\eta z_\xi - x_\xi z_\eta)^2 + (y_\eta z_\xi - y_\xi z_\eta)^2, \quad (21.1.41)$$

$$\alpha_2 = (x_\xi y_\xi - x_\xi y_\xi)^2 + (x_\xi z_\xi - x_\xi z_\xi)^2 + (y_\xi z_\xi - y_\xi z_\xi)^2, \quad (21.1.42)$$

$$\alpha_3 = (x_\xi y_\eta - x_\eta y_\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2 + (y_\xi z_\eta - y_\eta z_\xi)^2. \quad (21.1.43)$$

$$\begin{aligned} \beta_1 = & (x_\xi x_\xi + y_\xi y_\xi + z_\xi z_\xi)(x_\eta x_\xi + y_\eta y_\xi + z_\eta z_\xi) \\ & - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\xi^2 + y_\xi^2 + z_\xi^2), \end{aligned} \quad (21.1.44)$$

$$\begin{aligned} \beta_2 = & (x_\eta x_\xi + y_\eta y_\xi + z_\eta z_\xi)(x_\xi x_\xi + y_\xi y_\xi + z_\xi z_\xi) \\ & - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\xi^2 + y_\xi^2 + z_\xi^2), \end{aligned} \quad (21.1.45)$$

$$\begin{aligned} \beta_3 = & (x_\eta x_\xi + y_\eta y_\xi + z_\eta z_\xi)(x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta) \\ & - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\eta^2 + y_\eta^2 + z_\eta^2), \end{aligned} \quad (21.1.46)$$

We difference [Equations \(21.1.40\)](#) in a cube in the rectangular $\xi \eta \zeta$ space with unit spacing between the coordinate surfaces, using subscript i to represent the ξ direction, j the η direction, and k the ζ direction, as shown in [Figure 21.1](#).

Using central differencing, we obtain the following finite difference approximations for the coordinate derivatives:

$$\mathbf{r}_{\xi} = (\mathbf{r}_{i+1} - \mathbf{r}_{i-1})/2, \quad (21.1.47)$$

$$\mathbf{r}_{\eta} = (\mathbf{r}_{j+1} - \mathbf{r}_{j-1})/2, \quad (21.1.48)$$

$$\mathbf{r}_{\zeta} = (\mathbf{r}_{k+1} - \mathbf{r}_{k-1})/2, \quad (21.1.49)$$

$$\mathbf{r}_{\xi\xi} = (\mathbf{r}_{i+1} - 2\mathbf{r} + \mathbf{r}_{i-1}), \quad (21.1.50)$$

$$\mathbf{r}_{\eta\eta} = (\mathbf{r}_{j+1} - 2\mathbf{r} + \mathbf{r}_{j-1}), \quad (21.1.51)$$

$$\mathbf{r}_{\zeta\zeta} = (\mathbf{r}_{k+1} - 2\mathbf{r} + \mathbf{r}_{k-1}), \quad (21.1.52)$$

$$\mathbf{r}_{\xi\eta} = \frac{1}{4} [(\mathbf{r}_{i+1,j+1} + \mathbf{r}_{i-1,j-1}) - (\mathbf{r}_{i+1,j-1} + \mathbf{r}_{i-1,j+1})], \quad (21.1.53)$$

$$\mathbf{r}_{\eta\zeta} = \frac{1}{4} [(\mathbf{r}_{j+1,k+1} + \mathbf{r}_{j-1,k-1}) - (\mathbf{r}_{j+1,k-1} + \mathbf{r}_{j-1,k+1})], \quad (21.1.54)$$

$$\mathbf{r}_{\xi\zeta} = \frac{1}{4} [(\mathbf{r}_{i+1,k+1} + \mathbf{r}_{i-1,k-1}) - (\mathbf{r}_{i+1,k-1} + \mathbf{r}_{i-1,k+1})], \quad (21.1.55)$$

where for brevity we have omitted subscripts i, j , or k (e.g., $k + 1$ stands for $i, j, k + 1$). Note that these difference expressions use only coordinate planes that pass through the central point, and therefore do not include the eight corners of the cube.

Substituting Equations (17.1.26) into (17.1.24,17.1.25) and collecting terms, we get

$$\sum_{m=1}^{18} \omega_m (\mathbf{r}_m - \mathbf{r}) = 0, \quad (21.1.56)$$

where the sum is over the 18 nearest (in the transform space) neighbors of the given point. The coefficients ω_m are given in Table 17.1.

Equations (21.1.56) can be written

$$\mathbf{r}_m = \frac{\sum_m \omega_m \mathbf{r}_m}{\sum_m \omega_m}, \quad (21.1.57)$$

expressing the position of the central point as a weighted mean of its 18 nearest neighbors. The denominator of (21.1.57) is equal to $2(\alpha_1 + \alpha_2 + \alpha_3)$ which is guaranteed to be positive by (17.1.25). This vector equation is equivalent to the three scalar equations

$$x = \frac{\sum_m \omega_m x_m}{\sum_m \omega_m}, \quad (21.1.58)$$

$$y = \frac{\sum_m \omega_m y_m}{\sum_m \omega_m}, \quad (21.1.59)$$

<u>m</u>	<u>Index</u>	ω_m
1	$i + 1$	α_1
2	$i - 1$	α_1
3	$j + 1$	α_2
4	$j - 1$	α_2
5	$k + 1$	α_3
6	$k - 1$	α_3
7	$i + 1, j + 1$	$\beta_1/2$
8	$i - 1, j - 1$	$\beta_1/2$
9	$i + 1, j - 1$	$-\beta_1/2$
10	$i - 1, j + 1$	$-\beta_1/2$
11	$j + 1, k + 1$	$\beta_2/2$
12	$j - 1, k - 1$	$\beta_2/2$
13	$j + 1, k - 1$	$-\beta_2/2$
14	$j - 1, k + 1$	$-\beta_2/2$
15	$i + 1, k + 1$	$\beta_3/2$
16	$i - 1, k - 1$	$\beta_3/2$
17	$i + 1, k - 1$	$-\beta_3/2$
18	$i - 1, k + 1$	$-\beta_3/2$

Table 17.1. 3D Zoning Weight Coefficients

$$z = \frac{\sum_m \omega_m z_m}{\sum_m \omega_m}. \quad (21.1.60)$$

the same weights ω_m appearing in each equation.

These equations are nonlinear, since the coefficients are functions of the coordinates. Therefore we solve them by an iterative scheme, such as SOR. When applied to [Equation \(21.1.58\)](#), for example, this gives for the $(n+1)$ st iteration

$$x^{n+1} = (1 - f)x^n + f \left(\frac{\sum_m \omega_m x_m}{\sum_m \omega_m} \right), \quad (21.1.61)$$

where the over relaxation factor f must satisfy $0 < f < 2$. In [\(21.1.61\)](#) the values of x_m at the neighboring points are the latest available values. The coefficients ω_m are recalculated before each iteration using Table 17.1 and Equations (17.1.25).

To smooth one interior point in a three-dimensional mesh, let the point to be smoothed be the interior point of Figure 17.1, assuming that its neighborhood has the logical structure shown. Even though Equations (17.1.29) are nonlinear, the ω_m do not

involve the coordinates of the central point, since the α 's and β 's do not. Hence we simply solve Equations (17.1.29) for the new coordinates (x, y, z) , holding the 18 neighboring points fixed, without needing to iterate.

If we wish to smooth a group of interior points, we solve iteratively for the coordinates using equations of the form (21.1.61).

21.1.2 Simple Averaging

The coordinates of a node is the simple average of the coordinates of its surrounding nodes.

$$x_{SA}^{n+1} = \frac{1}{m^{\text{tot}}} \sum_{m=1}^{m^{\text{tot}}} x_m^n. \quad (21.1.62)$$

21.1.3 Kikuchi's Algorithm

Kikuchi proposed an algorithm that uses a volume-weighted average of the coordinates of the centroids of the elements surrounding a node. Variables that are subscripted with Greek letters refer to element variables, and subscripts with capital letters refer to the local node numbering within an element.

$$x_\alpha^n = \frac{1}{8} \sum_A x_{A'}^n, \quad (21.1.63)$$

$$x_K^{n+1} = \frac{\sum_{\alpha=1}^{\alpha^{\text{tot}}} V_a x_\alpha^n}{\sum_{\alpha=1}^{\alpha^{\text{tot}}} V_a}. \quad (21.1.64)$$

21.1.4 Surface Smoothing

The surfaces are smoothed by extending the two-dimensional equipotential stencils to three dimensions. Notice that the form of Equation (21.1.2a) and (21.1.2b) for the x and y directions are identical. The third dimension, z , takes the same form. When Equation (16.1.2) is applied to all three dimensions, it tends to flatten out the surface and alter the total volume. To conserve the volume and retain the curvature of the surface, the point given by the relaxation stencil is projected on to the tangent plane defined by the normal at the node.

21.1.5 Combining Smoothing Algorithms

The user has the option of using a weighted average of all three algorithms to generate a composite algorithm, where the subscripts E, SA, and K refer to the equipotential, simple averaging, and Kikuchi's smoothing algorithm respectively, and ww is the weighting factor.

$$x^{n+1} = w_E x_E^{n+1} + w_{SA} x_{SA}^{n+1} + w_K x_K^{n+1}. \quad (21.1.65)$$

21.2 Advection Algorithms

LS-DYNA follows the SALE3D strategy for calculating the transport of the element-centered variables (i.e., density, internal energy, the stress tensor and the history variables). The van Leer MUSCL scheme [van Leer 1977] is used instead of the donor cell algorithm to calculate the values of the solution variables in the transport fluxes to achieve second order accurate monotonic results. To calculate the momentum transport, two algorithms have been implemented. The less expensive of the two is the one that is implemented in SALE3D, but it has known dispersion problems and may violate monotonicity (i.e., introduce spurious oscillations) [Benson 1992]. As an alternative, a significantly more expensive method [Benson 1992], which can be shown analytically to not have either problem on a regular mesh, has also been implemented.

In this section the donor cell and van Leer MUSCL scheme are discussed. Both methods are one-dimensional and their extensions to multidimensional problems are discussed later.

21.2.1 Advection Methods in One Dimension

In this section the donor cell and van Leer MUSCL scheme are discussed. Both methods are one-dimensional and their extensions to multidimensional problems are discussed later.

The remap step maps the solution from a distorted Lagrangian mesh on to the new mesh. The underlying assumptions of the remap step are 1) the topology of the mesh is fixed (a complete rezone does not have this limitation), and 2) the mesh motion during a step is less than the characteristic lengths of the surrounding elements. Within the fluids community, the second condition is simply stated as saying the Courant number, C , is less than one.

$$C = \frac{u\Delta t}{\Delta x} = \frac{f}{V} \leq 1, \quad (21.2.66)$$

Since the mesh motion does not occur over any physical time scale, Δt is arbitrary, and $u\Delta t$ is the transport volume, f , between adjacent elements. The transport volume calculation is purely geometrical for ALE formulations and it is not associated with any of the physics of the problem.

The algorithms for performing the remap step are taken from the computational fluids dynamics community, and they are referred to as “advection” algorithms after the

first order, scalar conservation equation that is frequently used as a model hyperbolic problem.

$$\frac{\partial \phi}{\partial t} + a(x) \frac{\partial \phi}{\partial x} = 0. \quad (21.2.67)$$

A good advection algorithm for the remap step is accurate, stable, conservative and monotonic. Although many of the solution variables, such as the stress and plastic strain, are not governed by conservation equations like momentum and energy, it is still highly desirable that the volume integral of all the solution variables remain unchanged by the remap step. Monotonicity requires that the range of the solution variables does not increase during the remap. This is particularly important with mass and energy, where negative values would lead to physically unrealistic solutions.

Much of the research on advection algorithms has focused on developing monotonic algorithms with an accuracy that is at least second order. Not all recent algorithms are monotonic. For example, within the finite element community, the streamline upwind Petrov-Galerkin (SUPG) method developed by Hughes and coworkers [Brooks and Hughes 1982] is not monotonic. Johnson et al., [1984] have demonstrated that the oscillations in the SUPG solution are localized, and its generalization to systems of conservation equations works very well for the Euler equations. Mizukami and Hughes [1985] later developed a monotonic SUPG formulation. The essentially non-oscillatory (ENO) [Harten 1989] finite difference algorithms are also not strictly monotonic, and work well for the Euler equations, but their application to hydrodynamics problems has resulted in negative densities [McGlaun 1990]. Virtually all the higher order methods that are commonly used were originally developed for solving the Euler equations, usually as higher order extensions to Godunov's method. Since the operator split approach is the dominant one in Eulerian hydrocodes, these methods are implemented only to solve the scalar advection equation.

The Donor Cell Algorithm. Aside from its first order accuracy, it is everything a good advection algorithm should be: stable, monotonic, and simple. The value of f_j^ϕ is dependent on the sign of a at node j , which defines the upstream direction.

$$\phi_{j+1/2}^{n+1} = \phi_{j+1/2}^n + \frac{\Delta t}{\Delta x} (f_j^\phi - f_{j+1}^\phi), \quad (21.2.68)$$

$$f_j^\phi = \frac{a_j}{2} (\phi_{j-1/2}^n + \phi_{j+1/2}^n) + \frac{|a_j|}{2} (\phi_{j-1/2}^n - \phi_{j+1/2}^n). \quad (21.2.69)$$

The donor cell algorithm is a first order Godunov method applied to the advection equation. The initial values of ϕ to the left and the right of node j are $\phi_{j-1/2}^n$ and $\phi_{j+1/2}^n$, and the velocity of the contact discontinuity at node j is a_j .

The Van Leer MUSCL Algorithm. Van Leer [1977] introduced a family of higher order Godunov methods by improving the estimates of the initial values of left and right states for the Riemann problem at the nodes. The particular advection algorithm that is presented in this section is referred to as the MUSCL (monotone upwind schemes for conservation laws) algorithm for brevity, although MUSCL really refers to the family of algorithms that can be applied to systems of equations.

The donor cell algorithm assumes that the distribution of ϕ is constant over an element. Van Leer replaces the piecewise constant distribution with a higher order interpolation function, $\phi_{j+1/2}^n(x)$ that is subject to an element level conservation constraint. The value of ϕ at the element centroid is regarded in this context as the average value of ϕ over the element instead of the spatial value at $x_{j+1/2}$.

$$\phi_{j+1/2}^n = \int_{x_j}^{x_{j+1}} \phi_{j+1/2}^n(x) dx, \quad (21.2.70)$$

To determine the range of ϕ , $[\phi_{j+1/2}^{\min}, \phi_{j+1/2}^{\max}]$, for imposing the monotonicity constraint, the maximum and minimum values of $\phi_{j-1/2}^n$, $\phi_{j+1/2}^n$, and $\phi_{j+3/2}^n$ are used. Monotonicity can be imposed in either of two ways. The first is to require that the maximum and minimum values of $\phi_{j+1/2}^n(x)$ fall within the range determined by the three elements. The second is to restrict the average value of ϕ in the transport volumes associated with element $j + 1/2$. While the difference may appear subtle, the actual difference between the two definitions is quite significant even at relatively low Courant numbers. The second definition allows the magnitude of the ϕ transported to adjacent elements to be larger than the first definition. As a consequence, the second definition is better able to transport solutions with large discontinuities. The magnitude of ϕ an algorithm is able to transport before its monotonicity algorithm restricts ϕ is a measure of the algorithm's "compressiveness."

The first step up from a piecewise constant function is a piecewise linear function, where x is now the volume coordinate. The volume coordinate of a point is simply the volume swept along the path between the element centroid and the point. Conservation is guaranteed by expanding the linear function about the element centroid.

$$\phi_{j+1/2}^n(x) = S_{j+1/2}^n (x - x_{j+1/2}^n) + \phi_{j+1/2}^n. \quad (21.2.71)$$

Letting $s_{j+1/2}^n$ be a second order approximation of the slope, the monotonicity limited value of the slope, $s_{j+1/2}^n$, according to the first limiting approach, is determined by assuming the maximum permissible values at the element boundaries.

$$S_{j+1/2}^n = \frac{1}{2} (\text{sgn}(s^L) + \text{sgn}(s^R)) \times \min(|s^L|, |s_{j+1/2}^n|, |s^R|), \quad (21.2.72)$$

$$s^L = \frac{\phi_{j+1/2}^n - \phi_{j-1/2}^n}{\frac{1}{2}\Delta x_{j+1/2}}, \quad (21.2.73)$$

$$s^R = \frac{\phi_{j+3/2}^n - \phi_{j+1/2}^n}{\frac{1}{2}\Delta x_{j+1/2}}. \quad (21.2.74)$$

The second limiter is similar to the first, but it assumes that the maximum permissible values occur at the centroid of the transport volumes. Note that as stated in Equation (17.2.6), this limiter still limits the slope at the element boundary even if the element is the downstream element at that boundary. A more compressive limiter would not limit the slope based on the values of ϕ at the downstream boundaries. For example, if a_j is negative, only S_R would limit the value of S_n in Equation (17.2.6). If the element is the downstream element at both boundaries, then the slope in the element has no effect on the solution.

$$s^L = \frac{\phi_{j+1/2}^n - \phi_{j-1/2}^n}{\frac{1}{2}\Delta x_{j+1/2} - \frac{1}{2}\max(0, a_j\Delta t)}. \quad (21.2.75)$$

$$s^R = \frac{\phi_{j+3/2}^n - \phi_{j+1/2}^n}{\frac{1}{2}\Delta x_{j+1/2} + \frac{1}{2}\min(0, a_{j+1}\Delta t)}. \quad (21.2.76)$$

The flux at node j is evaluated using the upstream approximation of ϕ .

$$f_j^\phi = \frac{a_j}{2}(\phi_j^- + \phi_j^+) + \frac{|a_j|}{2}(\phi_j^- - \phi_j^+), \quad (21.2.77)$$

$$\phi_j^+ = S_{j+1/2}^n(x^C - x_{j+1/2}^n) + \phi_{j+1/2}^n, \quad (21.2.78)$$

$$\phi_j^- = S_{j-1/2}^n(x^C - x_{j-1/2}^n) + \phi_{j-1/2}^n, \quad (21.2.79)$$

$$x^C = x_j^n + \frac{1}{2}a_j\Delta t. \quad (21.2.80)$$

The method for obtaining the higher order approximation of the slope is not unique. Perhaps the simplest approach is to fit a parabola through the centroids of the three adjacent elements and evaluate its slope at $x_{j+1/2}^n$. When the value of ϕ at the element centroids is assumed to be equal to the element average this algorithm defines a projection.

$$s_{j+1/2}^n = \frac{(\varphi_{j+3/2}^n - \varphi_{j+1/2}^n) \Delta x_j^2 + (\varphi_{j+1/2}^n - \varphi_{j-1/2}^n) \Delta x_{j+1}^2}{\Delta x_j \Delta x_{j+1} (\Delta x_j + \Delta x_{j+1})}, \quad (21.2.81)$$

$$\Delta x_j = x_{j+1/2}^n - x_{j-1/2}^n. \quad (21.2.82)$$

21.2.2 Advection Methods in Three Dimensions

For programs that use a logically regular mesh, one-dimensional advection methods are extended to two and three dimensions by using a sequence of one-dimensional sweeps along the logically orthogonal mesh lines. This strategy is not possible for unstructured meshes because they don't have uniquely defined sweep directions through the mesh. CAVEAT [Addessio, et al., 1986] uses one-dimensional sweeps in the spatial coordinate system, but their approach is expensive relative to the other algorithms and it does not always maintain spherical symmetry, which is an important consideration in underwater explosion calculations.

The advection in LS-DYNA is performed isotropically. The fluxes through each face of element A are calculated simultaneously, but the values of ϕ in the transport volumes are calculated using the one-dimensional expressions developed in the previous sections.

$$\phi_A^{n+1} = \frac{1}{V_A^{n+1}} \left(V_A^n \phi_A^n + \sum_{j=1}^6 f_j^\phi \right). \quad (21.2.83)$$

The disadvantage of isotropic advection is that there is no coupling between an element and the elements that are joined to it only at its corners and edges (i.e., elements that don't share faces). The lack of coupling introduces a second order error that is significant only when the transport is along the mesh diagonals.

The one-dimensional MUSCL scheme, which requires elements on either side of the element whose transport is being calculated, cannot be used on the boundary elements in the direction normal to the boundary. Therefore, in the boundary elements, the donor cell algorithm is used to calculate the transport in the direction that is normal to the boundary, while the MUSCL scheme is used in the two tangential directions.

It is implicitly assumed by the transport calculations that the solution variables are defined per unit current volume. In LS-DYNA, some variables, such as the internal energy, are stored in terms of the initial volume of the element. These variables must be rescaled before transport, then the initial volume of the element is advected between the elements, and then the variables are rescaled using the new "initial" volumes. Hyperelastic materials are not currently advected in LS-DYNA because they require the deformation gradient, which is calculated from the initial geometry of the mesh. If the deformation gradient is integrated by using the midpoint rule, and it is advected with the

other solution variables, then hyperelastic materials can be advected without any difficulties.

$$F^{n+1} = (I - \frac{\Delta t}{2L^{n+1/2}})^{-1} \left(I + \frac{\Delta t}{2L^{n+1/2}} \right) F^n. \quad (21.2.84)$$

Advection of the Nodal Velocities. Except for the Godunov schemes, the velocity is centered at the nodes or the edges while the remaining variables are centered in the elements. Momentum is advected instead of the velocity in most codes to guarantee that momentum is conserved. The element-centered advection algorithms must be modified to advect the node-centered momentum. Similar difficulties are encountered when node-centered algorithms, such as the SUPG method [Brooks and Hughes 1982], are applied to element-centered quantities [Liu, Chang, and Belytschko, *to be published*]. There are two approaches: 1) construct a new mesh such that the nodes become the element centroids of the new mesh and apply the element-centered advection algorithms, and 2) construct an auxiliary set of element-centered variables from the momentum, advect them, and then reconstruct the new velocities from the auxiliary variables. Both approaches can be made to work well, but their efficiency is heavily dependent on the architecture of the codes. The algorithms are presented in detail for one dimension first for clarity. Their extensions to three dimensions, which are presented later, are straightforward even if the equations do become lengthy. A detailed discussion of the algorithms in two dimensions is presented in Reference [Benson 1992].

Notation. Finite difference notation is used in this section so that the relative locations of the nodes and fluxes are clear. The algorithms are readily applied, however, to unstructured meshes. To avoid limiting the discussion to a particular element-centered advection algorithm, the transport volume through node i is f , the transported mass is \tilde{f}_i , and the flux of ϕ is $\phi_i f_i$. Most of the element-centered flux-limited advection algorithms calculate the flux of ϕ directly, but the mean value of ϕ in the transport volumes is calculated by dividing the $\phi_i f_i$, by the transport volume. A superscript “-” or “+” denotes the value of a variable before or after the advection. Using this notation, the advection of ϕ in one dimension is represented by Equation (17.2.12), where the volume is V .

$$\phi_{j+1/2}^+ = \frac{(\phi_{j+1/2}^- V_{j+1/2}^- + \phi_i f_i - \phi_{i+1} f_{i+1})}{V^+}, \quad (21.2.85)$$

$$V_{j+1/2}^+ = V_{j+1/2}^- + f_i - f_{i+1}. \quad (21.2.86)$$

The Staggered Mesh Algorithm. YAQUI [Amsden and Hirt 1973] was the first code to construct a new mesh that is staggered with respect to original mesh for momentum advection. The new mesh is defined so that the original nodes become the centroids of the new elements. The element-centered advection algorithms are applied to the new mesh to advect the momentum. In theory, the momentum can be advected with the transport volumes or the velocity can be advected with the mass.

$$v_j^+ = \frac{(M_j^- v_j^- + v_{j-1/2} \tilde{f}_{j-1/2} - v_{j+1/2} \tilde{f}_{j+1/2})}{M_j^+}, \quad (21.2.87)$$

$$v_j^+ = \frac{(M_j^- v_j^- + \{\rho v\}_{j-1/2} f_{j-1/2} - \{\rho v\}_{j+1/2} f_{j+1/2})}{M_j^+}, \quad (21.2.88)$$

$$M_j^+ = M_j^- + \tilde{f}_{j-1/2} - \tilde{f}_{j+1/2}. \quad (21.2.89)$$

A consistency condition, first defined by DeBar [1974], imposes a constraint on the formulation of the staggered mesh algorithm: if a body has a uniform velocity and a spatially varying density before the advection, then the velocity should be uniform and unchanged after the advection. The new mass of a node can be expressed in terms of the quantities used to advect the element-centered mass.

$$M_j^+ = \frac{1}{2} (M_{j-1/2}^+ + M_{j+1/2}^+), \quad (21.2.90)$$

$$M_j^+ = \frac{1}{2} (M_{j-1/2}^- + \rho_{j-1} f_{j-1} - \rho_j f_j + M_{j+1/2}^- + \rho_j f_j - \rho_{j+1} f_{j+1}), \quad (21.2.91)$$

$$M_j^+ = M_j^- + \frac{1}{2} [(\rho_{j-1} f_{j-1} - \rho_j f_j) + (\rho_j f_j - \rho_{j+1} f_{j+1})]. \quad (21.2.92)$$

The staggered mass fluxes and transport volumes are defined by equating [Equation \(21.2.90\)](#) and Equation (17.2.15).

$$\rho_{j+1/2} f_{j+1/2} = \tilde{f}_{j+1/2} = \frac{1}{2} (\rho_j f_j + \rho_{j+1} f_{j+1}). \quad (21.2.93)$$

The density $\rho_{j+1/2}$ is generally a nonlinear function of the volume $f_{j+1/2}$, hence calculating $f_{j+1/2}$ from [Equation \(21.2.93\)](#) requires the solution of a nonlinear equation for each transport volume. In contrast, the mass flux is explicitly defined by [Equation \(21.2.93\)](#). Most codes, including KRAKEN [Debar 1974], CSQ [Thompson 1975], CTH [McGlaun 1989], and DYNA2D [Hallquist 1980], use mass fluxes with the staggered mesh algorithm because of their simplicity.

The dispersion characteristics of this algorithm are identical to the underlying element-centered algorithm by construction. This is not true, however, for some of the element-centered momentum advection algorithms. There are some difficulties in implementing the staggered mesh method in multi-dimensions. First, the number of edges defining a staggered element equals the number of elements surrounding the corresponding node. On an unstructured mesh, the arbitrary connectivity results in an arbitrary number of edges for each staggered element. Most of the higher order accurate advection algorithms assume a logically regular mesh of quadrilateral elements, making it difficult to use them with the staggered mesh. Vectorization also becomes difficult

because of the random number of edges that each staggered element might have. In the ALE calculations of DYNA2D, only the nodes that have a locally logically regular mesh surrounding them can be moved in order to avoid these difficulties [Benson 1992]. These difficulties do not occur in finite difference codes which process logically regular blocks of zones. Another criticism is the staggered mesh algorithm tends to smear out shocks because not all the advected variables are element-centered [Margolin 1989]. This is the primary reason, according to Margolin [1989], that the element-centered algorithm was adopted in SALE [Amdsden, Ruppel, and Hirt 1980].

The SALE Algorithm. SALE advects an element-centered momentum and redistributes its changes to the nodes [Amdsden, Ruppel, and Hirt 1980]. The mean element velocity, $\bar{\mathbf{v}}_{j+1/2}$, specific momentum, $\mathbf{p}_{j+1/2}$, element momentum, $\mathbf{P}_{j+1/2}$, and nodal momentum are defined by Equation (17.2.17).

$$\bar{\mathbf{v}}_{j+1/2} = \frac{1}{2}(\mathbf{v}_j + \mathbf{v}_{j+1}), \quad (21.2.94)$$

$$\mathbf{p}_{j+1/2} = \rho_{j+1/2} \bar{\mathbf{v}}_{j+1/2} \quad (21.2.95)$$

$$\mathbf{P}_{j+1/2} = M_{j+1/2} \bar{\mathbf{v}}_{j+1/2}. \quad (21.2.96)$$

Denoting the change in the element momentum $\Delta \mathbf{P}_{j+1/2}$, the change in the velocity at a node is calculated by distributing half the momentum change from the two adjacent elements.

$$\Delta \mathbf{P}_{j-1/2} = \mathbf{p}_{j-1} f_{j-1} - \mathbf{p}_j f_j, \quad (21.2.97)$$

$$\mathbf{P}_j^+ = \mathbf{P}_j^- + \frac{1}{2}(\Delta \mathbf{P}_{j-1/2} + \Delta \mathbf{P}_{j+1/2}), \quad (21.2.98)$$

$$\mathbf{v}_j^+ = \frac{\mathbf{P}_j^+}{M_j^+}. \quad (21.2.99)$$

This algorithm can also be implemented by advecting the mean velocity, $\bar{\mathbf{v}}_{j+1/2}$ with the transported mass, and the transported momentum $\mathbf{p}_j f_j$ is changed to $\bar{\mathbf{v}}_j \tilde{f}_j$.

The consistency condition is satisfied regardless of whether masses or volumes are used. Note that the velocity is not updated from the updated values of the adjacent element momenta. The reason for this is the original velocities are not recovered if $f_i = 0$, which indicates that there is an inversion error associated with the algorithm.

The HIS (Half Index Shift) Algorithm. Benson [1992] developed this algorithm based on his analysis of other element-centered advection algorithms. It is designed to overcome the dispersion errors of the SALE algorithm and to preserve the monotonicity of

the velocity field. The SALE algorithm is a special case of a general class of algorithms. To sketch the idea behind the HIS algorithm, the discussion is restricted to the scalar advection equation. Two variables, $\Psi_{1,j+1/2}$ and $\Psi_{2,j+1/2}$ are defined in terms of a linear transformation of ϕ_j and ϕ_{j+1} . The linear transformation may be a function of the element $j + 1/2$.

$$\begin{Bmatrix} \Psi_{1,j+1/2}^- \\ \Psi_{2,j+1/2}^- \end{Bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{Bmatrix} \phi_j^- \\ \phi_{j+1}^- \end{Bmatrix}. \quad (21.2.100)$$

This relation is readily inverted.

$$\begin{Bmatrix} \phi_j^+ \\ \phi_{j+1}^+ \end{Bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{Bmatrix} \Psi_{1,j+1/2}^+ \\ \Psi_{2,j+1/2}^+ \end{Bmatrix}. \quad (21.2.101)$$

A function is monotonic over an interval if its derivative does not change sign. The sum of two monotonic functions is monotonic, but their difference is not necessarily monotonic. As a consequence, $\Psi_{1,j+1/2}^-$ and $\Psi_{2,j+1/2}^-$ are monotonic over the same intervals as ϕ_j^- if all the coefficients in the linear transformation have the same sign. On the other hand, ϕ_j^+ is not necessarily monotonic even if $\Psi_{1,j+1/2}^+$ and $\Psi_{2,j+1/2}^+$ are monotonic because of the appearance of the negative signs in the inverse matrix. Monotonicity can be maintained by transforming in both directions provided that the transformation matrix is diagonal. Symmetry in the overall algorithm is obtained by using a weighted average of the values of ϕ_j calculated in elements $j + 1/2$ and $j - 1/2$.

A monotonic element-centered momentum advection algorithm is obtained by choosing the identity matrix for the transformation and by using mass weighting for the inverse relationship.

$$\begin{Bmatrix} \Psi_{1,j+1/2} \\ \Psi_{2,j+1/2} \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} v_j^- \\ v_{j+1}^- \end{Bmatrix} \quad (21.2.102)$$

To conserve momentum, Ψ is advected with the transport masses.

$$\Psi_{m,j+1/2}^+ = \frac{\left(M_{j+1/2}^- \Psi_{m,j+1/2}^- + \Psi_{m,j}^- \tilde{f}_j - \Psi_{m,j+1}^- \tilde{f}_{j+1} \right)}{M_{j+1/2}^+}, \quad (21.2.103)$$

$$v_j = \frac{1}{2M_j} (M_{j+1/2} \Psi_{1,j+1/2} + M_{j-1/2} \Psi_{2,j-1/2}). \quad (21.2.104)$$

Dispersion Errors. A von Neumann analysis [Trefethen 1982] characterizes the dispersion errors of linear advection algorithms. Since the momentum advection algorithm modifies the underlying element-centered advection algorithm, the momentum

advection algorithm does not necessarily have the same dispersion characteristics as the underlying algorithm. The von Neumann analysis provides a tool to explore the changes in the dispersion characteristics without considering a particular underlying advection algorithm.

The model problem is the linear advection equation with a constant value of c . A class of solutions can be expressed as complex exponentials, where i is $\sqrt{-1}$, ω is the frequency, and χ is the wave number.

$$\frac{\partial \varphi}{\partial t} + c \frac{\partial \varphi}{\partial x} = 0, \quad (21.2.105)$$

$$\varphi(x, t) = e^{i(\omega t - \chi x)}. \quad (21.2.106)$$

For Equation (17.2.24), the dispersion equation is $\omega = c\chi$, but for discrete approximations of the equation and for general hyperbolic equations, the relation is $\omega = \omega\chi$. The phase velocity, c_p , and the group speed, c_g , are defined by Equation (17.2.25).

$$c_p = \frac{\omega}{\chi}, \quad (21.2.107)$$

$$c_g = \frac{\partial \omega}{\partial \chi}. \quad (21.2.108)$$

The mesh spacing is assumed to have a constant value J , and the time step, h , is also constant. The + and - states in the previous discussions correspond to times n and $n + 1$ in the dispersion analysis. An explicit linear advection method that has the form given by Equation (1.2.109) results in a complex dispersion equation, Equation (17.2.27), where Π is a complex polynomial.

$$\varphi_j^{n+1} = \varphi_j^n + F(c, h, J, \dots, \varphi_{j-1}^n, \varphi_j^n, \varphi_{j+1}^n, \dots), \quad (21.2.109)$$

$$e^{i\omega h} = 1 + P(e^{i\chi J}), \quad (21.2.110)$$

$$\Pi(e^{i\chi J}) = \sum_j \beta_j e^{i\chi_j J}. \quad (21.2.111)$$

The dispersion equation has the general form given in Equation (1.2.112), where Π_r and Π_i denote the real and imaginary parts of Π , respectively.

$$\omega h = \tan^{-1} \left(\frac{\Pi_i}{1 + \Pi_r} \right). \quad (21.2.112)$$

Recognizing that the relations in the above equations are periodic in ωh and χ^J , the normalized frequency and wave number are defined to simplify the notation.

$$\bar{\omega} = \omega h, \quad \bar{\chi} = \chi^J. \quad (21.2.113)$$

The von Neumann analysis of the SALE algorithm proceeds by first calculating the increment in the cell momentum.

$$p_{j+1/2}^n = \frac{1}{2}(v_j^n + v_{j+1}^n), \quad (21.2.114)$$

$$p_{j+1/2}^n = \frac{1}{2}(1 + e^{-i\bar{\chi}})v_j^n, \quad (21.2.115)$$

$$\Delta p_{j+1/2}^{n+1} = p_{j+1/2}^{n+1} - p_{j+1/2}^n, \quad (21.2.116)$$

$$\Delta p_{j+1/2}^{n+1} = \frac{1}{2}(1 + e^{-i\bar{\chi}})\Pi v_j^n. \quad (21.2.117)$$

The velocity is updated from the changes in the cell momentum.

$$v_j^{n+1} = v_j^n + \frac{1}{2}(\Delta p_{j+1/2}^{n+1} + \Delta p_{j-1/2}^{n+1}), \quad (21.2.118)$$

$$v_j^{n+1} = \frac{1}{4}(1 + e^{i\bar{\chi}})(1 + e^{-i\bar{\chi}})\Pi v_j^n, \quad (21.2.119)$$

$$v_j^{n+1} = \frac{1}{2}(1 + \cos(\bar{\chi}))\Pi v_j^n. \quad (21.2.120)$$

The dispersion relation for the SALE advection algorithm is given by [Equation \(1.2.121\)](#).

$$\bar{\omega} = \tan^{-1} \left(\frac{\frac{1}{2}(1 + \cos(\bar{\chi}))\Pi_i}{1 + \frac{1}{2}(1 + \cos(\bar{\chi}))\Pi_r} \right). \quad (21.2.121)$$

By comparing [Equation \(21.2.112\)](#) and [Equation \(21.2.121\)](#), the effect of the SALE momentum advection algorithm on the dispersion is to introduce a factor λ , equal to $\frac{1}{2}(1 + \cos(\bar{\chi}))\Pi$, into the spatial part of the advection stencil. For small values of $\bar{\chi}$, λ is close to one, and the dispersion characteristics are not changed, but when $\bar{\chi}$ is π , the phase and group velocity go to zero and the amplification factor is one independent of the underlying advection algorithm. Not only is the wave not transported, it is not damped out. The same effect is found in two dimensions, where λ , has the form $\frac{1}{4}(1 + \cos(\bar{\chi}) + \cos(\bar{\bar{\chi}}) + \cos(\bar{\chi})\cos(\bar{\bar{\chi}}))$.

In contrast, none of the other algorithms alter the dispersion characteristics of the underlying algorithm. Benson has demonstrated for the element-centered algorithms that the SALE inversion error and the dispersion problem are linked. Algorithms that fall into the same general class as the SALE and HIS algorithms will, therefore, not have dispersion problems [Benson 1992].

Three-Dimensional Momentum Advection Algorithms. The momentum advection algorithms discussed in the previous sections are extended to three dimensions in a straightforward manner. The staggered mesh algorithm requires the construction of a staggered mesh and the appropriate transport masses. Based on the consistency arguments, the appropriate transport masses are given by [Equation \(1.2.122\)](#).

$$\tilde{f}_{j+1/2,k,l} = \frac{1}{8} \sum_{J=j}^{j+1} \sum_{K=k-1/2}^{k+1/2} \sum_{L=l-1/2}^{l+1/2} f_{J,K,L}. \quad (21.2.122)$$

The SALE advection algorithm calculates the average momentum of the element from the four velocities at the nodes and distributes 1/8 of the change in momentum to each node.

$$P_{j+1/2,k+1/2,l+1/2} = \frac{1}{8} \rho_{j+1/2,k+1/2,l+1/2} \sum_{J=j}^{j+1} \sum_{K=k}^{k+1} \sum_{L=l}^{l+1} v_{JKL}, \quad (21.2.123)$$

$$P_{j+1/2,k+1/2,l+1/2} = \frac{1}{8} M_{j+1/2,k+1/2,l+1/2} \sum_{J=j}^{j+1} \sum_{K=k}^{k+1} \sum_{L=l}^{l+1} v_{JKL}, \quad (21.2.124)$$

$$v_{j,k,l}^+ = \frac{1}{M_{j,k,l}^+} \left(M_{j,k,l}^- v_{j,k,l}^- + \frac{1}{8} \sum_{J=j-1/2}^{j+1/2} \sum_{K=k-1/2}^{k+1/2} \sum_{L=l-1/2}^{l+1/2} \Delta P_{J,K,L} \right). \quad (21.2.125)$$

The HIS algorithm is also readily extended to three dimensions. The variable definitions are given in [Equation \(1.2.126\)](#) and [Equation \(1.2.127\)](#), where the subscript A refers to the local numbering of the nodes in the element. In an unstructured mesh, the relative orientation of the nodal numbering within the elements may change. The subscript A is always with reference to the numbering in element j,k,l . The subscript \tilde{A} is the local node number in an adjacent element that refers to the same global node number as A.

$$\Psi_{A,j+1/2,k+1/2,l+1/2} = v_{A,j+1/2,k+1/2,l+1/2}, \quad (21.2.126)$$

$$v_{j,k,l}^+ = \frac{1}{M_{j,k,l}^+} \sum_{J=j-1/2}^{j+1/2} \sum_{K=k-1/2}^{k+1/2} \sum_{L=l-1/2}^{l+1/2} M_{J,K,L} \Psi_{\tilde{A},J,K,L}^+. \quad (21.2.127)$$

21.3 The Manual Rezone

The central limitation to the simplified ALE formulation is that the topology of the mesh is fixed. For a problem involving large deformations, a mesh that works well at early times may not work at late times regardless of how the mesh is distributed over the material domain. To circumvent this difficulty, a manual rezoning capability has been implemented in LS-DYNA. The general procedure is to 1) interrupt the calculation when the mesh is no longer acceptable, 2) generate a new mesh with INGRID by using the current material boundaries from LS-DYNA (the topologies of the new and old mesh are unrelated), 3) remap the solution from the old mesh to the new mesh, and 4) restart the calculation.

This chapter will concentrate on the remapping algorithm since the mesh generation capability is documented in the INGRID manual [Stillman and Hallquist 1992]. The remapping algorithm first constructs an interpolation function on the original mesh by using a least squares procedure, and then interpolates for the solution values on the new mesh.

The one point quadrature used in LS-DYNA implies a piecewise constant distribution of the solution variables within the elements. A piecewise constant distribution is not acceptable for a rezoner since it implies that for even moderately large changes in the locations of the nodes (say, displacements on the order of fifty percent of the elements characteristic lengths) that there will be no changes in the values of the element-centered solution variables. A least squares algorithm is used to generate values for the solution variables at the nodes from the element-centered values. The values of the solution variables can then be interpolated from the nodal values, ϕ_A , using the standard trilinear shape functions anywhere within the mesh.

$$\phi(\xi, \eta, \zeta) = \phi_A N_A(\xi, \eta, \zeta). \quad (21.3.128)$$

The objective function for minimization, J , is defined material by material, and each material is remapped independently.

$$J = \frac{1}{2} \int_V (\phi_A N_A - \phi)^2 dV. \quad (21.3.129)$$

The objective function is minimized by setting the derivatives of J with respect to ϕ_A equal to zero.

$$\frac{\partial J}{\partial \phi_A} = \int_V (\phi_B N_B - \phi) N_A dV = 0. \quad (21.3.130)$$

The least square values of ϕ_A are calculated by solving the system of linear equations, Equation (17.3.4).

$$M_{AB}\phi_B = \int_V N_A \phi dV, \quad (21.3.131)$$

$$M_{AB} = \int_V N_A N_B dV. \quad (21.3.132)$$

The “mass matrix”, M_{AB} , is lumped to give a diagonal matrix. This eliminates the spurious oscillations that occur in a least squares fit around the discontinuities in the solution (e.g., shock waves) and facilitates an explicit solution for ϕ_A . The integral on the right hand side of Equation (21.3.131) is evaluated using one point integration. By introducing these simplifications, Equation (17.3.4) is reduced to Equation (1.3.133), where the summation over a is restricted to the elements containing node A .

$$\phi_A = \frac{\sum_{\alpha} \phi_{\alpha} V_{\alpha}}{\sum_{\alpha} V_{\alpha}}. \quad (21.3.133)$$

The value of ϕ_{α} is the mean value of ϕ in element a . From this definition, the value of ϕ_{α} is calculated using Equation (1.3.134).

$$\phi_{\alpha} = \frac{1}{V_{\alpha}} \int_{V_{\alpha}} \phi_A N_A dV. \quad (21.3.134)$$

The integrand in Equation (21.3.134) is defined on the old mesh, so that Equation (21.3.134) is actually performed on the region of the old mesh that overlaps element α in the new mesh, where the superscript “*” refers to elements on the old mesh.

$$\phi_{\alpha} = \frac{1}{V_{\alpha}} \sum_{\beta}^* \int_{V_{\alpha} \cap V_{\beta}^*} \phi_A N_A dV^*. \quad (21.3.135)$$

One point integration is currently used to evaluate Equation (21.3.135), although it would be a trivial matter to add higher order integration. By introducing this simplification, Equation (21.3.135) reduces to interpolating the value of ϕ_{α} from the least squares fit on the old mesh.

$$\phi_a = \phi_A N_A(\xi^*, \eta^*, \zeta^*). \quad (21.3.136)$$

The isoparametric coordinates in the old mesh that correspond to the spatial location of the new element centroid must be calculated for Equation (21.3.136). The algorithm that is described here is from Shapiro [1990], who references [Thompson and Maffeo 1985, Maffeo 1984, Maffeo 1985] as the motivations for his current strategy, and we follow his notation. The algorithm uses a “coarse filter” and a “fine filter” to make the search for the correct element in the old mesh efficient.

The coarse filter calculates the minimum and maximum coordinates of each element in the old mesh. If the new element centroid, (x_s, y_s, z_s) , lies outside of the box

defined by the maximum and minimum values of an old element, then the old element does not contain the new element centroid.

Several elements may pass the coarse filter but only one of them contains the new centroid. The fine filter is used to determine if an element actually contains the new centroid. The fine filter algorithm will be explained in detail for the two-dimensional case since it is easier to visualize than the three-dimensional case, but the final equations will be given for the three-dimensional case.

The two edges adjacent to each node in [Figure 21.2](#) (taken from [Shapiro 1990]) define four skew coordinate systems. If the coordinates for the new centroid are positive for each coordinate system, then the new centroid is located within the old element. Because of the overlap of the four positive quarter spaces defined by the skew coordinate systems, only two coordinate systems actually have to be checked. Using the first and third coordinate systems, the coordinates, α_i , are the solution of Equation (17.3.9).

$$V_s = V_1 + a_1 V_{12} + a_2 V_{14}, \quad (21.3.137)$$

$$V_s = V_3 + a_3 V_{32} + a_4 V_{34}. \quad (21.3.138)$$

Two sets of linear equations are generated for the α_i by expanding the vector equations.

$$\begin{bmatrix} x_2 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_4 - y_1 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{Bmatrix} x_s - x_1 \\ y_s - y_1 \end{Bmatrix}, \quad (21.3.139)$$

$$\begin{bmatrix} x_2 - x_3 & x_4 - x_3 \\ y_2 - y_3 & y_4 - y_3 \end{bmatrix} \begin{Bmatrix} \alpha_3 \\ \alpha_4 \end{Bmatrix} = \begin{Bmatrix} x_s - x_3 \\ y_s - y_3 \end{Bmatrix}. \quad (21.3.140)$$

The generalization of Equation (17.3.9) to three dimensions is given by Equation (17.3.11), and it requires the solution of four sets of three equations. The numbering convention for the nodes in Equation (17.3.11) follows the standard numbering scheme used in LS-DYNA for eight node solid elements.

$$V_s = V_1 + a_1 V_{12} + a_2 V_{14} + a_3 V_{15}, \quad (21.3.141)$$

$$V_s = V_3 + a_4 V_{37} + a_5 V_{34} + a_6 V_{32}, \quad (21.3.142)$$

$$V_s = V_6 + a_7 V_{62} + a_8 V_{65} + a_9 V_{67}, \quad (21.3.143)$$

$$V_s = V_8 + a_{10} V_{85} + a_{11} V_{84} + a_{12} V_{87}. \quad (21.3.144)$$

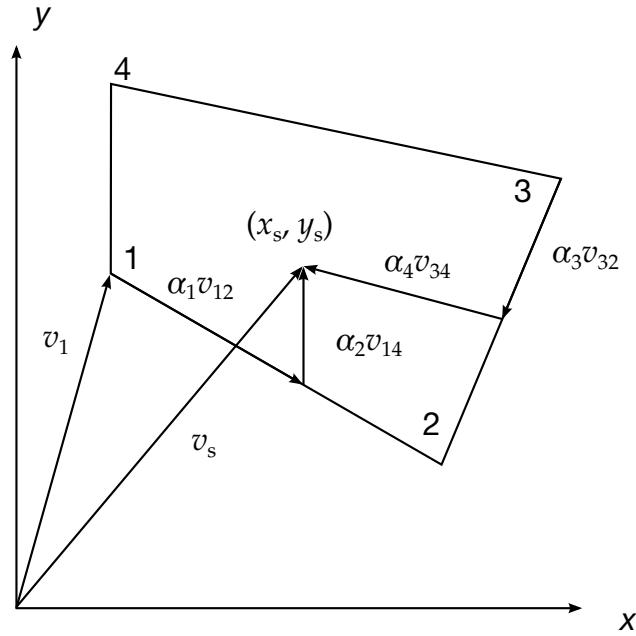


Figure 21.2. Skew Coordinate System

The fine filter sometimes fails to locate the correct element when the mesh is distorted. When this occurs, the element that is closest to the new centroid is located by finding the element for which the sum of the distances between the new centroid and the nodes of the element is a minimum.

Once the correct element is found, the isoparametric coordinates are calculated using the Newton-Raphson method, which usually converges in three or four iterations.

$$\begin{bmatrix} x_A \frac{\partial N_A}{\partial \xi} & x_A \frac{\partial N_A}{\partial \eta} & x_A \frac{\partial N_A}{\partial \zeta} \\ y_A \frac{\partial N_A}{\partial \xi} & y_A \frac{\partial N_A}{\partial \eta} & y_A \frac{\partial N_A}{\partial \zeta} \\ z_A \frac{\partial N_A}{\partial \xi} & z_A \frac{\partial N_A}{\partial \eta} & z_A \frac{\partial N_A}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \Delta \xi \\ \Delta \eta \\ \Delta \zeta \end{Bmatrix} = \begin{Bmatrix} x_s - x_A N_A \\ y_s - y_A N_A \\ z_s - z_A N_A \end{Bmatrix}, \quad (21.3.145)$$

$$\begin{aligned} \xi^{i+1} &= \xi^i + \Delta \xi, \\ \eta^{i+1} &= \eta^i + \Delta \eta, \\ \zeta^{i+1} &= \zeta^i + \Delta \zeta. \end{aligned} \quad (21.3.146)$$

22

Stress Update Overview

22.1 Jaumann Stress Rate

Stresses for material which exhibit elastic-plastic and soil-like behavior (hypoelastic) are integrated incrementally in time:

$$\sigma_{ij}(t + dt) = \sigma_{ij}(t) + \dot{\sigma}_{ij}dt, \quad (22.1)$$

Here, and in equations which follow, we neglect the contribution of the bulk viscosity to the stress tensor. In Equation (22.1), the dot denotes the material time derivative given by

$$\dot{\sigma}_{ij} = \sigma_{ij}^{\nabla} + \sigma_{ik}\omega_{kj} + \sigma_{jk}\omega_{ki}, \quad (22.2)$$

in which

$$\omega_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right), \quad (22.3)$$

is the spin tensor and

$$\sigma_{ij}^{\nabla} = C_{ijkl}\dot{\epsilon}_{kl}, \quad (22.4)$$

is the Jaumann (co-rotational) stress rate. In Equation (22.4), C_{ijkl} is the stress dependent constitutive matrix, v_i , is the velocity vector, and $\dot{\epsilon}_{ij}$ is the strain rate tensor:

$$\dot{\epsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \quad (22.5)$$

In the implementation of Equation (22.1) we first perform the stress rotation, Equation (22.2), and then call a constitutive subroutine to add the incremental stress components σ_{ij}^{∇} . This may be written as

$$\sigma_{ij}^{n+1} = \sigma_{ij}^n + r_{ij}^n + \sigma_{ij}^{\nabla n+1/2} \Delta t^{n+1/2}, \quad (22.6)$$

where

$$\begin{aligned}\sigma_{ij}^{n+1/2} \Delta t^{n+1/2} &= C_{ijkl} \Delta \varepsilon_{kl}^{n+1/2}, \\ \Delta \varepsilon_{ij}^{n+1/2} &= \dot{\varepsilon}_{ij}^{n+1/2} \Delta t^{n+1/2},\end{aligned}\tag{22.7}$$

and r_{ij}^n gives the rotation of the stress at time t^n to the configuration at t^{n+1}

$$r_{ij}^n = \left(\sigma_{ip}^n \omega_{pj}^{n+1/2} + \sigma_{jp}^n \omega_{pi}^{n+1/2} \right) \Delta t^{n+1/2}.\tag{22.8}$$

In the implicit NIKE2D/3D [Hallquist 1981b] codes, which are used for low frequency structural response, we do a half-step rotation, apply the constitutive law, and complete the second half-step rotation on the modified stress. This approach has also been adopted for some element formulations in LS-DYNA when the invariant stress update is active. An exact or second order accurate rotation is performed rather than the approximate one represented by Equation (22.3), which is valid only for small incremental rotations. A typical implicit time step is usually 100 to 1000 or more times larger than the explicit time step; consequently, the direct use of Equation (22.7) could lead to very significant errors.

22.2 Jaumann Stress Rate Used With Equations of State

If pressure is defined by an equation of state as a function of relative volume, V , and energy, E , or temperature, T ,

$$p = p(V, E) = p(V, T),\tag{22.9}$$

we update the deviatoric stress components

$$s_{ij}^{n+1} = \sigma_{ij}^n + r_{ij}^n + p^n \delta_{ij} + C_{ijkl} \dot{\varepsilon}_{kl}'^{n+1/2} \Delta t^{n+1/2},\tag{22.10}$$

where $\dot{\varepsilon}_{ij}'^{n+1/2}$ is the deviatoric strain rate tensor:

$$\dot{\varepsilon}_{ij}'^{n+1/2} = \dot{\varepsilon}_{ij} - \frac{1}{3} \dot{\varepsilon}_{kk} \delta_{ij}.\tag{22.11}$$

Before the equation of state, Equation (22.9), is evaluated, we compute the bulk viscosity, q , and update the total internal energy e of the element being processed to a trial value e^* :

$$e^{*n+1} = e^n - \frac{1}{2} \Delta v \left(p^n + q^{n-1/2} + q^{n+1/2} \right) + v^{n+1/2} s_{ij}^{n+1/2} \Delta \varepsilon_{ij}^{n+1/2},\tag{22.12}$$

where v is the element volume and

$$\Delta v = v^{n+1} - v^n, \quad v^{n+1/2} = \frac{1}{2}(v^n + v^{n+1}), \quad s_{ij}^{n+1/2} = \frac{1}{2}(s_{ij}^n + s_{ij}^{n+1}). \quad (22.13)$$

The time-centering of the viscosity is explained by Noh [1976].

Assume we have an equation of state that is linear in internal energy of the form

$$p^{n+1} = A^{n+1} + B^{n+1}E^{n+1}, \quad (22.14)$$

where

$$E^{n+1} = \frac{e^{n+1}}{v_0}, \quad (22.15)$$

and v_0 is the initial volume of the element. Noting that

$$e^{n+1} = e^{*n+1} - \frac{1}{2}\Delta v p^{n+1}, \quad (22.16)$$

pressure can be evaluated exactly by solving the implicit form

$$p^{n+1} = \frac{(A^{n+1} + B^{n+1}E^{*n+1})}{\left(1 + \frac{1}{2}B^{n+1}\frac{\Delta v}{v_0}\right)}, \quad (22.17)$$

and the internal energy can be updated in [Equation \(22.16\)](#). If the equation of state is not linear in internal energy, a one-step iteration is used to approximate the pressure

$$p^{*n+1} = p(V^{n+1}, E^{*n+1}). \quad (22.18)$$

Internal energy is updated to $n + 1$ using p^{*n+1} in [Equation \(22.16\)](#) and the final pressure is then computed:

$$p^{n+1} = p(V^{n+1}, E^{n+1}). \quad (22.19)$$

This is also the iteration procedure used in KOVEC [Woodruff 1973]. All the equations of state in LS-DYNA are linear in energy except the ratio of polynomials.

22.3 Green-Naghdi Stress Rate

The Green-Naghdi rate is defined as

$$\sigma_{ij}^{\nabla} = \dot{\sigma}_{ij} + \sigma_{ik}\Omega_{kj} + \sigma_{jk}\Omega_{ki} = R_{ik}R_{jl}\dot{\tau}_{kl}, \quad (22.20)$$

where Ω_{ij} is defined as

$$\Omega_{ij} = \dot{R}_{ik}R_{jk}, \quad (22.21)$$

and \mathbf{R} is found by application of the polar decomposition theorem

$$F_{ij} = R_{ik}U_{kj} = V_{ik}R_{kj}, \quad (22.22)$$

F_{ij} is the deformation gradient matrix and U_{ij} and V_{ij} are the positive definite right and left stretch tensors:

$$F_{ij} = \frac{\partial x_i}{\partial X_j}. \quad (22.23)$$

Stresses are updated for all materials by adding to the rotated Cauchy stress at time n .

$$\tau_{ij}^n = R_{ki}^n R_{lj}^n \sigma_{kl}^n, \quad (22.24)$$

the stress increment obtained by an evaluation of the constitutive equations,

$$\Delta \tau_{ij}^{n+1/2} = C_{ijkl} \Delta d_{kl}^{n+1/2}, \quad (22.25)$$

where

$$\Delta d_{ij}^{n+1/2} = R_{ki}^{n+1/2} R_{lj}^{n+1/2} \Delta \varepsilon_{kl}^{n+1/2} \quad (22.26)$$

C_{ijkl} = constitutive matrix

$\Delta \varepsilon_{kl}$ = increment in strain

and to obtain the rotated Cauchy stress at t_{n+1} , i.e.,

$$\tau_{ij}^{n+1} = \tau_{ij}^n + \Delta \tau_{ij}^{n+1/2}. \quad (22.27)$$

The desired Cauchy stress at $n + 1$ can now be found

$$\sigma_{ij}^{n+1} = R_{ik}^{n+1} R_{jl}^{n+1} \tau_{kl}^{n+1}. \quad (22.28)$$

Because we evaluate our constitutive models in the rotated configuration, we avoid the need to transform history variables such as the back stress that arises in kinematic hardening.

In the computation of \mathbf{R} , Taylor and Flanagan [1989] did an incremental update in contrast with the direct polar decomposition used in the NIKE3D code. Following their notation the algorithm is given by.

$$\begin{aligned}
z_i &= e_{ijk} V_{jm} \dot{\epsilon}_{mk}, \\
\omega_i &= e_{ijk} \omega_{jk} - 2[V_{ij} - \delta_{ij} V_{kk}]^{-1} z_j, \\
\Omega_{ij} &= \frac{1}{2} e_{ijk} \omega_k, \\
\left(\delta_{ik} - \frac{\Delta t}{2} \Omega_{ik} \right) R_{kj}^{n+1} &= \left(\delta_{ik} + \frac{\Delta t}{2} \Omega_{ik} \right) R_{kj}^n, \\
\dot{V}_{ij} &= (\dot{\epsilon}_{ik} + \omega_{ik}) V_{kj} - V_{ik} \Omega_{kj}, \\
V_{ij}^{n+1} &= V_{ij}^n + \Delta t \dot{V}_{ij}.
\end{aligned} \tag{22.29}$$

We have adopted the PRONTO3D approach in LS-DYNA due to numerical difficulties with the polar decomposition in NIKE3D. We believe the PRONTO3D approach is reliable. Several disadvantages of the PRONTO3D approach include 300+ operations (at least that is the number we got), the requirement of 15 additional variables per integration point, and if we rezone the material in the future the initial geometry will need to be remapped and the 15 variables initialized.

22.4 Elastoplastic Materials

At low stress levels in elastoplastic materials the stresses, σ_{ij} , depends only on the state of strain; however, above a certain stress level, called the yield stress, $\sigma_y(a_i)$,

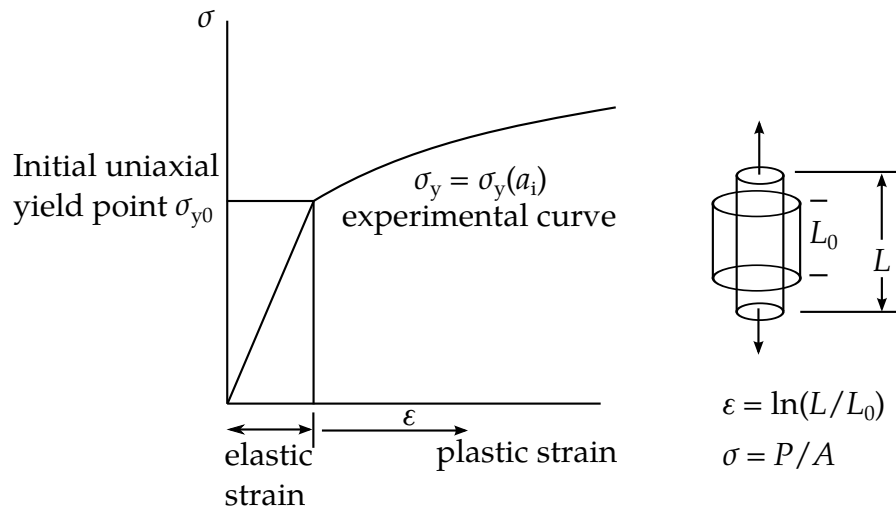


Figure 22.1. The uniaxial tension test demonstrates plastic behavior.

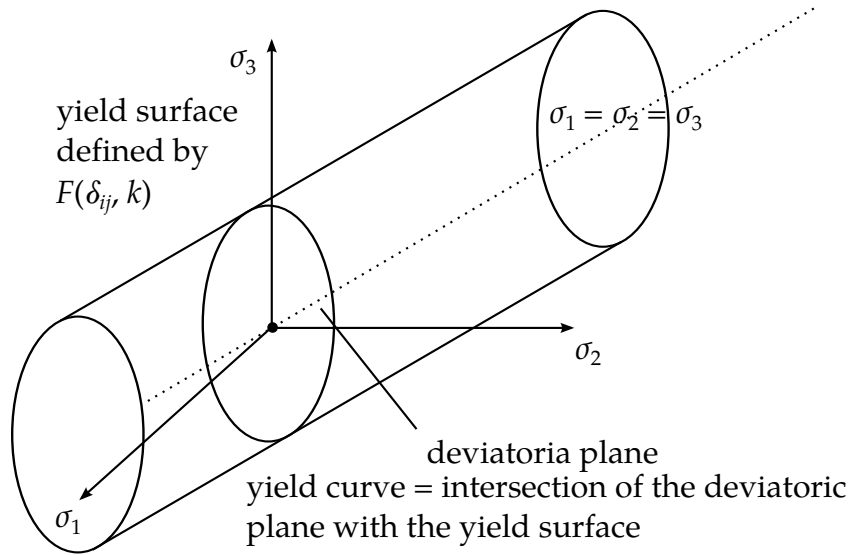


Figure 22.2. The yield surface in principal stress space in pressure independent.

nonrecoverable plastic deformations are obtained. The yield stress changes with increasing plastic deformations, which are measured by internal variables, a_i .

In the uniaxial tension test, a curve like that in [Figure 22.1](#) is generated where logarithmic uniaxial strain is plotted against the uniaxial true stress which is defined as the applied load P divided by the actual cross-sectional area, A .

For the simple von Mises plasticity models the yield stress is pressure independent and the yield surface is a cylinder in principal stress space as shown in [Figure 22.2](#). With isotropic hardening the diameter of the cylinder grows but the shape remains circular. In kinematic hardening the diameter may remain constant but will translate in the plane as a function of the plastic strain tensor, See [Figure 22.3](#).

The equation describing the pressure independent yield surface, F , is a function of the deviatoric stress tensor, s_{ij} , of a form given in [Equation \(1.30\)](#).

$$F(s_{ij}, a_i) = f(s_{ij}) - \sigma_y(a_i) = 0, \quad (22.30)$$

$f(s_{ij})$ = determines the shape,

$\sigma_y(a_i)$ = determines the translation and size.

The existence of a potential function, g , called the plastic potential, is assumed

$$g = g(s_{ij}). \quad (22.31)$$

Stability and uniqueness require that:

$$d\epsilon_{ij}^P = \lambda \frac{\partial g}{\partial s_{ij}}, \quad (22.32)$$

where λ is a proportionality constant.

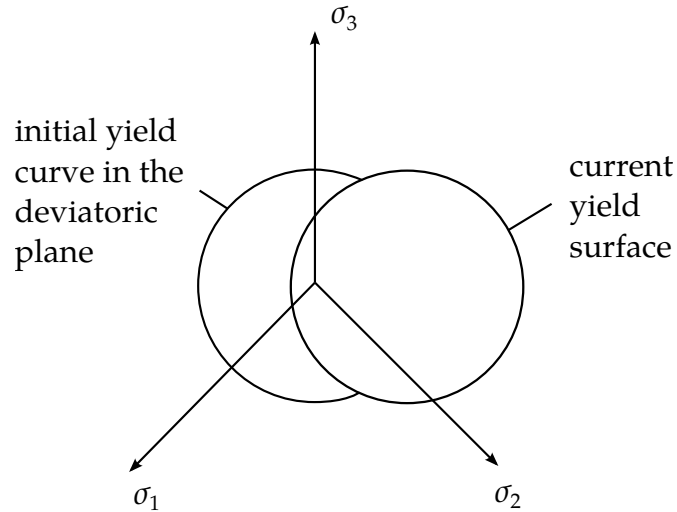


Figure 22.3. With kinematic hardening the yield surface may shift as a function of plastic strain.

As depicted in [Figure 22.5](#) the plastic strain increments $d\epsilon_{ij}^p$ are normal to the plastic potential function. This is the normality rule of plasticity.

The plastic potential g is identical with the yield condition $F(s_{ij})$

$$g \equiv F. \quad (22.33)$$

Hence:

$$d\epsilon_{ij}^p = \lambda \frac{\partial f}{\partial s_{ij}} = \lambda \text{ grad} f \quad (22.34)$$

and the stress increments ds_{ij} are normal to the plastic flow $\frac{\partial g}{\partial s_{ij}}$.

Post-yielding behavior from uniaxial tension tests typically show the following behaviors illustrated in [Figure 22.4](#):

The behavior of these hardening laws are characterized in [Table 1.1](#). below. Although LS-DYNA permits softening to be defined and used, such softening behavior will result in strain localization and nonconvergence with mesh refinement.

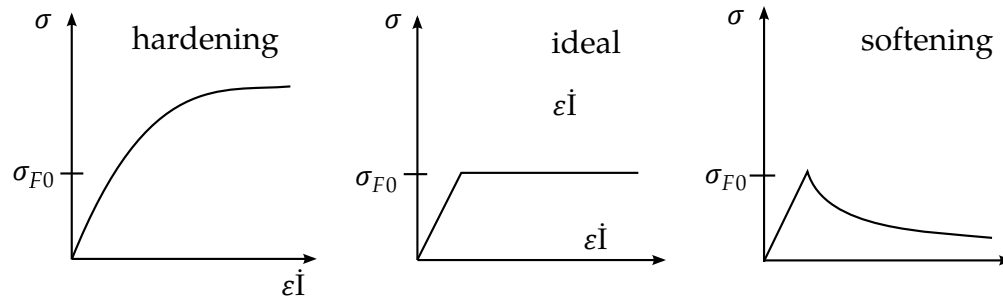


Figure 22.4. Hardening, ideal, and softening plasticity models.

	Hardening	Ideal	Softening
Behavior	$\sigma_y(a_i)$ is monotonic increasing	$\sigma_y(a_i)$ is constant	$\sigma_y(a_i)$ is monotonic decreasing
Stability	yes	yes	No
Uniqueness	yes	yes	No
Applications	metals, concrete, rock with small deformations	crude idealization for steel, plastics, etc.	dense sand, concrete with large deformations

Table 22.1. Plastic hardening, ideal plasticity, and softening.

22.5 Hyperelastic Materials

Stresses for elastic and hyperelastic materials are path independent; consequently, the stress update is not computed incrementally. The methods described here are well known and the reader is referred to Green and Adkins [1970] and Ogden [1984] for more details.

A rectangular cartesian coordinate system is used so that the covariant and contravariant metric tensors in the reference (undeformed) and deformed configuration are:

$$\begin{aligned}
 g_{ij} &= g^{ij} = \delta_{ij}, \\
 G_{ij} &= \frac{\partial x_k}{\partial X_i} \frac{\partial x_k}{\partial X_j}, \\
 G^{ij} &= \frac{\partial X_i}{\partial x_k} \frac{\partial X_j}{\partial x_k}.
 \end{aligned}
 \tag{22.35}$$

The Green-St. Venant strain tensor and the principal strain invariants are defined as

$$\gamma_{ij} = \frac{1}{2}(G_{ij} - \delta_{ij}), \quad (22.36)$$

$$\begin{aligned} I_1 &= \delta^{ij} G_{ij}, \\ I_2 &= \frac{1}{2}(\delta^{ir} \delta^{js} G_{ri} G_{sj} - \delta^{ir} \delta^{js} G_{ij} G_{rs}), \\ I_3 &= \det(G_{ij}), \end{aligned} \quad (22.37)$$

For a compressible elastic material the existence of a strain energy functional, W , is assumed

$$W = W(I_1, I_2, I_3), \quad (22.38)$$

which defines the energy per unit undeformed volume. The stress measured in the deformed configuration is given as [Green and Adkins, 1970]:

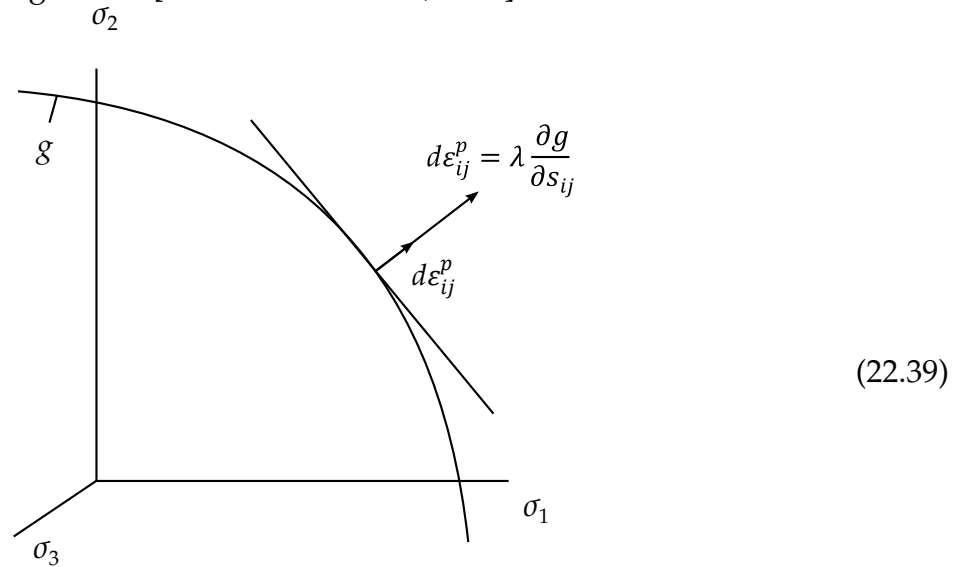


Figure 22.5. The plastic strain is normal to the yield surface.

$$s^{ij} = \Phi g^{ij} + \Psi B^{ij} + p G^{ij},$$

where

$$\begin{aligned} \Phi &= \frac{2}{\sqrt{I_3}} \frac{\partial W}{\partial I_1}, \\ \Psi &= \frac{2}{\sqrt{I_3}} \frac{\partial W}{\partial I_2}, \\ p &= 2\sqrt{I_3} \frac{\partial W}{\partial I_3}, \\ B^{ij} &= I_1 \delta^{ij} - \delta^{ir} \delta^{js} G_{rs}. \end{aligned} \quad (22.40)$$

This stress is related to the second Piola-Kirchhoff stress tensor:

$$S^{ij} = s^{ij} \sqrt{I_3}. \quad (22.41)$$

Second Piola-Kirchhoff stresses are transformed to physical (Cauchy) stresses according to the relationship:

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl}. \quad (22.42)$$

22.6 Layered Composites

The composite models for shell elements in LS-DYNA include models for elastic behavior and inelastic behavior. The approach used here for updating the stresses also applies to the airbag fabric model.

To allow for an arbitrary orientation of the shell elements within the finite element mesh, each ply in the composite has a unique orientation angle, β , which measures the offset from some reference in the element. Each integration point through the shell thickness, typically though not limited to one point per ply, requires the definition of β at that point. The reference is determined by the angle Ψ , which can be defined for each element on the element card, and is measured from the 1-2 element side. [Figures 22.6 and 22.7](#) depict these angles.

We update the stresses in the shell in the local shell coordinate system which is defined by the 1-2 element side and the cross product of the diagonals. Thus to transform the stress tensor into local system determined by the fiber directions entails a transformation that takes place in the plane of the shell.

In the implementation of the material model we first transform the Cauchy stress and velocity strain tensor d_{ij} into the coordinate system of the material denoted by the subscript L

$$\begin{aligned} \sigma_L &= \mathbf{q}^T \sigma \mathbf{q}, \\ \mathbf{q}_L &= \mathbf{q}^T \mathbf{d} \mathbf{q}, \\ \sigma_L &= \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{32} & \sigma_{32} & \sigma_{33} \end{bmatrix}, \\ \varepsilon_L &= \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{32} & d_{32} & d_{33} \end{bmatrix}, \end{aligned} \quad (22.43)$$

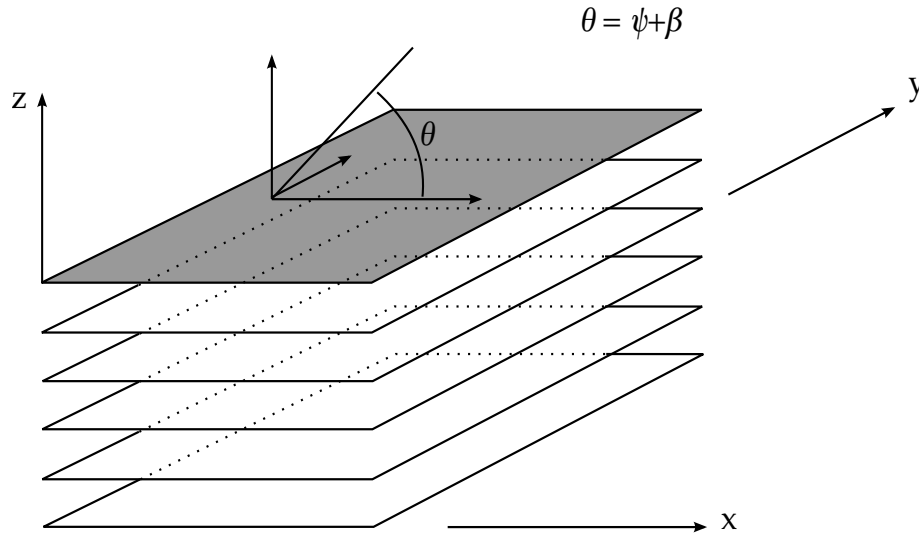


Figure 22.7. A multi-layer laminate can be defined. The angle β_i is defined for the i th lamina.

The Arabic subscripts on the stress and strain (σ and ϵ) are used to indicate the principal material directions where 1 indicates the fiber direction and 2 indicates the transverse fiber direction (in the plane). The orthogonal 3×3 transformation matrix is given by

$$\mathbf{q} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (22.44)$$

In shell theory we assume a plane stress condition, i.e., that the normal stress, σ_{33} , to the mid-surface is zero. We can now incrementally update the stress state in the material coordinates

$$\sigma_L^{n+1} = \sigma_L^n + \Delta\sigma_L^{n+1/2}, \quad (22.45)$$

where for an elastic material

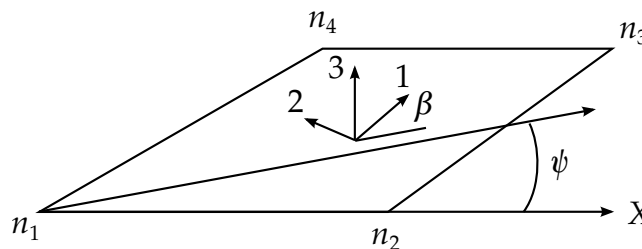


Figure 22.6. Orientation of material directions relative to the 1-2 side.

$$\Delta \sigma_L^{n+1/2} = \begin{bmatrix} \Delta \sigma_{11} \\ \Delta \sigma_{22} \\ \Delta \sigma_{12} \\ \Delta \sigma_{23} \\ \Delta \sigma_{31} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 & 0 & 0 \\ Q_{12} & Q_{22} & 0 & 0 & 0 \\ 0 & 0 & Q_{44} & 0 & 0 \\ 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} d_{11} \\ d_{22} \\ d_{12} \\ d_{23} \\ d_{31} \end{bmatrix}_L \Delta t. \quad (22.46)$$

The terms Q_{ij} are referred to as reduced components of the lamina and are defined as

$$\begin{aligned} Q_{11} &= \frac{E_{11}}{1 - \nu_{12}\nu_{21}}, \\ Q_{22} &= \frac{E_{22}}{1 - \nu_{12}\nu_{21}}, \\ Q_{12} &= \frac{\nu_{12}E_{11}}{1 - \nu_{12}\nu_{21}}, \\ Q_{44} &= G_{12}, \\ Q_{55} &= G_{23}, \\ Q_{66} &= G_{31}. \end{aligned} \quad (22.47)$$

Because of the symmetry properties,

$$\nu_{ji} = \nu_{ij} \frac{E_{jj}}{E_{ii}}, \quad (22.48)$$

where ν_{ij} is Poisson's ratio for the transverse strain in j th direction for the material undergoing stress in the i th-direction, E_{ij} are the Young's moduli in the i th direction, and G_{ij} are the shear moduli.

After completion of the stress update we transform the stresses back into the local shell coordinate system.

$$\sigma = \mathbf{q} \sigma_L \mathbf{q}^T. \quad (22.49)$$

22.7 Constraints on Orthotropic Elastic Constants

The inverse of the constitutive matrix \mathbf{C}_l is generally defined in terms of the local material axes for an orthotropic material is given in terms of the nine independent elastic constants as

$$\mathbf{C}_1^{-1} = \begin{bmatrix} \frac{1}{E_{11}} & -\frac{\nu_{21}}{E_{22}} & -\frac{\nu_{31}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_{11}} & \frac{1}{E_{22}} & -\frac{\nu_{32}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_{11}} & -\frac{\nu_{23}}{E_{22}} & \frac{1}{E_{33}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix}. \quad (22.50)$$

As discussed by Jones [1975], the constants for a thermodynamically stable material must satisfy the following inequalities:

$$\begin{aligned} E_1, E_2, E_3, G_{12}, G_{23}, G_{31} &> 0, \\ C_{11}, C_{22}, C_{33}, C_{44}, C_{55}, C_{66} &> 0, \\ (1 - \nu_{23}\nu_{32}), (1 - \nu_{13}\nu_{31}), (1 - \nu_{12}\nu_{21}) &> 0, \\ 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13} &> 0. \end{aligned} \quad (22.51)$$

Using Equation (22.48) and (22.51) leads to:

$$\begin{aligned} |\nu_{21}| &< \left(\frac{E_{22}}{E_{11}}\right)^{1/2} & |\nu_{12}| &< \left(\frac{E_{11}}{E_{22}}\right)^{1/2}, \\ |\nu_{32}| &< \left(\frac{E_{33}}{E_{22}}\right)^{1/2} & |\nu_{23}| &< \left(\frac{E_{22}}{E_{33}}\right)^{1/2}, \\ |\nu_{13}| &< \left(\frac{E_{11}}{E_{33}}\right)^{1/2} & |\nu_{31}| &< \left(\frac{E_{33}}{E_{11}}\right)^{1/2}. \end{aligned} \quad (22.52)$$

22.8 Local Material Coordinate Systems in Solid Elements

In solid elements there is a number of different ways of defining a local coordinate system. Perhaps the most general is by defining a triad for each element that is oriented in the local material directions, See Figure 22.8. In this approach two vectors **a** and **d** are defined. The local **c** direction is found from the cross product, **c** = **a** × **d**, the local **b** direction is the cross product **b** = **c** × **a**. This triad is stored as history data at each integration point.

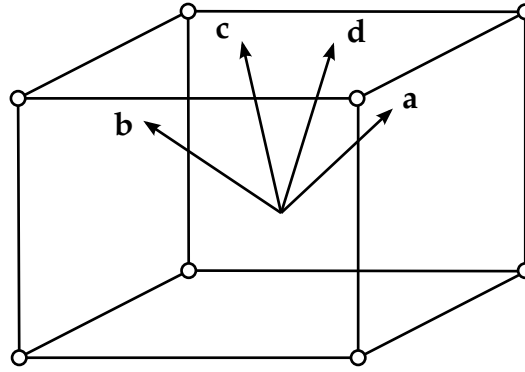


Figure 22.8. Local material directions are defined by a triad which can be input for each solid element.

The biggest concern when dealing with local material directions is that the results are not invariant with element numbering since the orientation of the local triad is fixed with respect to the base of the brick element, nodes 1-4, in [Figure 22.9](#). For Hyperelastic materials where the stress tensor is computed in the initial configuration, this is not a problem, but for materials like the honeycomb foams, the local directions can change due to element distortion causing relative movement of nodes 1-4. In honeycomb foams we assume that the material directions are orthogonal in the deformed configuration since the stress update is performed in the deformed configuration.

22.9 General Erosion Criteria for Solid Elements

Several erosion criteria are available that are independent of the material models. Each one is applied independently, and once any one of them is satisfied, the element is deleted from the calculation. The criteria for failure are:

- $P \geq P_{\min}$ where P is the pressure (positive in compression), and P_{\min} is the pressure at failure.
- $\sigma_1 \geq \sigma_{\max}$, where σ_1 is the maximum principal stress, and σ_{\max} is the principal stress at failure.
- $\sqrt{\frac{3}{2} \sigma'_{ij} \sigma'_{ij}} \geq \bar{\sigma}_{\max}$, where σ'_{ij} are the deviatoric stress components, and $\bar{\sigma}_{\max}$ is the equivalent stress at failure.
- $\epsilon_1 \geq \epsilon_{\max}$, where ϵ_1 is the maximum principal strain, and ϵ_{\max} is the principal strain at failure.
- $\gamma_1 \geq \gamma_{\max}$, where γ_1 is the shear strain, and γ_{\max} is the shear strain at failure.
- The Tuler-Butcher criterion,

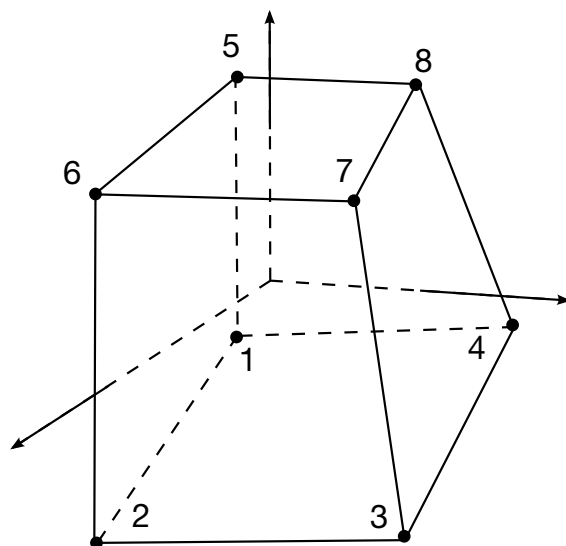


Figure 22.9. The orientation of the triad for the local material directions is stored relative to the base of the solid element. The base is defined by nodes 1-4 of the element connectivity.

$$\int_0^t [\max(0, \sigma_1 - \sigma_0)]^2 dt \geq K_f, \quad (22.53)$$

where σ_1 is the maximum principal stress, σ_0 is a specified threshold stress, $\sigma_1 \geq \sigma_0 \geq 0$, and K_f is the stress impulse for failure. Stress values below the threshold value are too low to cause fracture even for very long duration loadings. Typical constants are given in [Table 1.2](#) below [Rajendran, 1989].

These failure models apply to solid elements with one point integration in 2 and 3 dimensions.

Material	σ_0 (Kbar)		
1020 Steel	10.0	2	12.5
OFHC Copper	3.60	2	10.0
C1008	14.0	2	0.38
HY100	15.7	2	61.0
7039-T64	8.60	2	3.00

Table 22.2. Typical constants for the Tuler-Bucher criterion.

22.10 Strain Output to the LS-DYNA Database

The strain tensors that are output to the LS-DYNA database from the solid, shell, and beam elements are integrated in time. These strains are similar to the logarithmic strain measure and are based on an integration of the strain rate tensor. Admittedly, the

shear strain components do not integrate as logarithmic strain measures, but in spite of this, we have found that the strains output from LS-DYNA are far more useful than those computed in LS-DYNA. The time integration of the strain tensor in LS-DYNA maintains objectivity in the sense that rigid body motions do not cause spurious straining.

Recall, the spin tensor and strain rate tensor, [Equations \(22.3\)](#) and [\(22.5\)](#), respectively:

$$\omega_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right), \quad (22.54)$$

$$\dot{\epsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \quad (22.55)$$

In updating the strains from time n to $n + 1$, the following formula is used:

$$\epsilon_{ij}^{n+1} = \epsilon_{ij}^n + \rho_{ij}^n + \dot{\epsilon}_{ij}^{n+1/2} \Delta t^{n+1/2}, \quad (22.56)$$

where ρ_{ij}^n gives the rotational correction that transforms the strain tensor at time t^n into the configuration at t^{n+1}

$$\rho_{ij}^n = \left(\epsilon_{ip}^n \omega_{pj}^{n+1/2} + \epsilon_{jp}^n \omega_{pi}^{n+1/2} \right) \Delta t^{n+1/2}. \quad (22.57)$$

For shell elements we integrate the strain tensor at the inner and outer integration points and output two tensors per element. When the mid surface strains are plotted in LS-PREPOST, these are the average values.

22.12 Strain Rate Effects in Material Models

In a constitutive modeling context, the term *rate effects* is used to indicate that the material response depends on the (time) rate of a certain quantity, here called e for the sake of generality. This quantity is usually some kind of strain measure, and an example is when the stress depends on the rate of strain. In explicit simulations, high frequency strain content yields a very noisy strain rate and calls for some kind of smoothing before being used in the stress calculations. This section presents three of the common ways to do this averaging and that are used frequently in the material models.

22.12.1 Running N-average option

One option is to do a running average of current and previous strain rates. With this option the rate \dot{e} of a quantity e is averaged according to the following algorithm

$$\dot{\bar{e}} = \text{unfiltered rate (raw data)} \quad (22.1)$$

$$\dot{e}_n = \frac{\dot{\bar{e}} + \sum_{i=n-N+1}^{n-1} \dot{e}_i}{N} \quad (22.2)$$

$$\dot{e} = \dot{e}_n = \text{rate used in material routine} \quad (22.3)$$

In these equations, the subscript denotes stored history variables necessary for computing the running average strain rates and n denotes the current cycle number. This requires storage of $N - 1$ history variables, for most materials $N = 12$.

22.12.2 Last N-average option

The second option is to compute the rate as the average of the last N computed rates. Here the rate is evaluated according to

$$\dot{\bar{e}} = \text{unfiltered rate (raw data)} \quad (22.4)$$

$$\dot{e}_n = \dot{\bar{e}} \quad (22.5)$$

$$\dot{e} = \frac{\sum_{i=n-N+1}^n \dot{e}_i}{N} = \text{rate used in material routine} \quad (22.6)$$

As for the running average option, the subscript denotes stored history variables necessary for computing the running average strain rates and n is the cycle number. This also requires storage of $N - 1$ history variables, for most materials $N = 12$.

22.12.3 Averaging over a fixed amount of time T

This option was introduced in an attempt to suppress the time step dependence as much as possible. With this option we use N history variables to store the approximate values of the quantity of interest from the time $t_n - T$ to current time t_n . That is, we set

$$e_{n-i} = e^n \left(t_n - \frac{iT}{N-1} \right), i = 0, \dots, N-1 \quad (22.7)$$

Here $e^n(t)$ is an approximate function of the function of interest, $e(t)$, and will be given more specifically below. Assuming $e^n(t)$ is known, the rate used in the material routine in cycle n is simply

$$\dot{e} = \frac{e_n - e_{n-N+1}}{T} = \text{rate used in material routine} \quad (22.8)$$

For the update of history variables we assume that the N points in (22.7) together with the newly calculated quantity (raw data) at $t_{n+1} = t + \Delta t$,

$$e_{n+1} = e(t_{n+1}), \quad (22.9)$$

completely define the function $e^n(t)$ from time $t_n - T$ to time t_{n+1} by linear interpolation between each control point. That is, the function $e^n(t)$ is extended to time t_{n+1} by the new value,

$$e^n(t_{n+1}) = e_{n+1}. \quad (22.10)$$

This function is illustrated in the figure below by the dashed line and red control points. The updated set of history variables are

$$e_{n+1-i} = e^n \left(t_{n+1} - \frac{iT}{N-1} \right), i = 0, \dots, N-1, \quad (22.11)$$

and these updated points define a new approximate function $e^{n+1}(t)$, again by linear interpolation between the control points. In Figure 22.1, the updated function is illustrated by the solid line and blue control points, note that the last point is in fact both red and blue since $e^n(t_{n+1}) = e^{n+1}(t_{n+1}) = e_{n+1}$. In sum, the function used for calculated the effective rate is completely redefined between steps.

For a time step Δt that is greater than or approximately equal to $T/(N-1)$, this option is more or less exactly the average rate from $t - T$ to t . For smaller time steps (which is usually the case in explicit analysis) the previous values are in practice smoothed out since only N variables are available for storing this time history. That is, the high frequency content of the history is lost and this could be an interesting alternative to the previous two options that are otherwise commonly used for these types of situations.

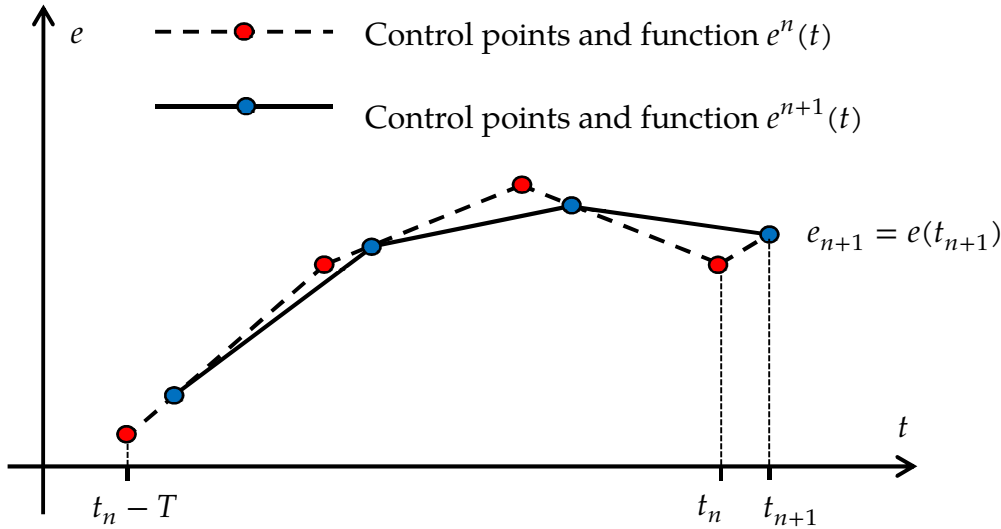


Figure 22.1. Illustration of control point update for $N = 4$

22.13 Algorithmic Consistent Tangent Modulus for Plasticity

For materials used in implicit analysis, the tangent modulus is needed for global assembly of the stiffness matrix. In particular, the softening effect in elastic-plastic materials must be predicted when in a plastic state. The following is a derivation of the tangent modulus that is consistent with the algorithmic stress update, meaning that it provides the exact variation of the stress with respect to the rate of deformation. We restrict ourselves to the plane stress update, as the full 3D situation is less complicated.

As a prerequisite, the algorithmic stress update must be established mathematically. For this, and the rest of this section, we introduce the variables

$$\begin{aligned}\Delta\sigma & - \text{stress increment, excluding through thickness stress } \Delta\sigma_3 \\ \Delta\varepsilon_3 & - \text{through thickness strain increment} \\ \Delta\varepsilon_p & - \text{plastic strain increment} \\ \Delta\alpha & - \text{back stress increment, full tensor}\end{aligned}$$

to be sought, the parameters

$$\begin{aligned}C & - 3D \text{ elastic tensor} \\ \Delta\varepsilon & - \text{strain increment, excluding through thickness strain } \Delta\varepsilon_3 \\ P & - \text{projection from 3D to plane stress space} \\ p^T & - \text{projection from 3D to out of plane space}\end{aligned}$$

that are given, and the functions

$$\begin{aligned}f(P^T\Delta\sigma - \Delta\alpha, \Delta\varepsilon_p) & - \text{yield function} \\ g(P^T\Delta\sigma - \Delta\alpha, \Delta\varepsilon_p) & - \text{plastic flow function} \\ h(P^T\Delta\sigma - \Delta\alpha, \Delta\varepsilon_p) & - \text{back stress potential}\end{aligned}$$

that define the plasticity. This theory thus incorporates full anisotropy as well as non-associative plasticity. The stress update can be summarized by the combination of four equations; the stress update

$$\Delta\sigma = PC \left(P^T\Delta\varepsilon + p\Delta\varepsilon_3 - \left\{ \frac{\partial g}{\partial \sigma} \right\}^T \Delta\varepsilon_p \right), \quad (22.12)$$

the yield condition

$$f = \sigma \left(P^T(\sigma_0 + \Delta\sigma) - (\alpha_0 + \Delta\alpha) \right) - \sigma_Y(\varepsilon_0 + \Delta\varepsilon_p) = 0, \quad (22.13)$$

the evolution of back stress

$$\Delta\alpha = \left\{ \frac{\partial h}{\partial \sigma} \right\}^T \Delta\varepsilon_p, \quad (22.14)$$

and the plane stress condition

$$\Delta\sigma_3 = p^T C \left(P^T\Delta\varepsilon + p\Delta\varepsilon_3 - \left\{ \frac{\partial g}{\partial \sigma} \right\}^T \Delta\varepsilon_p \right) = 0. \quad (22.15)$$

The expression for f in (22.13) is in terms of *effective stress* σ equals the *yield stress* σ_Y , but the theory is not restricted to this setting. It is assumed that the stress σ_0 , back stress α_0

and the plastic strain ε_0 in the previous step fulfils the corresponding conditions (22.12), (22.13), (22.14) and (22.15) with respect to the state at that time, possibly with inequality in (22.13).

Taking the variation of these four equations results in

$$\delta\Delta\sigma = PC \left(P^T \delta\Delta\varepsilon + p \delta\Delta\varepsilon_3 - \left\{ \frac{\partial g}{\partial \sigma} \right\}^T \delta\Delta\varepsilon_p - \frac{\partial^2 g}{\partial \sigma^2} (P^T \delta\Delta\sigma - \delta\Delta\alpha) \Delta\varepsilon_p - \frac{\partial^2 g}{\partial \sigma \partial \varepsilon_p} \delta\Delta\varepsilon_p \Delta\varepsilon_p \right), \quad (22.16)$$

$$\frac{\partial f}{\partial \sigma} (P^T \delta\Delta\sigma - \delta\Delta\alpha) + \frac{\partial f}{\partial \varepsilon_p} \delta\Delta\varepsilon_p = 0, \quad (22.17)$$

$$\delta\Delta\alpha = \left\{ \frac{\partial h}{\partial \sigma} \right\}^T \delta\Delta\varepsilon_p + \frac{\partial^2 h}{\partial \sigma^2} (P^T \delta\Delta\sigma - \delta\Delta\alpha) \Delta\varepsilon_p + \frac{\partial h}{\partial \sigma \partial \varepsilon_p} \delta\Delta\varepsilon_p \Delta\varepsilon_p, \quad (22.18)$$

$$p^T C \left(P^T \delta\Delta\varepsilon + p \delta\Delta\varepsilon_3 - \left\{ \frac{\partial g}{\partial \sigma} \right\}^T \delta\Delta\varepsilon_p - \frac{\partial^2 g}{\partial \sigma^2} (P^T \delta\Delta\sigma - \delta\Delta\alpha) \Delta\varepsilon_p - \frac{\partial^2 g}{\partial \sigma \partial \varepsilon_p} \delta\Delta\varepsilon_p \Delta\varepsilon_p \right) = 0. \quad (22.19)$$

Solving (22.16) and (22.18) for $P^T \delta\Delta\sigma - \delta\Delta\alpha$ results in

$$P^T \delta\Delta\sigma - \delta\Delta\alpha = A^{-1} P^T PC P^T \delta\Delta\varepsilon + A^{-1} P^T PC p \delta\Delta\varepsilon_3 - A^{-1} F \delta\Delta\varepsilon_p \quad (22.20)$$

where

$$\begin{aligned} A &= I + P^T PC \frac{\partial^2 g}{\partial \sigma^2} \Delta\varepsilon_p + \frac{\partial^2 h}{\partial \sigma^2} \Delta\varepsilon_p \\ G &= \left\{ \frac{\partial g}{\partial \sigma} \right\}^T + \frac{\partial^2 g}{\partial \sigma \partial \varepsilon_p} \Delta\varepsilon_p \\ H &= \left\{ \frac{\partial h}{\partial \sigma} \right\}^T + \frac{\partial^2 h}{\partial \sigma \partial \varepsilon_p} \Delta\varepsilon_p \\ F &= P^T PC G + H. \end{aligned} \quad (22.21)$$

Inserting (22.20) into (22.17) and (22.19) results in a system of equations

$$\begin{pmatrix} p^T ECp & p^T L \\ -\frac{\partial f}{\partial \sigma} A^{-1} P^T PC p & \frac{\partial f}{\partial \sigma} A^{-1} F - \frac{\partial f}{\partial \varepsilon_p} \end{pmatrix} \begin{pmatrix} \delta\Delta\varepsilon_3 \\ \delta\Delta\varepsilon_p \end{pmatrix} = \begin{pmatrix} -p^T E \\ \frac{\partial f}{\partial \sigma} A^{-1} P^T P \end{pmatrix} C P^T \delta\Delta\varepsilon. \quad (22.22)$$

where

$$\begin{aligned}
 E &= I - C \frac{\partial^2 g}{\partial \sigma^2} A^{-1} P^T P \Delta \varepsilon_p \\
 L &= C \left(\frac{\partial^2 g}{\partial \sigma^2} A^{-1} F \Delta \varepsilon_p - G \right).
 \end{aligned} \tag{22.23}$$

Solving (22.22) results in

$$\delta \Delta \varepsilon_3 = \mathbf{a}_3^T \mathbf{B} \mathbf{C} \mathbf{P}^T \delta \Delta \varepsilon \tag{22.24}$$

and

$$\delta \Delta \varepsilon_p = \mathbf{a}_p^T \mathbf{B} \mathbf{C} \mathbf{P}^T \delta \Delta \varepsilon \tag{22.25}$$

where \mathbf{a}_3^T and \mathbf{a}_p^T are rows in the inverse of the system matrix in (22.22) and \mathbf{B} can be identified on the right-hand side of the same equation. Inserting (22.24) and (22.25) into (22.20) results in

$$\delta \Delta \sigma = P (E + E C p \mathbf{a}_3^T \mathbf{B} + L \mathbf{a}_p^T \mathbf{B}) C \mathbf{P}^T \delta \Delta \varepsilon \tag{22.26}$$

from which the consistent tangent matrix can be identified as

$$D = P (E + E C p \mathbf{a}_3^T \mathbf{B} + L \mathbf{a}_p^T \mathbf{B}) C \mathbf{P}^T. \tag{22.27}$$

Even though the involved expressions seem complicated, many expressions are repeated, and they simplify quite a bit when considering special cases. As an example, consider isotropic von Mises plasticity with linear hardening and back stress (mixed formulation) for which we have

$$\begin{aligned}
 f &= g = \sigma - \sigma_Y - \beta H \varepsilon_p \\
 h &= \frac{2}{3} (1 - \beta) H f
 \end{aligned} \tag{22.28}$$

where $\sigma = \sqrt{\frac{3}{2} \mathbf{s}^T \mathbf{s}}$ and \mathbf{s} is the deviatoric stress, including subtraction of back stress α . Then

$$\begin{aligned}
 \frac{\partial f}{\partial \sigma} &= \frac{\partial g}{\partial \sigma} = \frac{3}{2\sigma} \mathbf{s}^T \\
 \frac{\partial h}{\partial \sigma} &= \frac{(1 - \beta) H}{\sigma} \mathbf{s}^T
 \end{aligned} \tag{22.29}$$

and

$$\begin{aligned}
 \frac{\partial^2 g}{\partial \sigma^2} &= \frac{3}{2\sigma} \left(\mathbf{I}^{dev} - \frac{3}{2\sigma^2} \mathbf{s} \mathbf{s}^T \right) \\
 \frac{\partial^2 h}{\partial \sigma^2} &= \frac{(1 - \beta) H}{\sigma} \left(\mathbf{I}^{dev} - \frac{3}{2\sigma^2} \mathbf{s} \mathbf{s}^T \right)
 \end{aligned} \tag{22.30}$$

If the elastic matrix \mathbf{C} is isotropic, then we have

$$\frac{\partial f}{\partial \sigma} \mathbf{C} = \frac{3G}{\sigma} \mathbf{s}^T \quad (22.31)$$

and

$$\mathbf{C} \frac{\partial^2 g}{\partial \sigma^2} = \frac{3G}{\sigma} \left(\mathbf{I}^{dev} - \frac{3}{2\sigma^2} \mathbf{s} \mathbf{s}^T \right) \quad (22.32)$$

with G being the shear modulus, resulting in an \mathbf{A} matrix that is invertible on closed form. We also have

$$\frac{\partial^2 g}{\partial \sigma \partial \varepsilon_p} = \frac{\partial^2 h}{\partial \sigma \partial \varepsilon_p} = 0. \quad (22.33)$$

In 3D constitutive laws, $\mathbf{P} = \mathbf{I}$ and $\mathbf{p} = \mathbf{0}$, so the system (22.22) reduces to a scalar equation with $\mathbf{a}_3^T = \mathbf{0}$ in (22.27), and so on, so the implementation is very much tractable.

23

Material Models

LS-DYNA accepts a wide range of material and equation of state models, each with a unique number of history variables. Approximately 150 material models are implemented, and space has been allotted for up to 10 user-specified models.

- 1 Elastic
- 2 Orthotropic Elastic
- 3 Kinematic/Isotropic Elastic-Plastic
- 4 Thermo-Elastic-Plastic
- 5 Soil and Crushable/Non-crushable Foam
- 6 Viscoelastic
- 7 Blatz - Ko Rubber
- 8 High Explosive Burn
- 9 Null Hydrodynamics
- 10 Isotropic-Elastic-Plastic-Hydrodynamic
- 11 Temperature Dependent, Elastoplastic, Hydrodynamic
- 12 Isotropic-Elastic-Plastic
- 13 Elastic-Plastic with Failure Model
- 14 Soil and Crushable Foam with Failure Model
- 15 Johnson/Cook Strain and Temperature Sensitive Plasticity
- 16 Pseudo TENSOR Concrete/Geological Model
- 17 Isotropic Elastic-Plastic Oriented Crack Model
- 18 Power Law Isotropic Plasticity
- 19 Strain Rate Dependent Isotropic Plasticity
- 20 Rigid
- 21 Thermal Orthotropic Elastic
- 22 Composite Damage Model
- 23 Thermal Orthotropic Elastic with 12 Curves
- 24 Piecewise Linear Isotropic Plasticity
- 25 Inviscid Two Invariant Geologic Cap Model
- 26 Metallic Honeycomb

- 27 Compressible Mooney-Rivlin Rubber
- 28 Resultant Plasticity
- 29 Force Limited Resultant Formulation
- 30 Closed-Form Update Shell Plasticity
- 31 Slightly Compressible Rubber Model
- 32 Laminated Glass Model
- 33 Barlat's Anisotropic Plasticity Model
- 34 Fabric
- 35 Kinematic/Isotropic Elastic-Plastic Green-Naghdi Rate
- 36 Barlat's 3-Parameter Plasticity Model
- 37 Transversely Anisotropic Elastic-Plastic
- 38 Blatz-Ko Compressible Foam
- 39 Transversely Anisotropic Elastic-Plastic with FLD
- 40 Nonlinear Elastic Orthotropic Material
- 41-50 User Defined Material Models
- 42 Planar Anisotropic Plasticity Model
- 48 Strain Rate Dependent Plasticity with Size Dependent Failure
- 51 Temperature and Rate Dependent Plasticity
- 52 Sandia's Damage Model
- 53 Low Density Closed Cell Polyurethane Foam
- 54-55 Composite Damage Model
- 57 Low Density Urethane Foam
- 58 Laminated Composite Fabric
- 59 Composite Failure
- 60 Elastic with Viscosity
- 61 Maxwell/Kelvin Viscoelastic
- 62 Viscous Foam
- 63 Isotropic Crushable Foam
- 64 Strain Rate Sensitive Power-Law Plasticity
- 65 Modified Zerilli/Armstrong
- 66 Linear Stiffness/Linear Viscous 3D Discrete Beam
- 67 Nonlinear Stiffness/Viscous 3D Discrete Beam
- 68 Nonlinear Plastic/Linear Viscous 3D Discrete Beam
- 69 Side Impact Dummy Damper, SID Damper
- 70 Hydraulic/Gas Damper
- 71 Cable
- 72 Concrete Damage Model
- 73 Low Density Viscoelastic Foam
- 74 Elastic Spring for the Discrete Beam
- 75 Bilkhu/Dubois Foam Model
- 76 General Viscoelastic
- 77 Hyperviscoelastic Rubber
- 78 Soil/Concrete
- 79 Hysteretic Soil
- 80 Ramberg-Osgood Plasticity

81	Plastic with Damage
82	Isotropic Elastic-Plastic with Anisotropic Damage
83	Fu-Chang's Foam with Rate Effects
84-85	Winfrith Concrete
84	Winfrith Concrete Reinforcement
86	Orthotropic-Viscoelastic
87	Cellular Rubber
88	MTS Model
89	Plasticity Polymer
90	Acoustic
91	Soft Tissue
93	Elastic 6DOF Spring Discrete Beam
94	Inelastic Spring Discrete Beam
95	Inelastic 6DOF Spring Discrete Beam
96	Brittle Damage Model
97	General Joint Discrete Beam
100	Spot weld
101	GE Thermoplastics
102	Hyperbolic Sin
103	Anisotropic Viscoplastic
104	Damage 1
105	Damage 2
106	Elastic Viscoplastic Thermal
110	Johnson-Holmquist Ceramic Model
111	Johnson-Holmquist Concrete Model
112	Finite Elastic Strain Plasticity
113	Transformation Induced Plasticity
114	Layered Linear Plasticity
115	Elastic Creep Model
116	Composite Lay-Up Model
117-118	Composite Matrix
119	General Spring and Damper Model
120	Gurson Dilational-Plastic Model
120	Gurson Model with Rc-Dc
121	Generalized Nonlinear 1DOF Discrete Beam
122	Hill 3RC
123	Modified Piecewise Linear Plasticity
124	Tension-Compression Plasticity
126	Metallic Honeycomb
127	Arruda-Boyce rubber
128	Anisotropic heart tissue
129	Lung tissue
130	Special Orthotropic
131	Isotropic Smeared Crack
132	Orthotropic Smeared Crack

133	Barlat YLD2000
134	Viscoelastic Fabric
139	Modified Force Limited
140	Vacuum
141	Rate Sensitive Polymer
142	Transversely Anisotropic Crushable Foam
143	Wood Model
144	Pitzer Crushable Foam
145	Schwer Murray Cap Model
146	1DOF Generalized Spring
147	FHWA Soil Model
148	Gas Mixture
150	CFD
151	EMMI
154	Deshpande-Fleck Foam
156	Muscle
158	Rate Sensitive Composite Fabric
159	Continuous Surface Cap Model
161-162	Composite MSC
163	Modified Crushable Foam
164	Brain Linear Viscoelastic
166	Moment Curvature Beam
169	Arup Adhesive
170	Resultant Anisotropic
175	Viscoelastic Maxwell
176	Quasilinear Viscoelastic
177	Hill Foam
178	Viscoelastic Hill Foam
179	Low Density Synthetic Foam
181	Simplified Rubber/Foam
183	Simplified Rubber with Damage
184	Cohesive Elastic
185	Cohesive TH
191	Seismic Beam
192	Soil Brick
193	Drucker Prager
194	RC Shear Wall
195	Concrete Beam
196	General Spring Discrete Beam
197	Seismic Isolator
198	Jointed Rock

In the table below, a list of the available material models and the applicable element types are given. Some materials include strain rate sensitivity, failure, equations of state, and thermal effects and this is also noted. General applicability of the materials to certain kinds of behavior is suggested in the last column.

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydrodyn Mt Metal Pl Plastic Rb Rubber Sl Soil/C onc
1	Elastic	Y	Y	Y	Y					Gn, Fl
2	Orthotropic Elastic (Anisotropic - solids)	Y		Y	Y					Cm, Mt
3	Plastic Kinematic/Isotropic	Y	Y	Y	Y	Y	Y			Cm, Mt, Pl
4	Elastic Plastic Thermal	Y	Y	Y	Y				Y	Mt, Pl
5	Soil and Foam	Y								Fm, Sl
6	Linear Viscoelastic	Y	Y	Y		Y				Rb
7	Blatz-Ko Rubber	Y		Y						Rb, Polyurethane
8	High Explosive Burn	Y						Y		Hy
9	Null Material	Y					Y	Y	Y	Fl, Hy
10	Elastic Plastic Hydro(dynamic)	Y					Y	Y		Hy, Mt
11	Steinberg: Temp. Dependent Elastoplastic	Y				Y	Y	Y	Y	Hy, Mt
12	Isotropic Elastic Plastic	Y		Y	Y					Mt
13	Isotropic Elastic Plastic with Failure	Y					Y			Mt
14	Soil and Foam with Failure	Y					Y			Fm, Sl
15	Johnson/Cook Plasticity Model	Y		Y		Y	Y	Y	Y	Hy, Mt
16	Pseudo TENSOR Geological Model	Y				Y	Y	Y		Sl
17	Oriented Crack (Elastoplastic with Fracture)	Y					Y	Y		Hy, Mt, Pl

18	Power Law Plasticity (Isotropic)	Y	Y	Y	Y	Y	Mt, Pl
19	Strain Rate Dependent Plasticity	Y		Y	Y	Y	Mt, Pl
20	Rigid	Y	Y	Y	Y		

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn Gen- eral Cm Com- posites Cr Ce- ramics Fl Fluids Fm Foam Gl Glass Hy Hy- dro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/C onc
21	Orthotropic Thermal (Elastic)	Y		Y	Y			Y		Gn
22	Composite Damage	Y		Y	Y		Y			Cm
23	Temperature Dependent Orthotropic	Y		Y	Y			Y		Cm
24	Piecewise Linear Plasticity (Isotropic)	Y	Y	Y	Y	Y	Y			Mt, Pl
25	Inviscid Two Invariant Geologic Cap	Y								Sl
26	Honeycomb	Y				Y	Y			Cm, Fm, Sl
27	Mooney-Rivlin Rubber	Y		Y						Rb
28	Resultant Plasticity		Y	Y						Mt
29	Force Limited Resultant Formulation		Y							
30	Closed Form Update Shell Plasticity			Y	Y					Mt
31	Slightly Compressible Rubber	Y								Rb
32	Laminated Glass (Composite)			Y	Y		Y			Cm, Gl
33	Barlat Anisotropic Plasticity	Y		Y	Y					Cr, Mt
34	Fabric			Y						
35	Plastic Green-Naghdi Rate	Y				Y				Mt
36	3-Parameter Barlat Plasticity			Y						Mt
37	Transversely Anisotropic Elastic Plastic			Y	Y					Mt
38	Blatz-Ko Foam	Y		Y						Fm, Pl
39	FLD Transversely Anisotropic			Y	Y					Mt
40	Nonlinear Orthotropic			Y			Y	Y		Cm

41-	User Defined Materials	Y	Y	Y	Y	Y	Y	Y	Gn
42	Planar Anisotropic Plasticity Model								

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Composites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro-dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/C onc
51	Bamman (Temp/Rate Dependent Plasticity)	Y		Y	Y	Y		Y		Gn
52	Bamman Damage	Y		Y	Y	Y	Y		Y	Mt
53	Closed Cell Foam (Low Density Polyurethane)	Y								Fm
54	Composite Damage with Change Failure			Y			Y			Cm
55	Composite Damage with Tsai-Wu Failure			Y			Y			Cm
56										
57	Low Density Urethane Foam	Y				Y	Y			Fm
58	Laminated Composite Fabric			Y						
59	Composite Failure (Plasticity Based)	Y		Y			Y			Cm, Cr
60	Elastic with Viscosity (Viscous Glass)	Y		Y		Y		Y		Gl
61	Kelvin-Maxwell Viscoelastic	Y				Y				Fm
62	Viscous Foam (Crash Dummy Foam)	Y				Y				Fm
63	Isotropic Crushable Foam	Y				Y				Fm
64	Rate Sensitive Powerlaw Plasticity	Y		Y	Y	Y				Mt
65	Zerilli-Armstrong (Rate/Temp Plasticity)	Y		Y		Y		Y	Y	Mt
66	Linear Elastic Discrete Beam		Y			Y				
67	Nonlinear Elastic Discrete Beam		Y			Y				

68	Nonlinear Plastic Discrete Beam	Y	Y Y	
69	SID Damper Discrete Beam	Y	Y	
70	Hydraulic Gas Damper Discrete Beam	Y	Y	
71	Cable Discrete Beam (Elastic)	Y		
72	Concrete Damage	Y	Y Y Y	Sl

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	c	Notes:
											Gn General Cm Compo- sites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro- dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Con
73	Low Density Viscous Foam	Y				Y Y				Fm	
74	Elastic Spring for the Discrete Beam										
75	Bilkhui/Dubois Foam (Isotropic)	Y				Y				Fm	
76	General Viscoelastic (Maxwell Model)	Y				Y				Rb	
77	Hyperelastic and Ogden Rubber	Y								Rb	
78	Soil Concrete	Y				Y				Sl	
79	Hysteretic Soil (Elasto-Perfectly Plastic)	Y					Y			Sl	
80	Ramberg Osgood Plasticity										
81	Plasticity with Damage (Elasto-Plastic)	Y	Y	Y	Y	Y	Y			Mt, Pl	
82	Isotropic Elastic-Plastic with Anisotropic Damage										
83	Fu Chang Foam	Y				Y	Y			Fm	
84	Winfrith Concrete Reinforcement	Y									
85											
86	Orthotropic Viscoelastic		Y			Y				Rb	

87	Cellular Rubber	Y		Y		Rb
88	MTS	Y	Y	Y	Y	Mt
89	Plasticity Polymer		Y			Fl
90	Acoustic	Y				
91	Soft Tissue	Y	Y			
93	Elastic 6DOF Spring Discrete Beam		Y			
94	Inelastic Spring Discrete Beam		Y			

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Compo- sites Cr Ceram- ics Fl Fluids Fm Foam Gl Glass Hy Hydro- dyn Mt Metal Pl Plastic Rb Rubber Sl nc Soil/Co
95	Inelastic 6DOF Spring Discrete Beam		Y							
96	Brittle Damage	Y				Y	Y			
97	General Joint Discrete Beam		Y							
98	Simplified Johnson Cook	Y	Y	Y	Y					
99	Simplified Johnson Cook Orthotropic Damage									
100	Spotweld		Y							
101	GEPLASTIC Strate2000a			Y						
102	Inv Hyperbolic Sin	Y								
103	Anisotropic Viscoplastic	Y		Y						
104	Damage 1	Y		Y						

105	Damage 2	Y	Y		
106	Elastic Viscoplastic Thermal	Y	Y	Y	
107					
108					
109					
110	Johnson Holmquist Ceramics	Y			
111	Johnson Holmquist Concrete	Y			
112	Finite Elastic Strain Plasticity	Y			
113	TRIP		Y Y	Y	Mt
114	Layered Linear Plasticity		Y Y		
115	Unified Creep	Y			

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	Notes:
										Gn General Cm Compo- sites Cr Ceram- ics Fl Fluids Fm Foam Gl Glass Hy Hydro- dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Co nc
116	Composite Layup		Y							
117	Composite Matrix		Y							
118	Composite Direct		Y							
119	General Nonlinear 6DOF Discrete Beam	Y				Y	Y			
120	Gurson		Y							
121	Generalized Nonlinear 1DOF Discrete Beam	Y								

122	Hill 3RC			
123	Modified Piecewise Linear Plasticity		Y Y	
124	Plasticity Compression Tension	Y		
126	Modified Honeycomb	Y		
127	Arruda Boyce Rubber	Y		
128	Heart Tissue	Y		
129	Lung Tissue	Y		
130	Special Orthotropic		Y	
131	Isotropic Smeared Crack	Y	Y	Mt, Cm
132	Orthotropic Smeared Crack	Y	Y	Mt, Cm
133	Barlat YLD2000			
139	Modified Force Limited	Y		
140	Vacuum			
141	Rate Sensitive Polymer			

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	c	Notes:
142	Transversely Anisotropic Crushable Foam										Gn General
143	Wood										Cm Compo- sites
144	Pitzer Crushable Foam										Cr Ceramics
145	Schwer Murray Cap Model										Fl Fluids
											Fm Foam
											Gl Glass
											Hy Hydro- dyn
											Mt Metal
											Pl Plastic
											Rb Rubber
											Sl Soil/Con

146	1DOF Generalized Spring				
147	FHWA Soil				
147	FHWA Soil Nebraska				
148	Gas Mixture				
150	CFD				
151	EMMI	Y	Y	Y Y Y	Mt
154	Deshpande Fleck Foam				
156	Muscle	Y		Y	
158	Rate Sensitive Composite Fabric		Y Y	Y Y	Cm
159	CSMC	Y	Y	Y Y	Sl
161	Composite MSC	Y			
163	Modified Crushable Foam				
164	Brain Linear Viscoelastic				
166	Moment Curvature Beam	Y			
169	Arup Adhesive	Y	Y	Y	Pb
170	Resultant Anisotropic		Y Y		Pl
175	Viscoelastic Thermal	Y	Y Y	Y Y	Rb
176	Quasilinear Viscoelastic				

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	c	Notes:
											Gn General Cm Compo- sites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro- dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Con
177	Hill Foam										
178	Viscoelastic Hill Foam										

179	Low Density Synthetic Foam			
181	Simplified Rubber			
183	Simplified Rubber with Damage	Y Y Y	Y Y	Rb
184	Cohesive Elastic	Y	Y	Cm, Mt
185	Cohesive TH	Y	Y	Cm, Mt
191	Seismic Beam	Y		
192	Soil Brick	Y		
193	Drucker Prager	Y		
194	RC Shear Wall	Y		
195	Concrete Beam	Y		
196	General Spring Discrete Beam	Y		
197	Seismic Isolator	Y	Y	Mt
198	Jointed Rock	Y	Y	
DS 1	Spring Elastic (Linear)	Y		
DS 2	Damper Viscous (Linear)	Y	Y	
DS 3	Spring Elastoplastic (Isotropic)	Y		
DS 4	Spring Nonlinear Elastic	Y	Y	
DS 5	Damper Nonlinear Elastic	Y	Y	
DS 6	Spring General Nonlinear	Y		

Material Number	Material Title	Bricks	Beams	Thin Shells	Thick Shells	Strain-Rate Effects	Failure	Equation-of-State	Thermal Effects	c	Notes:
											Gn General Cm Compo- sites Cr Ceramics Fl Fluids Fm Foam Gl Glass Hy Hydro- dyn Mt Metal Pl Plastic Rb Rubber Sl Soil/Con
DS 7	Spring Maxwell (Three Parameter Viscoelastic)	Y				Y					
DS 8	Spring Inelastic (Tension or Compression)	Y									
DS 13	Spring Trilinear Degrading										
DS 14	Spring Squat Shearwall										
DS 15	Spring Muscle										
SB1	Seatbelt										
T01	Thermal Isotropic	Y	Y					Y			
T02	Thermal Orthotropic	Y	Y					Y			
T03	Thermal Isotropic (Temp. Dependent)	Y	Y					Y			
T04	Thermal Orthotropic (Temp. Dependent)	Y	Y					Y			
T05	Thermal Isotropic (Phase Change)	Y	Y					Y			
T06	Thermal Isotropic (Temp Dep-Load Curve)	Y	Y					Y			
T11	Thermal User Defined	Y	Y					Y			

23.1 Material Model 1: Elastic

In this elastic material we compute the co-rotational rate of the deviatoric Cauchy stress tensor as

$$s_{ij}^{\nabla^{n+1/2}} = 2G\dot{\varepsilon}_{ij}^{n+1/2}, \quad (23.1.1)$$

and pressure

$$p^{n+1} = -K \ln V^{n+1}, \quad (23.1.2)$$

where G and K are the elastic shear and bulk moduli, respectively, and V is the relative volume, i.e., the ratio of the current volume to the initial volume.

23.2 Material Model 2: Orthotropic Elastic

The implementation of this material model depends on the element type and in some cases the solver type as well.

For solids with an explicit formulation or a nonlinear implicit formulation (meaning NSOLVR $\neq 1$ on *CONTROL_IMPLICIT_SOLUTION), the constitutive model for this material is the St. Venant Kirchhoff model, a simple hyperelastic model. In this case the material law that relates second Piola-Kirchhoff stress \mathbf{S} to the Green-St. Venant strain \mathbf{E} is

$$\mathbf{S} = \mathbf{C} \cdot \mathbf{E} = \mathbf{T}^T \mathbf{C}_1 \mathbf{T} \cdot \mathbf{E}, \quad (23.2.1)$$

where \mathbf{T} is the transformation matrix [Cook 1974]:

$$\mathbf{T} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & (l_1 m_2 + l_2 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 l_2 + n_2 l_1) \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & (l_2 m_3 + l_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 l_3 + n_3 l_2) \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & (l_3 m_1 + l_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 l_1 + n_1 l_3) \end{bmatrix}, \quad (23.2.2)$$

l_i, m_i, n_i are the direction cosines

$$x'_i = l_i x_1 + m_i x_2 + n_i x_3, \quad i = 1, 2, 3, \quad (23.2.3)$$

and x'_i denotes the material axes. The constitutive matrix \mathbf{C}_1 is defined in terms of the material axes as

$$\mathbf{C}_1^{-1} = \begin{bmatrix} \frac{1}{E_{11}} & -\frac{\nu_{21}}{E_{22}} & -\frac{\nu_{31}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_{11}} & \frac{1}{E_{22}} & -\frac{\nu_{32}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_{11}} & -\frac{\nu_{23}}{E_{22}} & \frac{1}{E_{33}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix}, \quad (23.2.4)$$

where the subscripts denote the material axes, i.e.,

$$\nu_{ij} = \nu_{x'_i x'_j} \quad \text{and} \quad E_{ii} = E_{x'_i}. \quad (23.2.5)$$

Since \mathbf{C}_1 is symmetric

$$\frac{v_{12}}{E_{11}} = \frac{v_{21}}{E_{22}}, \text{ etc.} \quad (23.2.6)$$

The vector of Green-St. Venant strain components is

$$\mathbf{E}^T = [E_{11} \ E_{22} \ E_{33} \ E_{12} \ E_{23} \ E_{31}]. \quad (23.2.7)$$

After computing S_{ij} , we use

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl} \quad (23.2.8)$$

to obtain the Cauchy stress.

When applied to shell elements within an explicit formulation or nonlinear implicit formulation, Material 2 is implemented in the local coordinates of the shell as linear elasticity. In terms of the polar decomposition for \mathbf{F} , $\mathbf{F} = \mathbf{R}\mathbf{U}$, the rotation, \mathbf{R} , is fully finite, but the stretch, \mathbf{U} , corresponds to a small deformation. While the material response is linear, the shells themselves can undergo finite rotations consistent with applied forces.

For the implicit linear solver (NSOLVR = 1 on *CONTROL_IMPLICIT_SOLVER), this material model is a linear elasticity model.

23.3 Material Model 3: Elastic Plastic with Kinematic Hardening

Isotropic, kinematic, or a combination of isotropic and kinematic hardening may be obtained by varying a parameter, called β between 0 and 1. For β equal to 0 and 1, respectively, kinematic and isotropic hardening are obtained as shown in [Figure 23.3.1](#). Krieg and Key [1976] formulated this model and the implementation is based on their paper.

In isotropic hardening, the center of the yield surface is fixed but the radius is a function of the plastic strain. In kinematic hardening, the radius of the yield surface is fixed but the center translates in the direction of the plastic strain. Thus the yield condition is

$$\phi = \frac{1}{2} \tilde{\zeta}_{ij} \tilde{\zeta}_{ij} - \frac{\sigma_y^2}{3} = 0, \quad (23.3.1)$$

where

$$\tilde{\zeta}_{ij} = s_{ij} - \alpha_{ij} \quad (23.3.2)$$

$$\sigma_y = \sigma_0 + \beta E_p \varepsilon_{\text{eff}}^p. \quad (23.3.3)$$

The co-rotational rate of α_{ij} is

$$\alpha_{ij}^{\nabla} = (1 - \beta) \frac{2}{3} E_p \dot{\varepsilon}_{ij}^p. \quad (23.3.4)$$

Hence,

$$\alpha_{ij}^{n+1} = \alpha_{ij}^n + \left(\alpha_{ij}^{\nabla n+1/2} + \alpha_{ik}^n \Omega_{kj}^{n+1/2} + \alpha_{jk}^n \Omega_{ki}^{n+1/2} \right) \Delta t^{n+1/2}. \quad (23.3.5)$$

Strain rate is accounted for using the Cowper-Symonds [Jones 1983] model which scales the yield stress by a strain rate dependent factor

$$\sigma_y = \left[1 + \left(\frac{\dot{\varepsilon}}{C} \right)^{\frac{1}{p}} \right] (\sigma_0 + \beta E_p \varepsilon_{\text{eff}}^p), \quad (23.3.6)$$

where p and C are user defined input constants and $\dot{\varepsilon}$ is the strain rate defined as:

$$\dot{\varepsilon} = \sqrt{\dot{\varepsilon}_{ij} \dot{\varepsilon}_{ij}}. \quad (23.3.7)$$

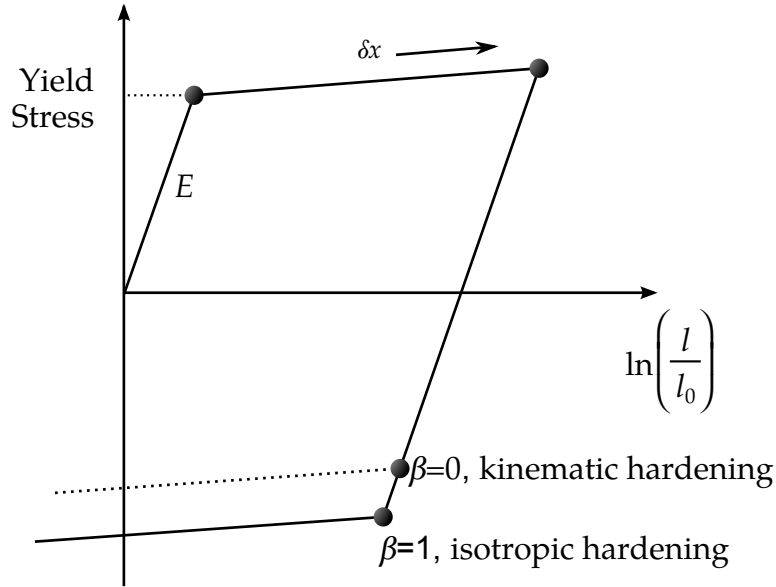


Figure 23.3.1. Elastic-plastic behavior with isotropic and kinematic hardening where l_0 and l are the undeformed and deformed length of uniaxial tension specimen, respectively.

The current radius of the yield surface, σ_y , is the sum of the initial yield strength, σ_0 , plus the growth $\beta E_p \varepsilon_{\text{eff}}^p$, where E_p is the plastic hardening modulus

$$E_p = \frac{E_t E}{E - E_t}, \quad (23.3.8)$$

and $\varepsilon_{\text{eff}}^p$ is the effective plastic strain

$$\varepsilon_{\text{eff}}^p = \int_0^t \left(\frac{2}{3} \dot{\varepsilon}_{ij}^p \dot{\varepsilon}_{ij}^p \right)^{1/2} dt. \quad (23.3.9)$$

The plastic strain rate is the difference between the total and elastic (right superscript e) strain rates:

$$\dot{\varepsilon}_{ij}^p = \dot{\varepsilon}_{ij} - \dot{\varepsilon}_{ij}^e. \quad (23.3.10)$$

In the implementation of this material model, the deviatoric stresses are updated elastically, as described for model 1, but repeated here for the sake of clarity:

$$\sigma_{ij}^* = \sigma_{ij}^n + C_{ijkl} \Delta \varepsilon_{kl}, \quad (23.3.11)$$

where

- σ_{ij}^* is the trial stress tensor,
- σ_{ij}^n is the stress tensor from the previous time step,
- C_{ijkl} is the elastic tangent modulus matrix,

$\Delta\epsilon_{kl}$ is the incremental strain tensor.
and, if the yield function is satisfied, nothing else is done. If, however, the yield function is violated, an increment in plastic strain is computed, the stresses are scaled back to the yield surface, and the yield surface center is updated.

Let s_{ij}^* represent the trial elastic deviatoric stress state at $n + 1$

$$s_{ij}^* = \sigma_{ij}^* - \frac{1}{3}\sigma_{kk}^* \quad (23.3.12)$$

and

$$\zeta_{ij}^* = s_{ij}^* - \alpha_{ij}. \quad (23.3.13)$$

Define the yield function,

$$\phi = \frac{3}{2}\zeta_{ij}^*\zeta_{ij}^* - \sigma_y^2 = \Lambda^2 - \sigma_y^2 \begin{cases} \leq 0 & \text{for elastic or neutral loading} \\ > 0 & \text{for plastic hardening} \end{cases}, \quad (23.3.14)$$

For plastic hardening then

$$\epsilon_{\text{eff}}^{p^{n+1}} = \epsilon_{\text{eff}}^{p^n} + \frac{\Lambda - \sigma_y}{3G + E_p} = \epsilon_{\text{eff}}^{p^n} + \Delta\epsilon_{\text{eff}}^p, \quad (23.3.15)$$

scale back the stress deviators

$$\sigma_{ij}^{n+1} = \sigma_{ij}^* - \frac{3G\Delta\epsilon_{\text{eff}}^p}{\Lambda}\zeta_{ij}^*, \quad (23.3.16)$$

and update the center:

$$\alpha_{ij}^{n+1} = \alpha_{ij}^n + \frac{(1 - \beta)E_p\Delta\epsilon_{\text{eff}}^p}{\Lambda}\zeta_{ij}^*. \quad (23.3.17)$$

Plane Stress Plasticity

The plane stress plasticity options apply to beams, shells, and thick shells. Since the stresses and strain increments are transformed to the lamina coordinate system for the constitutive evaluation, the stress and strain tensors are in the local coordinate system.

The application of the Jaumann rate to update the stress tensor allows for the possibility that the normal stress, σ_{33} , will not be zero. The first step in updating the stress tensor is to compute a trial plane stress update assuming that the incremental strains are elastic. In the above, the normal strain increment $\Delta\epsilon_{33}$ is replaced by the elastic strain increment

$$\Delta\epsilon_{33} = -\frac{\sigma_{33} + \lambda(\Delta\epsilon_{11} + \Delta\epsilon_{22})}{\lambda + 2\mu}, \quad (23.3.18)$$

where λ and μ are Lamé's constants.

When the trial stress is within the yield surface, the strain increment is elastic and the stress update is completed. Otherwise, for the plastic plane stress case, secant iteration is used to solve Equation (23.3.16) for the normal strain increment ($\Delta\epsilon_{33}$) required to produce a zero normal stress:

$$\sigma_{33}^i = \sigma_{33}^* - \frac{3G\Delta\epsilon_{\text{eff}}^{\text{p}^i}\bar{\epsilon}_{33}}{\Lambda}, \quad (23.3.19)$$

Here, the superscript i indicates the iteration number.

The secant iteration formula for $\Delta\epsilon_{33}$ (the superscript p is dropped for clarity) is

$$\Delta\epsilon_{33}^{i+1} = \Delta\epsilon_{33}^{i-1} - \frac{\Delta\epsilon_{33}^i - \Delta\epsilon_{33}^{i-1}}{\sigma_{33}^i - \sigma_{33}^{i-1}} \sigma_{33}^{i-1}, \quad (23.3.20)$$

where the two starting values are obtained from the initial elastic estimate and by assuming a purely plastic increment, i.e.,

$$\Delta\epsilon_{33}^1 = -(\Delta\epsilon_{11} - \Delta\epsilon_{22}). \quad (23.3.21)$$

These starting values should bound the actual values of the normal strain increment.

The iteration procedure uses the updated normal strain increment to update first the deviatoric stress and then the other quantities needed to compute the next estimate of the normal stress in Equation (23.3.19). The iterations proceed until the normal stress σ_{33}^i is sufficiently small. The convergence criterion requires convergence of the normal strains:

$$\frac{|\Delta\epsilon_{33}^i - \Delta\epsilon_{33}^{i-1}|}{|\Delta\epsilon_{33}^{i+1}|} < 10^{-4}. \quad (23.3.22)$$

After convergence, the stress update is completed using the relationships given in Equations (23.3.16) and (23.3.17)

23.4 Material Model 4: Thermo-Elastic-Plastic

This model was adapted from the NIKE2D [Hallquist 1979] code. A more complete description of its formulation is given in the NIKE2D user's manual.

Letting T represent the temperature, we compute the elastic co-rotational stress rate as

$$\sigma_{ij}^{\nabla} = C_{ijkl}(\dot{\epsilon}_{kl} - \dot{\epsilon}_{kl}^T) + \dot{\theta}_{ij}dT, \quad (23.4.1)$$

where

$$\dot{\theta}_{ij} = \frac{dC_{ijkl}}{dT} C_{klmn}^{-1} \dot{\sigma}_{mn}, \quad (23.4.2)$$

and C_{ijkl} is the temperature dependent elastic constitutive matrix:

$$C_{ijkl} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}, \quad (23.4.3)$$

where ν is Poisson's ratio. The thermal strain rate can be written in terms of the coefficient of thermal expansion α as:

$$\dot{\epsilon}_{ij}^T = \alpha \dot{T} \delta_{ij}, \quad (23.4.4)$$

When treating plasticity, we use a procedure analogous to that for material 3. We update the stresses elastically and check to see if we violate the isotropic yield function

$$\phi = \frac{1}{2} s_{ij} s_{ij} - \frac{\sigma_y(T)^2}{3}, \quad (23.4.5)$$

where

$$\sigma_y(T) = \sigma_o(T) + E_p(T) \epsilon_{\text{eff}}^p. \quad (23.4.6)$$

The initial yield, σ_o , and plastic hardening modulus, E_p , are temperature dependent. If the behavior is elastic we do nothing; otherwise, we scale back the stress deviators by the factor f_s :

$$s_{ij}^{n+1} = f_s s_{ij}^*, \quad (23.4.7)$$

where

$$f_s = \frac{\sigma_y}{\left(\frac{3}{2} s_{ij}^* s_{ij}^*\right)^{1/2}}, \quad (23.4.8)$$

and update the plastic strain by the increment

$$\Delta \varepsilon_{\text{eff}}^p = \frac{(1 - f_s) \left(\frac{3}{2} s_{ij}^* s_{ij}^*\right)^{1/2}}{G + 3E_p}. \quad (23.4.9)$$

23.5 Material Model 5: Soil and Crushable Foam

This model, due to Krieg [1972], provides a simple model for foam and soils whose material properties are not well characterized. We believe the other foam models in LS-DYNA are superior in their performance and are recommended over this model which simulates the crushing through the volumetric deformations. If the yield stress is too low, this foam model gives nearly fluid like behavior.

A pressure-dependent flow rule governs the deviatoric behavior:

$$\phi_s = \frac{1}{2} s_{ij} s_{ij} - (a_0 + a_1 p + a_2 p^2), \quad (23.5.1)$$

where a_0 , a_1 , and a_2 are user-defined constants. Volumetric yielding is determined by a tabulated curve of pressure versus volumetric strain. Elastic unloading from this curve is assumed to a tensile cutoff as illustrated in [Figure 23.5.1](#).

Implementation of this model is straightforward. One history variable, the maximum volumetric strain in compression, is stored. If the new compressive volumetric strain exceeds the stored value, loading is indicated. When the yield condition is violated, the updated trial stresses, s_{ij}^* , are scaled back using a simple radial return algorithm:

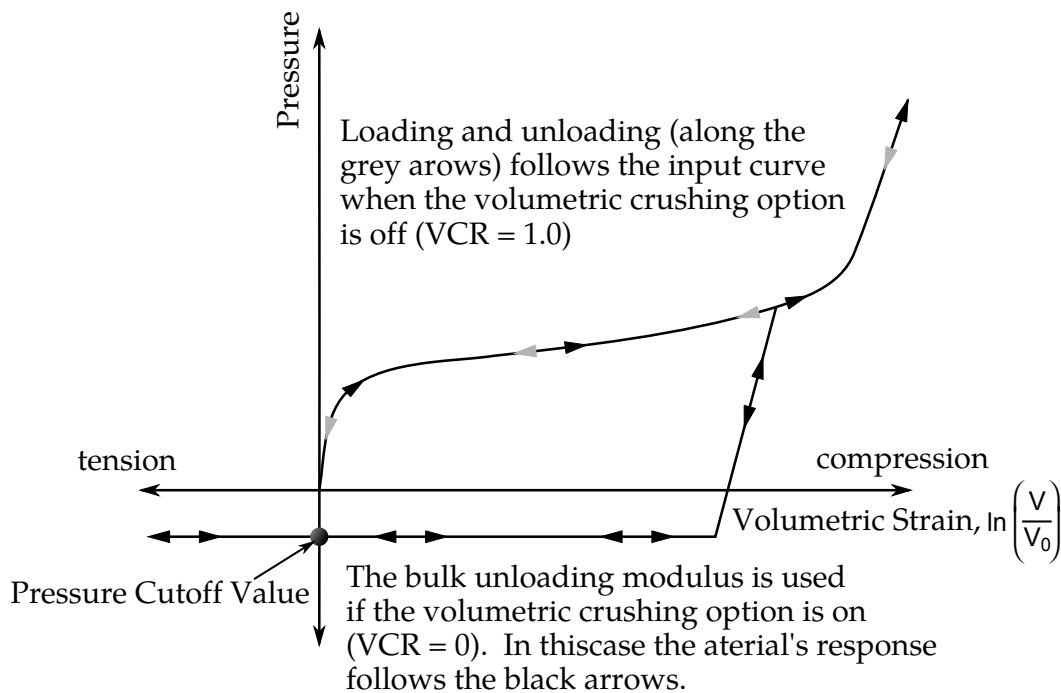


Figure 23.5.1. Volumetric strain versus pressure curve for soil and crushable foam model.

$$s_{ij}^{n+1} = \left(\frac{a_0 + a_1 p + a_2 p^2}{\frac{1}{2} s_{ij} s_{ij}} \right)^{1/2} s_{ij}^*. \quad (23.5.2)$$

If the hydrostatic tension exceeds the cutoff value, the pressure is set to the cutoff value and the deviatoric stress tensor is zeroed.

23.6 Material Model 6: Viscoelastic

In this model, linear viscoelasticity is assumed for the deviatoric stress tensor [Herrmann and Peterson 1968]:

$$s_{ij} = 2 \int_0^t \phi(t - \tau) \frac{\partial \varepsilon'_{ij}(\tau)}{\partial \tau} d\tau, \quad (23.6.1)$$

where

$$\phi(t) = G_\infty + (G_0 - G_\infty)e^{-\beta t}, \quad (23.6.2)$$

is the shear relaxation modulus. A recursion formula is used to compute the new value of the hereditary integral at time t^{n+1} from its value at time t^n . Elastic bulk behavior is assumed:

$$p = K \ln V, \quad (23.6.3)$$

where pressure is integrated incrementally.

23.7 Material Model 7: Continuum Rubber

The hyperelastic continuum rubber model was studied by Blatz and Ko [1962]. In this model, the second Piola-Kirchhoff stress is given by

$$S_{ij} = G \left(V^{-1} C_{ij} - V^{-\frac{1}{1-2\nu}} \delta_{ij} \right), \quad (23.7.1)$$

where G is the shear modulus, V is the relative volume, ν is Poisson's ratio, and C_{ij} is the right Cauchy-Green strain:

$$C_{ij} = \frac{\partial x_k}{\partial X_i} \frac{\partial x_k}{\partial X_j}, \quad (23.7.2)$$

after determining S_{ij} , it is transformed into the Cauchy stress tensor, σ_{ij} :

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl}, \quad (23.7.3)$$

where ρ_0 and ρ are the initial and current density, respectively. The default value of ν is 0.463.

23.8 Material Model 8: Explosive Burn

Burn fractions, which multiply the equations of states for high explosives, control the release of chemical energy for simulating detonations. In the initialization phase, a lighting time t_1 is computed for each element by dividing the distance from the detonation point to the center of the element by the detonation velocity D . If multiple detonation points are defined, the closest point determines t_1 . The burn fraction F is taken as the maximum

$$F = \max(F_1, F_2), \quad (23.8.1)$$

where

$$F_1 = \begin{cases} \frac{2}{3} \frac{(t - t_1)D}{\left(\frac{v_e}{A_{e_{\max}}}\right)} & t > t_1 \\ 0 & t \leq t_1 \end{cases} \quad (23.8.2)$$

$$F_2 = \frac{1 - V}{1 - V_{CJ}}, \quad (23.8.3)$$

where V_{CJ} is the Chapman-Jouguet relative volume and t is current time. If F exceeds 1, it is reset to 1. This calculation of the burn fraction usually requires several time steps for F to reach unity, thereby spreading the burn front over several elements. After reaching unity, F is held constant. This burn fraction calculation is based on work by Wilkins [1964] and is also discussed by Giroux [1973].

As an option, the high explosive material can behave as an elastic perfectly-plastic solid prior to detonation. In this case we update the stress tensor, to an elastic trial stress, s_{ij}^{*n+1} ,

$$s_{ij}^{*n+1} = s_{ij}^n + s_{ip}\Omega_{pj} + s_{jp}\Omega_{pi} + 2G\dot{\epsilon}'_{ij}dt, \quad (23.8.4)$$

where G is the shear modulus, and $\dot{\epsilon}'_{ij}$ is the deviatoric strain rate. The von Mises yield condition is given by:

$$\phi = J_2 - \frac{\sigma_y^2}{3}, \quad (23.8.5)$$

where the second stress invariant, J_2 , is defined in terms of the deviatoric stress components as

$$J_2 = \frac{1}{2} s_{ij}s_{ij}, \quad (23.8.6)$$

and the yield stress is σ_y . If yielding has occurred, i.e., $\phi > 0$, the deviatoric trial stress is scaled to obtain the final deviatoric stress at time $n + 1$:

$$s_{ij}^{n+1} = \frac{\sigma_y}{\sqrt{3}J_2} s_{ij}^{*n+1}, \quad (23.8.7)$$

If $\phi \leq 0$, then

$$s_{ij}^{n+1} = s_{ij}^{*n+1}. \quad (23.8.8)$$

Before detonation pressure is given by the expression

$$p^{n+1} = K \left(\frac{1}{V^{n+1}} - 1 \right). \quad (23.8.9)$$

where K is the bulk modulus. Once the explosive material detonates:

$$s_{ij}^{n+1} = 0. \quad (23.8.10)$$

and the material behaves like a gas.

The shadow burn option should be active when computing the lighting time if there exist elements within the mesh for which there is no direct line of sight from the detonation points. The shadow burn option is activated in the control section. The lighting time is based on the shortest distance through the explosive material. If inert obstacles exist within the explosive material, the lighting time will account for the extra time required for the detonation wave to travel around the obstacles. The lighting times also automatically accounts for variations in the detonation velocity if different explosives are used. No additional input is required for the shadow option but care must be taken when setting up the input. This option works for two and three-dimensional solid elements. It is recommended that for best results:

1. Keep the explosive mesh as uniform as possible with elements of roughly the same dimensions.
2. Inert obstacle such as wave shapers within the explosive must be somewhat larger than the characteristic element dimension for the automatic tracking to function properly. Generally, a factor of two should suffice. The characteristic element dimension is found by checking all explosive elements for the largest diagonal
3. The detonation points should be either within or on the boundary of the explosive. Offset points may fail to initiate the explosive.
4. Check the computed lighting times in the post processor LS-PrePost. The lighting times may be displayed at time = 0, state 1, by plotting component 7 (a component normally reserved for plastic strain) for the explosive material. The lighting times are stored as negative numbers. The negative lighting time is replaced by the burn fraction when the element ignites.

5. Line detonations may be approximated by using a sufficient number of detonation points to define the line. Too many detonation points may result in significant initialization cost.

23.9 Material Model 9: Null Material

For solid elements equations of state can be called through this model to avoid deviatoric stress calculations. A pressure cutoff may be specified to set a lower bound on the pressure. This model has been very useful when combined with the reactive high-explosive model, where material strength is often neglected. The null material should not be used to delete solid elements.

An optional viscous stress of the form

$$\sigma_{ij} = 2\mu\dot{\epsilon}'_{ij}, \quad (23.9.1)$$

is computed for nonzero μ where $\dot{\epsilon}'_{ij}$ is the deviatoric strain rate and μ is the dynamic viscosity.

Sometimes it is advantageous to model contact surfaces via shell elements which are not part of the structure but are necessary to define areas of contact within nodal rigid bodies or between nodal rigid bodies. Beams and shells that use this material type are completely bypassed in the element processing. The Young's modulus and Poisson's ratio are used only for setting the contact interface stiffnesses, and it is recommended that reasonable values be input.

23.10 Material Model 10: Elastic-Plastic-Hydrodynamic

For completeness we give the entire derivation of this constitutive model based on radial return plasticity.

The pressure, p , deviatoric strain rate, $\dot{\epsilon}'_{ij}$, deviatoric stress rate, \dot{s}_{ij} , volumetric strain rate, and $\dot{\epsilon}_v$, are defined in [Equation \(1.1.1\)](#):

$$\begin{aligned} p &= -\frac{1}{3} \sigma_{ij} \delta_{ij} & \dot{\epsilon}'_{ij} &= \dot{\epsilon}_{ij} - \frac{1}{3} \dot{\epsilon}_v \\ s_{ij} &= \sigma_{ij} + p \delta_{ij} & \dot{\epsilon}_v &= \dot{\epsilon}_{ij} \delta_{ij} \\ s_{ij}^{\nabla} &= 2\mu \dot{\epsilon}'_{ij} = 2G \dot{\epsilon}'_{ij}. \end{aligned} \quad (23.10.1)$$

The Jaumann rate of the deviatoric stress, s_{ij}^{∇} , is given by:

$$s_{ij}^{\nabla} = \dot{s}_{ij} - s_{ip} \Omega_{pj} - s_{jp} \Omega_{pi}. \quad (23.10.2)$$

First we update s_{ij}^n to s_{ij}^{n+1} elastically

$$^*s_{ij}^{n+1} = s_{ij}^n + s_{ip} \Omega_{pj} + s_{jp} \Omega_{pi} + 2G \dot{\epsilon}'_{ij} dt = \underbrace{s_{ij}^n + R_{ij}}_{s_{ij}^{R^n}} + \underbrace{2G \dot{\epsilon}'_{ij} dt}_{2G \Delta \epsilon'_{ij}}, \quad (23.10.3)$$

where the left superscript, $*$, denotes a trial stress value. The effective trial stress is defined by

$$s^* = \left(\frac{3}{2} {}^*s_{ij}^{n+1} {}^*s_{ij}^{n+1} \right)^{1/2}, \quad (23.10.4)$$

and if s^* exceeds yield stress σ_y , the Von Mises flow rule:

$$\phi = \frac{1}{2} s_{ij} s_{ij} - \frac{\sigma_y^2}{3} \leq 0, \quad (23.10.5)$$

is violated and we scale the trial stresses back to the yield surface, i.e., a radial return

$$s_{ij}^{n+1} = \frac{\sigma_y}{s^*} {}^*s_{ij}^{n+1} = m^* s_{ij}^{n+1}. \quad (23.10.6)$$

The plastic strain increment can be found by subtracting the deviatoric part of the strain increment that is elastic, $\frac{1}{2G} (s_{ij}^{n+1} - s_{ij}^{R^n})$, from the total deviatoric increment, $\Delta \epsilon'_{ij}$, i.e.,

$$\Delta \epsilon_{ij}^p = \Delta \epsilon'_{ij} - \frac{1}{2G} (s_{ij}^{n+1} - s_{ij}^{R^n}). \quad (23.10.7)$$

Recalling that,

$$\Delta \varepsilon'_{ij} = \frac{(*s_{ij}^{n+1} - s_{ij}^{R^n})}{2G}, \quad (23.10.8)$$

and substituting Equation (23.10.8) into (23.10.7) we obtain,

$$\Delta \varepsilon^P_{ij} = \frac{(*s_{ij}^{n+1} - s_{ij}^{n+1})}{2G}. \quad (23.10.9)$$

Substituting Equation (23.10.6)

$$s_{ij}^{n+1} = m^* s_{ij}^{n+1}, \quad (23.10.10)$$

into Equation (23.10.9) gives,

$$\Delta \varepsilon^P_{ij} = \left(\frac{1-m}{2G} \right) *s_{ij}^{n+1} = \frac{1-m}{2Gm} s_{ij}^{n+1} = d\lambda s_{ij}^{n+1}. \quad (23.10.11)$$

By definition an increment in effective plastic strain is

$$\Delta \varepsilon^P = \left(\frac{2}{3} \Delta \varepsilon^P_{ij} \Delta \varepsilon^P_{ij} \right)^{1/2}. \quad (23.10.12)$$

Squaring both sides of Equation (23.10.11) leads to:

$$\Delta \varepsilon^P_{ij} \Delta \varepsilon^P_{ij} = \left(\frac{1-m}{2G} \right)^2 *s_{ij}^{n+1} *s_{ij}^{n+1} \quad (23.10.13)$$

or from Equations (23.10.4) and (23.10.12):

$$\frac{3}{2} \Delta \varepsilon^{P^2} = \left(\frac{1-m}{2G} \right)^2 \frac{2}{3} s^{*2} \quad (23.10.14)$$

Hence,

$$\Delta \varepsilon^P = \frac{1-m}{3G} s^* = \frac{s^* - \sigma_y}{3G} \quad (23.10.15)$$

where we have substituted for m from Equation (23.10.6)

$$m = \frac{\sigma_y}{s^*} \quad (23.10.16)$$

If isotropic hardening is assumed then:

$$\sigma_y^{n+1} = \sigma_y^n + E^P \Delta \varepsilon^P \quad (23.10.17)$$

and from Equation (23.10.15)

$$\Delta \varepsilon^P = \frac{(s^* - \sigma_y^{n+1})}{3G} = \frac{(s^* - \sigma_y^n - E^P \Delta \varepsilon^P)}{3G}. \quad (23.10.18)$$

Thus,

$$(3G + E^P) \Delta \varepsilon^P = (s^* - \sigma_y^n), \quad (23.10.19)$$

and solving for the incremental plastic strain gives

$$\Delta \epsilon^P = \frac{(s^* - \sigma_y^n)}{(3G + E^P)}. \quad (23.10.20)$$

The algorithm for plastic loading can now be outlined in five simple steps. If the effective trial stress exceeds the yield stress then

1. Solve for the plastic strain increment:

$$\Delta \epsilon^P = \frac{(s^* - \sigma_y^n)}{(3G + E^P)}. \quad (23.10.21)$$

2. Update the plastic strain:

$$\epsilon^{P^{n+1}} = \epsilon^{P^n} + \Delta \epsilon^P. \quad (23.10.22)$$

3. Update the yield stress:

$$\sigma_y^{n+1} = \sigma_y^n + E^P \Delta \epsilon^P. \quad (23.10.23)$$

4. Compute the scale factor using the yield strength at time $n + 1$:

$$m = \frac{\sigma_y^{n+1}}{s^*}. \quad (23.10.24)$$

5. Radial return the deviatoric stresses to the yield surface:

$$s_{ij}^{n+1} = m^* s_{ij}^{n+1}. \quad (23.10.25)$$

23.11 Material Model 11: Elastic-Plastic With Thermal Softening

Steinberg and Guinan [1978] developed this model for treating plasticity at high strain rates (10^5 per second) where enhancement of the yield strength due to strain rate effects is saturated out.

Both the shear modulus G and yield strength σ_y increase with pressure but decrease with temperature. As a melt temperature is reached, these quantities approach zero. We define the shear modulus before the material melts as

$$G = G_0 \left[1 + bpV^{1/3} - h \left(\frac{E - E_c}{3R'} - 300 \right) \right] e^{-\frac{fE}{E_m - E}}, \quad (23.11.1)$$

where G_0 , b , h , and f are input parameters, E_c is the cold compression energy:

$$E_c(X) = \int_0^x p dx - \frac{900R' \exp(ax)}{(1 - X)^{2(\gamma_0 - a - \frac{1}{2})}}, \quad (23.11.2)$$

where,

$$X = 1 - V, \quad (23.11.3)$$

and E_m is the melting energy:

$$E_m(X) = E_c(X) + 3R'T_m(X), \quad (23.11.4)$$

which is a function of the melting temperature $T_m(X)$:

$$T_m(X) = \frac{T_{m0} \exp(2aX)}{(1 - X)^{2(\gamma_0 - a - \frac{1}{3})}}, \quad (23.11.5)$$

and the melting temperature T_{m0} at $\rho = \rho_0$. The constants γ_0 and a are input parameters. In the above equation, R' is defined by

$$R' = \frac{R\rho_0}{A}, \quad (23.11.6)$$

where R is the gas constant and A is the atomic weight. The yield strength σ_y is given by:

$$\sigma_y = \sigma'_0 \left[1 + b'pV^{\frac{1}{3}} - h \left(\frac{E - E_c}{3R'} - 300 \right) \right] e^{-\frac{fE}{E_m - E}}. \quad (23.11.7)$$

If E_m exceeds E_i . Here, σ'_0 is given by:

$$\sigma'_0 = \sigma_0 [1 + \beta(\gamma_i + \bar{\epsilon}^p)]^n. \quad (23.11.8)$$

where γ_1 is the initial plastic strain, and b' and σ'_0 are input parameters. Where σ'_0 exceeds σ_{\max} , the maximum permitted yield strength, σ'_0 is set to equal to σ_{\max} . After the material melts, σ_y and G are set to zero.

LS-DYNA fits the cold compression energy to a ten-term polynomial expansion:

$$E_c = \sum_{i=0}^9 EC_i \eta^i, \quad (23.11.9)$$

where EC_i is the i^{th} coefficient and $\eta = \frac{\rho}{\rho_0}$. The least squares method is used to perform the fit [Kreyszig 1972]. The ten coefficients may also be specified in the input.

Once the yield strength and shear modulus are known, the numerical treatment is similar to that for material model 10.

23.12 Material Model 12: Isotropic Elastic-Plastic

The von Mises yield condition is given by:

$$\phi = J_2 - \frac{\sigma_y^2}{3}, \quad (23.12.1)$$

where the second stress invariant, J_2 , is defined in terms of the deviatoric stress components as

$$J_2 = \frac{1}{2} s_{ij} s_{ij}, \quad (23.12.2)$$

and the yield stress, σ_y , is a function of the effective plastic strain, $\varepsilon_{\text{eff}}^p$, and the plastic hardening modulus, E_p :

$$\sigma_y = \sigma_0 + E_p \varepsilon_{\text{eff}}^p. \quad (23.12.3)$$

The effective plastic strain is defined as

$$\varepsilon_{\text{eff}}^p = \int_0^t d\varepsilon_{\text{eff}}^p, \quad (23.12.4)$$

where $d\varepsilon_{\text{eff}}^p = \sqrt{\frac{2}{3} d\varepsilon_{ij}^p d\varepsilon_{ij}^p}$, and the plastic tangent modulus is defined in terms of the input tangent modulus, E_t , as

$$E_p = \frac{EE_t}{E - E_t}. \quad (23.12.5)$$

Pressure is given by the expression

$$p^{n+1} = K \left(\frac{1}{V^{n+1}} - 1 \right), \quad (23.12.6)$$

where K is the bulk modulus. This is perhaps the most cost effective plasticity model. Only one history variable, $\varepsilon_{\text{eff}}^p$, is stored with this model.

This model is not recommended for shell elements. In the plane stress implementation, a one-step radial return approach is used to scale the Cauchy stress tensor to if the state of stress exceeds the yield surface. This approach to plasticity leads to inaccurate shell thickness updates and stresses after yielding. This is the only model in LS-DYNA for plane stress that does not default to an iterative approach.

23.13 Material Model 13: Isotropic Elastic-Plastic with Failure

This highly simplistic failure model is occasionally useful. Material model 12 is called to update the stress tensor. Failure is initially assumed to occur if either

$$p^{n+1} < p_{\min}, \quad (23.13.1)$$

or

$$\varepsilon_{\text{eff}}^p > \varepsilon_{\max}^p, \quad (23.13.2)$$

where p_{\min} and ε_{\max}^p are user-defined parameters. Once failure has occurred, pressure may never be negative and the deviatoric components are set to zero:

$$s_{ij} = 0 \quad (23.13.3)$$

for all time. The failed element can only carry loads in compression.

23.14 Material Model 14: Soil and Crushable Foam With Failure

This material model provides the same stress update as model 5. However, if pressure ever reaches its cutoff value, failure occurs and pressure can never again go negative. In material model 5, the pressure is limited to its cutoff value in tension.

23.15 Material Model 15: Johnson and Cook Plasticity Model

Johnson and Cook express the flow stress as

$$\sigma_y = (A + B\bar{\epsilon}^n)(1 + c \ln \dot{\epsilon}^*)(1 - T^{*m}), \quad (23.15.1)$$

where A , B , C , n , and m are user defined input constants, and:

$$\begin{aligned} \bar{\epsilon}^p &= \text{effective plastic strain} \\ \dot{\epsilon}^* &= \frac{\dot{\bar{\epsilon}}^p}{\dot{\epsilon}_0} = \text{effective plastic strain rate for } \dot{\epsilon}_0, \text{ in units of } \frac{1}{[\text{time}]} \\ T^* &= \frac{T - T_{\text{room}}}{T_{\text{melt}} - T_{\text{room}}} \end{aligned}$$

Constants for a variety of materials are provided in Johnson and Cook [1983].

Due to the nonlinearity in the dependence of flow stress on plastic strain, an accurate value of the flow stress requires iteration for the increment in plastic strain. However, by using a Taylor series expansion with linearization about the current time, we can solve for σ_y with sufficient accuracy to avoid iteration.

The strain at fracture is given by

$$\epsilon^f = [D_1 + D_2 \exp(D_3 \sigma^*)][1 + D_4 \ln \dot{\epsilon}^*][1 + D_5 T^*], \quad (23.15.2)$$

where D_i , $i = 1, \dots, 5$ are input constants and σ^* is the ratio of pressure divided by effective stress:

$$\sigma^* = \frac{p}{\sigma_{\text{eff}}}. \quad (23.15.3)$$

Fracture occurs when the damage parameter

$$D = \sum \frac{\Delta \bar{\epsilon}^p}{\epsilon^f} \quad (23.15.4)$$

reaches the value 1.

A choice of three spall models is offered to represent material splitting, cracking, and failure under tensile loads. The pressure limit model limits the minimum hydrostatic pressure to the specified value, $p \geq p_{\text{min}}$. If pressures more tensile than this limit are calculated, the pressure is reset to p_{min} . This option is not strictly a spall model since the deviatoric stresses are unaffected by the pressure reaching the tensile cutoff and the pressure cutoff value p_{min} remains unchanged throughout the analysis. The maximum principal stress spall model detects spall if the maximum principal stress, σ_{max} , exceeds the limiting value σ_p . Once spall is detected with this model, the deviatoric stresses are reset to zero and no hydrostatic tension is permitted. If tensile pressures are calculated, they

are reset to 0 in the spalled material. Thus, the spalled material behaves as rubble. The hydrostatic tension spall model detects spall if the pressure becomes more tensile than the specified limit, p_{\min} . Once spall is detected, the deviatoric stresses are set to zero and the pressure is required to be compressive. If hydrostatic tension is calculated then the pressure is reset to 0 for that element.

In addition to the above failure criterion, this material model also supports a shell element deletion criterion based on the maximum stable time step size for the element, Δt_{\max} . Generally, Δt_{\max} goes down as the element becomes more distorted. To assure stability of time integration, the global LS-DYNA time step is the minimum of the Δt_{\max} values calculated for all elements in the model. Using this option allows the selective deletion of elements whose time step Δt_{\max} has fallen below the specified minimum time step, Δt_{crit} . Elements which are severely distorted often indicate that material has failed and supports little load, but these same elements may have very small time steps and therefore control the cost of the analysis. This option allows these highly distorted elements to be deleted from the calculation, and, therefore, the analysis can proceed at a larger time step, and, thus, at a reduced cost. Deleted elements do not carry any load, and are deleted from all applicable slide surface definitions. Clearly, this option must be judiciously used to obtain accurate results at a minimum cost.

Material type 15 is applicable to the high rate deformation of many materials including most metals. Unlike the Steinberg-Guinan model, the Johnson-Cook model remains valid down to lower strain rates and even into the quasistatic regime. Typical applications include explosive metal forming, ballistic penetration, and impact.

23.16 Material Model 16: Pseudo Tensor

This model can be used in two major modes - a simple tabular pressure-dependent yield surface, and a potentially complex model featuring two yield versus pressure functions with the means of migrating from one curve to the other. For both modes, load curve N1 is taken to be a strain rate multiplier for the yield strength. Note that this model must be used with equation-of-state type 8 or 9.

Response Mode I. Tabulated Yield Stress Versus Pressure

This model is well suited for implementing standard geologic models like the Mohr-Coulomb yield surface with a Tresca limit, as shown in [Figure 23.16.1](#). Examples of converting conventional triaxial compression data to this type of model are found in (Desai and Siriwardane, 1984). Note that under conventional triaxial compression conditions, the LS-DYNA input corresponds to an ordinate of $\sigma_1 - \sigma_3$ rather than the more widely used $\frac{\sigma_1 - \sigma_3}{2}$, where σ_1 is the maximum principal stress and σ_3 is the minimum principal stress.

This material combined with equation-of-state type 9 (saturated) has been used very successfully to model ground shocks and soil-structure interactions at pressures up to 100kbar.

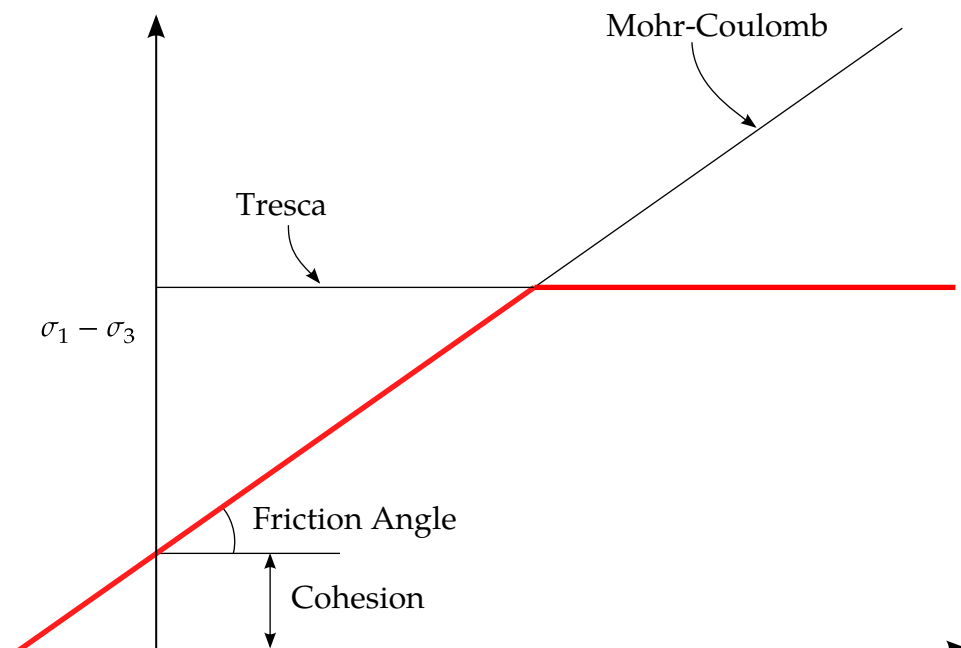


Figure 23.16.1. Mohr-Coulomb surface with a Tresca limit.

To invoke Mode I of this model, set a_0 , a_1 , a_2 , a_{0f} , and a_{1f} to zero. The tabulated values of pressure should then be specified on cards 4 and 5, and the corresponding values of yield stress should be specified on cards 6 and 7. The parameters relating to reinforcement properties, initial yield stress, and tangent modulus are not used in this response mode, and should be set to zero.

Simple tensile failure

Note that a_{1f} is reset internally to $1/3$ even though it is input as zero; this defines a material failure curve of slope $3p$, where p denotes pressure (positive in compression). In this case the yield strength is taken from the tabulated yield vs. pressure curve until the maximum principal stress (σ_1) in the element exceeds the tensile cut-off (σ_{cut}). For every time step that $\sigma_1 > \sigma_{\text{cut}}$ the yield strength is scaled back by a fraction of the distance between the two curves until after 20 time steps the yield strength is defined by the failure curve. The only way to inhibit this feature is to set σ_{cut} arbitrarily large.

Response Mode II. Two-Curve Model with Damage and Failure

This approach uses two yield versus pressure curves of the form

$$\sigma_y = a_0 + \frac{p}{a_1 + a_2 p}. \quad (23.16.1)$$

The upper curve is best described as the maximum yield strength curve and the lower curve is the material failure curve. There are a variety of ways of moving between the two curves and each is discussed below.

MODE II.A: Simple tensile failure

Define a_0 , a_1 , a_2 , a_{0f} and a_{1f} , set b_1 to zero, and leave cards 4 through 7 blank. In this case the yield strength is taken from the maximum yield curve until the maximum

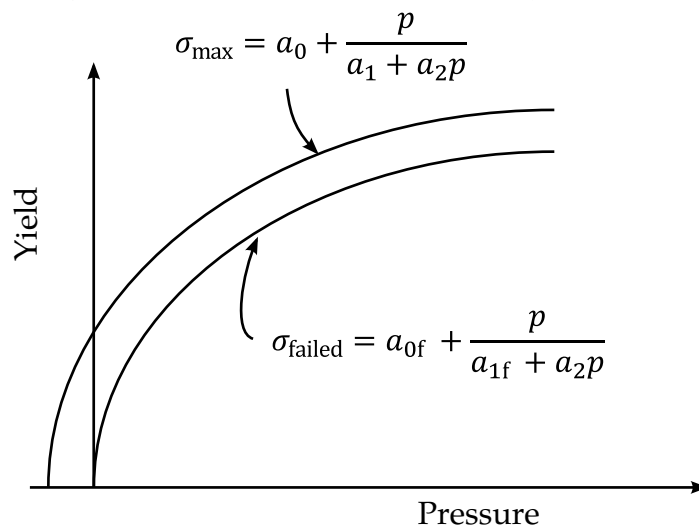


Figure 23.2. Two-curve concrete model with damage and failure.

principal stress (σ_1) in the element exceeds the tensile cut-off (σ_{cut}). For every time step

that $\sigma_1 > \sigma_{\text{cut}}$ the yield strength is scaled back by a fraction of the distance between the two curves until after 20 time steps the yield strength is defined by the failure curve.

Mode II.B: Tensile failure plus plastic strain scaling

Define a_0, a_1, a_2, a_{0f} and a_{1f} , set b_1 to zero, and user cards 4 through 7 to define a scale factor, η , versus effective plastic strain. LS-DYNA evaluates η at the current effective plastic strain and then calculated the yield stress as

$$\sigma_{\text{yield}} = \sigma_{\text{failed}} + \eta(\sigma_{\text{max}} - \sigma_{\text{failed}}), \quad (23.16.2)$$

where σ_{max} and σ_{failed} are found as shown in Figure 19.16.2. This yield strength is then subject to scaling for tensile failure as described above. This type of model allows the description of a strain hardening or softening material such as concrete.

Mode II.C: Tensile failure plus damage scaling

The change in yield stress as a function of plastic strain arises from the physical mechanisms such as internal cracking, and the extent of this cracking is affected by the hydrostatic pressure when the cracking occurs. This mechanism gives rise to the "confinement" effect on concrete behavior. To account for this phenomenon, a "damage" function was defined and incorporated. This damage function is given the form:

$$\lambda = \int_0^{\varepsilon^P} \left(1 + \frac{p}{\sigma_{\text{cut}}}\right)^{-b_1} d\varepsilon^P. \quad (23.16.3)$$

Define a_0, a_1, a_2, a_{0f} and a_{1f} , and b_1 . Cards 4 through 7 now give η as a function of λ and scale the yield stress as

$$\sigma_{\text{yield}} = \sigma_{\text{failed}} + \eta(\sigma_{\text{max}} - \sigma_{\text{failed}}), \quad (23.16.4)$$

and then apply any tensile failure criteria.

Mode II Concrete Model Options

Material Type 16 Mode II provides the option of automatic internal generation of a simple "generic" model for concrete. If a_0 is negative, then σ_{cut} is assumed to be the unconfined concrete compressive strength, f'_c and $-a_0$ is assumed to be a conversion factor from LS-DYNA pressure units to psi. (For example, if the model stress units are MPa, a_0 should be set to -145.) In this case the parameter values generated internally are

$$\begin{aligned}
\sigma_{\text{cut}} &= 1.7 \left(\frac{f'_c{}^2}{-a_0} \right)^{\frac{1}{3}} \\
a_0 &= \frac{f'_c}{4} \\
a_1 &= \frac{1}{3} \\
a_2 &= \frac{1}{3f'_c} \\
a_{0f} &= 0 \\
a_{1f} &= 0.385
\end{aligned} \tag{23.16.5}$$

Note that these a_{0f} and a_{1f} defaults will be overwritten by non-zero entries on Card 3. If plastic strain or damage scaling is desired, Cards 5 through 8 and b_1 should be specified in the input. When a_0 is input as a negative quantity, the equation-of-state can be given as 0 and a trilinear EOS Type 8 model will be automatically generated from the unconfined compressive strength and Poisson's ratio. The EOS 8 model is a simple pressure versus volumetric strain model with no internal energy terms, and should give reasonable results for pressures up to 5kbar (approximately 72,500 psi).

Mixture model

A reinforcement fraction, f_r , can be defined along with properties of the reinforcing material. The bulk modulus, shear modulus, and yield strength are then calculated from a simple mixture rule, i.e., for the bulk modulus the rule gives:

$$K = (1 - f_r)K_m + f_r K_r, \tag{23.16.6}$$

where K_m and K_r are the bulk moduli for the geologic material and the reinforcing material, respectively. This feature should be used with caution. It gives an isotropic effect in the material instead of the true anisotropic material behavior. A reasonable approach would be to use the mixture elements only where reinforcing material exists and plain elements elsewhere. When the mixture model is being used, the strain rate multiplier for the principal material is taken from load curve N1 and the multiplier for the reinforcement is taken from load curve N2.

23.17 Material Model 17: Isotropic Elastic-Plastic With Oriented Cracks

This is an isotropic elastic-plastic material which includes a failure model with an oriented crack. The von Mises yield condition is given by:

$$\phi = J_2 - \frac{\sigma_y^2}{3}, \quad (23.17.1)$$

where the second stress invariant, J_2 , is defined in terms of the deviatoric stress components as

$$J_2 = \frac{1}{2} s_{ij} s_{ij}, \quad (23.17.2)$$

and the yield stress, σ_y , is a function of the effective plastic strain, ϵ_{eff}^p , and the plastic hardening modulus, E_p :

$$\sigma_y = \sigma_0 + E_p \epsilon_{\text{eff}}^p. \quad (23.17.3)$$

The effective plastic strain is defined as:

$$\epsilon_{\text{eff}}^p = \int_0^t d\epsilon_{\text{eff}}^p, \quad (23.17.4)$$

where $d\epsilon_{\text{eff}}^p = \sqrt{\frac{2}{3} d\epsilon_{ij}^p d\epsilon_{ij}^p}$, and the plastic tangent modulus is defined in terms of the input tangent modulus, E_t , as

$$E_p = \frac{EE_t}{E - E_t}. \quad (23.17.5)$$

Pressure in this model is found from evaluating an equation of state. A pressure cutoff can be defined such that the pressure is not allowed to fall below the cutoff value.

The oriented crack fracture model is based on a maximum principal stress criterion. When the maximum principal stress exceeds the fracture stress, σ_f , the element fails on a plane perpendicular to the direction of the maximum principal stress. The normal stress and the two shear stresses on that plane are then reduced to zero. This stress reduction is done according to a delay function that reduces the stresses gradually to zero over a small number of time steps. This delay function procedure is used to reduce the ringing that may otherwise be introduced into the system by the sudden fracture.

After a tensile fracture, the element will not support tensile stress on the fracture plane, but in compression will support both normal and shear stresses. The orientation

of this fracture surface is tracked throughout the deformation, and is updated to properly model finite deformation effects. If the maximum principal stress subsequently exceeds the fracture stress in another direction, the element fails isotropically. In this case the element completely loses its ability to support any shear stress or hydrostatic tension, and only compressive hydrostatic stress states are possible. Thus, once isotropic failure has occurred, the material behaves like a fluid.

This model is applicable to elastic or elastoplastic materials under significant tensile or shear loading when fracture is expected. Potential applications include brittle materials such as ceramics as well as porous materials such as concrete in cases where pressure hardening effects are not significant.

23.18 Material Model 18: Power Law Isotropic Plasticity

Elastoplastic behavior with isotropic hardening is provided by this model. The yield stress, σ_y , is a function of plastic strain and obeys the equation:

$$\sigma_y = k\varepsilon^n = k(\varepsilon_0 + \bar{\varepsilon}^p)^n, \quad (23.18.1)$$

where ε_0 is the elastic strain to initial yield and $\bar{\varepsilon}^p$ is the effective plastic strain (logarithmic).

A parameter, SIGY, in the input governs how ε_0 is determined. If SIGY is set to zero, ε_0 is found by solving for the intersection of the linearly elastic loading equation with the strain hardening equation:

$$\begin{aligned} \sigma &= E\varepsilon, \\ \sigma &= k\varepsilon^n, \end{aligned} \quad (23.18.2)$$

which gives ε_0 as:

$$\varepsilon_0 = \left(\frac{E}{k}\right)^{\frac{1}{n-1}}. \quad (23.18.3)$$

If SIGY is nonzero but less than 0.02, $\varepsilon_0 = \text{SIGY}$. If SIGY is nonzero and greater than 0.02, the following equation gives ε_0 :

$$\varepsilon_0 = \left(\frac{\sigma_{y,0}}{k}\right)^{\frac{1}{n}}. \quad (23.18.4)$$

where the initial yield stress, $\sigma_{y,0}$, is SIGY.

Strain rate is accounted for using the Cowper-Symonds model which scales the yield stress with the factor

$$1 + \left(\frac{\dot{\varepsilon}}{C}\right)^{1/p}, \quad (23.18.5)$$

where $\dot{\varepsilon}$ is the strain rate. A fully viscoplastic formulation is optional with this model which incorporates the Cowper-Symonds formulation within the yield surface. An additional cost is incurred but the improvement allows for dramatic results.

23.19 Material Model 19: Strain Rate Dependent Isotropic Plasticity

In this model, a load curve is used to describe the yield strength σ_0 as a function of effective strain rate $\dot{\bar{\epsilon}}$ where

$$\dot{\bar{\epsilon}} = \left(\frac{2}{3} \dot{\epsilon}'_{ij} \dot{\epsilon}'_{ij} \right)^{1/2}, \quad (23.19.1)$$

and the prime denotes the deviatoric component. The yield stress is defined as

$$\sigma_y = \sigma_0(\dot{\bar{\epsilon}}) + E_p \bar{\epsilon}^P. \quad (23.19.2)$$

where $\bar{\epsilon}^P$ is the effective plastic strain and E_p is given in terms of Young's modulus and the tangent modulus by

$$E_p = \frac{E E_t}{E - E_t}. \quad (23.19.3)$$

Both Young's modulus and the tangent modulus may optionally be made functions of strain rate by specifying a load curve ID giving their values as a function of strain rate. If these load curve ID's are input as 0, then the constant values specified in the input are used.

Note that all load curves used to define quantities as a function of strain rate must have the same number of points at the same strain rate values. This requirement is used to allow vectorized interpolation to enhance the execution speed of this constitutive model.

This model also contains a simple mechanism for modeling material failure. This option is activated by specifying a load curve ID defining the effective stress at failure as a function of strain rate. For solid elements, once the effective stress exceeds the failure stress the element is deemed to have failed and is removed from the solution. For shell elements the entire shell element is deemed to have failed if all integration points through the thickness have an effective stress that exceeds the failure stress. After failure the shell element is removed from the solution.

In addition to the above failure criterion, this material model also supports a shell element deletion criterion based on the maximum stable time step size for the element, Δt_{\max} . Generally, Δt_{\max} goes down as the element becomes more distorted. To assure stability of time integration, the global LS-DYNA time step is the minimum of the Δt_{\max} values calculated for all elements in the model. Using this option allows the selective deletion of elements whose time step Δt_{\max} has fallen below the specified minimum time step, Δt_{crit} . Elements which are severely distorted often indicate that material has failed

and supports little load, but these same elements may have very small time steps and therefore control the cost of the analysis. This option allows these highly distorted elements to be deleted from the calculation, and, therefore, the analysis can proceed at a larger time step, and, thus, at a reduced cost. Deleted elements do not carry any load, and are deleted from all applicable slide surface definitions. Clearly, this option must be judiciously used to obtain accurate results at a minimum cost.

23.20 Material Model 20: Rigid

The rigid material type 20 provides a convenient way of turning one or more parts comprised of beams, shells, or solid elements into a rigid body. Approximating a deformable body as rigid is a preferred modeling technique in many real world applications. For example, in sheet metal forming problems the tooling can properly and accurately be treated as rigid. In the design of restraint systems the occupant can, for the purposes of early design studies, also be treated as rigid. Elements which are rigid are bypassed in the element processing and no storage is allocated for storing history variables; consequently, the rigid material type is very cost efficient.

Two unique rigid part IDs may not share common nodes unless they are merged together using the rigid body merge option. A rigid body may be made up of disjoint finite element meshes, however. LS-DYNA assumes this is the case since this is a common practice in setting up tooling meshes in forming problems.

All elements which reference a given part ID corresponding to the rigid material should be contiguous, but this is not a requirement. If two disjoint groups of elements on opposite sides of a model are modeled as rigid, separate part ID's should be created for each of the contiguous element groups if each group is to move independently. This requirement arises from the fact that LS-DYNA internally computes the six rigid body degrees-of-freedom for each rigid body (rigid material or set of merged materials), and if disjoint groups of rigid elements use the same part ID, the disjoint groups will move together as one rigid body.

Inertial properties for rigid materials may be defined in either of two ways. By default, the inertial properties are calculated from the geometry of the constituent elements of the rigid material and the density specified for the part ID. Alternatively, the inertial properties and initial velocities for a rigid body may be directly defined, and this overrides data calculated from the material property definition and nodal initial velocity definitions.

Young's modulus, E , and Poisson's ratio, ν are used for determining sliding interface parameters if the rigid body interacts in a contact definition. Realistic values for these constants should be defined since unrealistic values may contribute to numerical problem in contact.

23.21 Material Model 21: Thermal Orthotropic Elastic

In the implementation for three-dimensional continua a total Lagrangian formulation is used. In this approach the material law that relates second Piola-Kirchhoff stress \mathbf{S} to the Green-St. Venant strain \mathbf{E} is

$$\mathbf{S} = \mathbf{C} \cdot \mathbf{E} = \mathbf{T}^T \mathbf{C}_1 \mathbf{T} \cdot \mathbf{E}, \quad (23.21.1)$$

where \mathbf{T} is the transformation matrix [Cook 1974].

$$\mathbf{T} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & (l_1 m_2 + l_2 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 l_2 + n_2 l_1) \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & (l_2 m_3 + l_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 l_3 + n_3 l_2) \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & (l_3 m_1 + l_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 l_1 + n_1 l_3) \end{bmatrix}, \quad (23.21.2)$$

l_i, m_i, n_i are the direction cosines

$$x'_i = l_i x_1 + m_i x_2 + n_i x_3 \quad \text{for } i = 1, 2, 3, \quad (23.21.3)$$

and x'_i denotes the material axes. The constitutive matrix \mathbf{C}_1 is defined in terms of the material axes as

$$\mathbf{C}_1^{-1} = \begin{bmatrix} \frac{1}{E_{11}} & -\frac{\nu_{21}}{E_{22}} & -\frac{\nu_{31}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_{11}} & \frac{1}{E_{22}} & -\frac{\nu_{32}}{E_{33}} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_{11}} & -\frac{\nu_{23}}{E_{22}} & \frac{1}{E_{33}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}} \end{bmatrix}, \quad (23.21.4)$$

where the subscripts denote the material axes, i.e.,

$$\nu_{ij} = \nu_{x'_i x'_j} \quad \text{and} \quad E_{ii} = E_{x'_i}. \quad (23.21.5)$$

Since \mathbf{C}_1 is symmetric

$$\frac{\nu_{12}}{E_{11}} = \frac{\nu_{21}}{E_{22}}, \text{ etc.} \quad (23.21.6)$$

The vector of Green-St. Venant strain components is

$$\mathbf{E}^T = [E_{11} \ E_{22} \ E_{33} \ E_{12} \ E_{23} \ E_{31}], \quad (23.21.7)$$

which include the local thermal strains which are integrated in time:

$$\begin{aligned} \varepsilon_{aa}^{n+1} &= \varepsilon_{aa}^n + \alpha_a (T^{n+1} - T^n), \\ \varepsilon_{bb}^{n+1} &= \varepsilon_{bb}^n + \alpha_b (T^{n+1} - T^n), \\ \varepsilon_{cc}^{n+1} &= \varepsilon_{cc}^n + \alpha_c (T^{n+1} - T^n). \end{aligned} \quad (23.21.8)$$

After computing S_{ij} we use Equation (18.32) to obtain the Cauchy stress. This model will predict realistic behavior for finite displacement and rotations as long as the strains are small.

For shell elements, the stresses are integrated in time and are updated in the corotational coordinate system. In this procedure the local material axes are assumed to remain orthogonal in the deformed configuration. This assumption is valid if the strains remain small.

23.22 Material Model 22: Chang-Chang Composite Failure Model

For shells, five material parameters are used in the three failure criteria. These are [Chang and Chang 1987a, 1987b]:

- S_1 , longitudinal tensile strength
- S_2 , transverse tensile strength
- S_{12} , shear strength
- C_2 , transverse compressive strength
- α , nonlinear shear stress parameter.

S_1 , S_2 , S_{12} , and C_2 are obtained from material strength measurement. α is defined by material shear stress-strain measurements. In plane stress, the strain is given in terms of the stress as

$$\begin{aligned}\varepsilon_1 &= \frac{1}{E_1}(\sigma_1 - \nu_1\sigma_2), \\ \varepsilon_2 &= \frac{1}{E_2}(\sigma_2 - \nu_2\sigma_1), \\ 2\varepsilon_{12} &= \frac{1}{G_{12}}\tau_{12} + \alpha\tau_{12}^3.\end{aligned}\tag{23.22.1}$$

The third equation defines the nonlinear shear stress parameter α . (Note: In the LS-DYNA implementation, the term $\alpha\tau_{12}^3$ in the third equation is neglected.) A fiber matrix shearing term augments each damage mode:

$$\bar{\tau} = \frac{\frac{\tau_{12}^2}{2G_{12}} + \frac{3}{4}\alpha\tau_{12}^4}{\frac{S_{12}^2}{2G_{12}} + \frac{3}{4}\alpha S_{12}^4},\tag{23.22.2}$$

which is the ratio of the shear stress to the shear strength.

When $\sigma_2 > 0$, the matrix cracking failure criteria is determined from

$$F_{\text{matrix}} = \left(\frac{\sigma_2}{S_2}\right)^2 + \bar{\tau},\tag{23.22.3}$$

where failure is assumed whenever $F_{\text{matrix}} > 1$. If $F_{\text{matrix}} > 1$, then the material constants E_2 , G_{12} , ν_1 , and ν_2 are set to zero.

The compression failure criteria is given as

$$F_{\text{comb}} = \left(\frac{\sigma_2}{2S_{12}} \right)^2 + \left[\left(\frac{C_2}{2S_{12}} \right)^2 - 1 \right] \frac{\sigma_2}{C_2} + \bar{\tau}, \quad (23.22.4)$$

where failure is assumed whenever $F_{\text{comb}} > 1$. If $F_{\text{comb}} > 1$, then the material constants E_2 , ν_1 , and ν_2 are set to zero.

The final failure mode is due to fiber breakage when $\sigma_1 > 0$

$$F_{\text{fiber}} = \left(\frac{\sigma_1}{S_1} \right)^2 + \bar{\tau}, \quad (23.22.5)$$

Failure is assumed whenever $F_{\text{fiber}} > 1$. If $F_{\text{fiber}} > 1$, then the constants E_1 , E_2 , G_{12} , ν_1 and ν_2 are set to zero.

For solids, a fourth failure mode corresponding to delamination is computed as

$$F_{\text{delam}} = \left(\frac{\max(0.0, \sigma_3)}{S_3} \right)^2 + \left(\frac{\tau_{23}}{S_{23}} \right)^2 + \left(\frac{\tau_{31}}{S_{31}} \right)^2$$

This involves three additional material parameters.

- S_3 , normal tensile strength
 - S_{23} , transverse shear strength
 - S_{31} , transverse shear strength
- (23.22.140)

For solids, the effect of $F_{\text{matrix}} > 1$ is that all stress components are reduced to zero. Similarly σ_2 and σ_4 are modified if $F_{\text{matrix}} > 1$; only σ_2 is modified if $F_{\text{comb}} > 1$; and σ_3 , σ_5 and σ_6 are modified if $F_{\text{delam}} > 1$. For all of these cases, σ_1 , σ_2 and σ_3 are only reduced when these stresses are tensile.

23.23 Material Model 23: Thermal Orthotropic Elastic with 12 Curves

In the implementation for three-dimensional continua a total Lagrangian formulation is used. In this approach the material law that relates second Piola-Kirchhoff stress \mathbf{S} to the Green-St. Venant strain \mathbf{E} is

$$\mathbf{S} = \mathbf{C} \cdot \mathbf{E} = \mathbf{T}^T \mathbf{C}_1 \mathbf{T} \cdot \mathbf{E}, \quad (23.23.1)$$

where \mathbf{T} is the transformation matrix [Cook 1974].

$$\mathbf{T} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & (l_1 m_2 + l_2 m_1) & (m_1 n_2 + m_2 n_1) & (n_1 l_2 + n_2 l_1) \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & (l_2 m_3 + l_3 m_2) & (m_2 n_3 + m_3 n_2) & (n_2 l_3 + n_3 l_2) \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & (l_3 m_1 + l_1 m_3) & (m_3 n_1 + m_1 n_3) & (n_3 l_1 + n_1 l_3) \end{bmatrix}, \quad (23.23.2)$$

l_i, m_i, n_i are the direction cosines

$$x'_i = l_i x_1 + m_i x_2 + n_i x_3 \quad \text{for } i = 1, 2, 3, \quad (23.23.3)$$

and x'_i denotes the material axes. The temperature dependent constitutive matrix \mathbf{C}_1 is defined in terms of the material axes as

$$\mathbf{C}_1^{-1} = \begin{bmatrix} \frac{1}{E_{11}(T)} & -\frac{v_{21}(T)}{E_{22}(T)} & -\frac{v_{31}(T)}{E_{33}(T)} & 0 & 0 & 0 \\ \frac{v_{12}(T)}{E_{11}(T)} & \frac{1}{E_{22}(T)} & -\frac{v_{32}(T)}{E_{33}(T)} & 0 & 0 & 0 \\ -\frac{v_{13}(T)}{E_{11}(T)} & -\frac{v_{23}(T)}{E_{22}(T)} & \frac{1}{E_{33}(T)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{12}(T)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{23}(T)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{31}(T)} \end{bmatrix}, \quad (23.23.4)$$

where the subscripts denote the material axes, i.e.,

$$v_{ij} = v_{x'_i x'_j} \quad \text{and} \quad E_{ii} = E_{x'_i}. \quad (23.23.5)$$

Since \mathbf{C}_1 is symmetric

$$\frac{v_{12}}{E_{11}} = \frac{v_{21}}{E_{22}}, \text{ etc.} \quad (23.23.6)$$

The vector of Green-St. Venant strain components is

$$\mathbf{E}^T = [E_{11} \ E_{22} \ E_{33} \ E_{12} \ E_{23} \ E_{31}], \quad (23.23.7)$$

which include the local thermal strains which are integrated in time:

$$\begin{aligned} \varepsilon_{aa}^{n+1} &= \varepsilon_{aa}^n + \alpha_a \left(T^{n+\frac{1}{2}} \right) [T^{n+1} - T^n], \\ \varepsilon_{bb}^{n+1} &= \varepsilon_{bb}^n + \alpha_b \left(T^{n+\frac{1}{2}} \right) [T^{n+1} - T^n], \\ \varepsilon_{cc}^{n+1} &= \varepsilon_{cc}^n + \alpha_c \left(T^{n+\frac{1}{2}} \right) [T^{n+1} - T^n]. \end{aligned} \quad (23.23.8)$$

After computing S_{ij} we use Equation (16.32) to obtain the Cauchy stress. This model will predict realistic behavior for finite displacement and rotations as long as the strains are small.

For shell elements, the stresses are integrated in time and are updated in the corotational coordinate system. In this procedure the local material axes are assumed to remain orthogonal in the deformed configuration. This assumption is valid if the strains remain small.

23.24 Material Model 24: Piecewise Linear Isotropic Plasticity

This plasticity treatment in this model is quite similar to Model 10, but unlike 10, it includes strain rate effects and does not use an equation of state. Deviatoric stresses are determined that satisfy the yield function

$$\phi = \frac{1}{2} s_{ij} s_{ij} - \frac{\sigma_y^2}{3} \leq 0, \quad (23.24.1)$$

where

$$\sigma_y = \beta [\sigma_0 + f_h(\epsilon_{\text{eff}}^p)], \quad (23.24.2)$$

where the hardening function $f_h(\epsilon_{\text{eff}}^p)$ can be specified in tabular form as an option. Otherwise, linear hardening of the form

$$f_h(\epsilon_{\text{eff}}^p) = E_p(\epsilon_{\text{eff}}^p), \quad (23.24.3)$$

is assumed where E_p and ϵ_{eff}^p are given in [Equations \(23.3.6\)](#) and [\(9.61\)](#), respectively. The parameter β accounts for strain rate effects. For complete generality a table defining the yield stress versus plastic strain may be defined for various levels of effective strain rate.

In the implementation of this material model, the deviatoric stresses are updated elastically (see material model 1), the yield function is checked, and if it is satisfied the deviatoric stresses are accepted. If it is not, an increment in plastic strain is computed:

$$\Delta \epsilon_{\text{eff}}^p = \frac{\left(\frac{3}{2} s_{ij}^* s_{ij}^* \right)^{1/2} - \sigma_y}{3G + E_p}, \quad (23.24.4)$$

is the shear modulus and E_p is the current plastic hardening modulus. The trial deviatoric stress state s_{ij}^* is scaled back:

$$s_{ij}^{n+1} = \frac{\sigma_y}{\left(\frac{3}{2} s_{ij}^* s_{ij}^* \right)^{1/2}} s_{ij}^*. \quad (23.24.5)$$

For shell elements, the above equations apply, but with the addition of an iterative loop to solve for the normal strain increment, such that the stress component normal to the mid surface of the shell element approaches zero.

Three options to account for strain rate effects are possible:

1. Strain rate may be accounted for using the Cowper-Symonds model which scales the yield stress with the factor

$$\beta = 1 + \left(\frac{\dot{\epsilon}}{C} \right)^{1/p} \quad (23.24.6)$$

where $\dot{\epsilon}$ is the strain rate.

2. For complete generality a load curve, defining β , which scales the yield stress may be input instead. In this curve the scale factor versus strain rate is defined.
3. If different stress versus strain curves can be provided for various strain rates, the option using the reference to a table definition can be used. See Figure 19.24.1.

A fully viscoplastic formulation is optional which incorporates the different options above within the yield surface. An additional cost is incurred over the simple scaling but the improvement in results can be dramatic.

If a table ID is specified a curve ID is given for each strain rate, see Section 23. Intermediate values are found by interpolating between curves. Effective plastic strain versus yield stress is expected. If the strain rate values fall out of range, extrapolation is not used; rather, either the first or last curve determines the yield stress depending on whether the rate is low or high, respectively.

23.25 Material Model 25: Kinematic Hardening Cap Model

The implementation of an extended two invariant cap model, suggested by Stojko [1990], is based on the formulations of Simo, et al. [1988, 1990] and Sandler and Rubin [1979]. In this model, the two invariant cap theory is extended to include nonlinear kinematic hardening as suggested by Isenberg, Vaughn, and Sandler [1978]. A brief discussion of the extended cap model and its parameters is given below.

The cap model is formulated in terms of the invariants of the stress tensor. The square root of the second invariant of the deviatoric stress tensor, $\sqrt{J_{2D}}$ is found from the deviatoric stresses \mathbf{S} as

$$\sqrt{J_{2D}} \equiv \sqrt{\frac{1}{2} s_{ij} s_{ij}}, \quad (23.25.1)$$

and is the objective scalar measure of the distortional or shearing stress. The first invariant of the stress, J_1 , is the trace of the stress tensor.

The cap model consists of three surfaces in $\sqrt{J_{2D}} - J_1$ space, as shown in [Figure 23.25.1](#). First, there is a failure envelope surface, denoted f_1 in the figure. The functional form of f_1 is

$$f_1 = \sqrt{J_{2D}} - \min(F_e(J_1), T_{\text{mises}}), \quad (23.25.2)$$

where F_e is given by

$$F_e(J_1) \equiv \alpha - \gamma \exp(-\beta J_1) + \theta J_1. \quad (23.25.3)$$

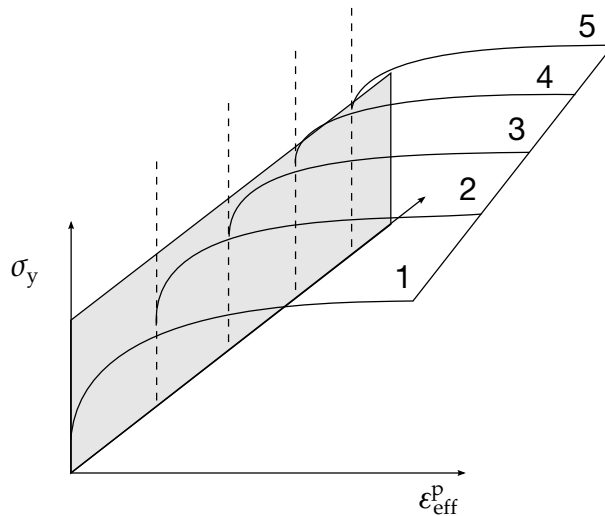


Figure 23.25.1. Rate effects may be accounted for by defining a table of curves.

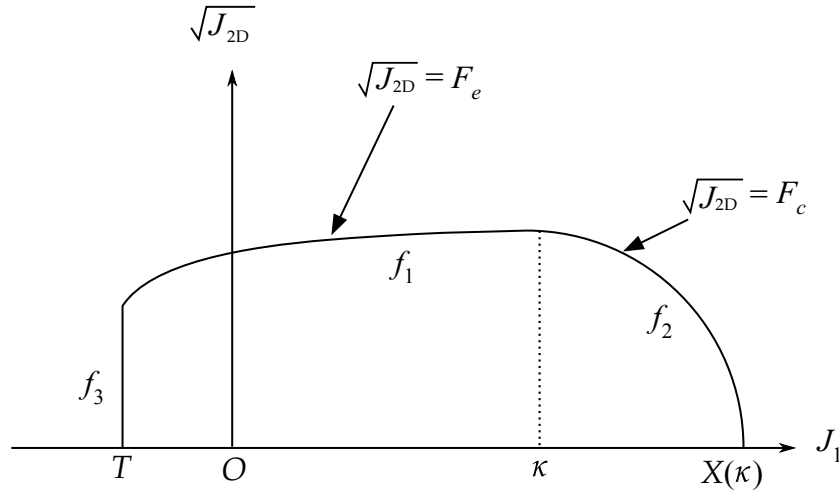


Figure 23.25.2. The yield surface of the two-invariant cap model in pressure $\sqrt{J_{2D}} - J_1$ space. Surface f_1 is the failure envelope, f_2 is the cap surface, and f_3 is the tension cutoff.

and $T_{\text{mises}} \equiv |X(\kappa_n) - L(\kappa_n)|$. This failure envelope surface is fixed in $\sqrt{J_{2D}} - J_1$ space, and therefore does not harden unless kinematic hardening is present. Next, there is a cap surface, denoted f_2 in the figure, with f_2 given by

$$f_2 = \sqrt{J_{2D}} - F_c(J_1, \kappa), \quad (23.25.4)$$

where F_c is defined by

$$F_c(J_1, \kappa) \equiv \frac{1}{R} \sqrt{[X(\kappa) - L(\kappa)]^2 - [J_1 - L(\kappa)]^2}, \quad (23.25.5)$$

$X(\kappa)$ is the intersection of the cap surface with the J_1 axis

$$X(\kappa) = \kappa + RF_e(\kappa), \quad (23.25.6)$$

and $L(\kappa)$ is defined by

$$L(\kappa) \equiv \begin{cases} \kappa & \kappa > 0 \\ 0 & \kappa \leq 0 \end{cases} \quad (23.25.7)$$

The hardening parameter κ is related to the plastic volume change ε_v^p through the hardening law

$$\varepsilon_v^p = W\{1 - \exp[-D(X(\kappa) - X_0)]\}. \quad (23.25.8)$$

Geometrically, κ is seen in the figure as the J_1 coordinate of the intersection of the cap surface and the failure surface. Finally, there is the tension cutoff surface, denoted f_3 in the figure. The function f_3 is given by

$$f_3 + T - J_1, \quad (23.25.9)$$

where T is the input material parameter which specifies the maximum hydrostatic tension sustainable by the material. The elastic domain in $\sqrt{J_{2D}} - J_1$ space is then bounded

by the failure envelope surface above, the tension cutoff surface on the left, and the cap surface on the right.

An additive decomposition of the strain into elastic and plastic parts is assumed:

$$\varepsilon = \varepsilon^e + \varepsilon^p, \quad (23.25.10)$$

where ε^e is the elastic strain and ε^p is the plastic strain. Stress is found from the elastic strain using Hooke's law,

$$\sigma = C(\varepsilon - \varepsilon^p), \quad (23.25.11)$$

where σ is the stress and C is the elastic constitutive tensor.

The yield condition may be written

$$\begin{aligned} f_1(\sigma) &\leq 0, \\ f_2(\sigma, \kappa) &\leq 0, \\ f_3(\sigma) &\leq 0, \end{aligned} \quad (23.25.12)$$

and the plastic consistency condition requires that

$$\begin{aligned} \dot{\lambda}_k f_k &= 0 \\ \dot{\lambda}_k &\geq 0 \end{aligned} \quad k = 1, 2, 3, \quad (23.25.13)$$

where λ_k is the plastic consistency parameter for surface k . If $f_k < 0$, then $\dot{\lambda}_k = 0$ and the response is elastic. If $f_k > 0$, then surface k is active and $\dot{\lambda}_k$ is found from the requirement that $\dot{f}_k = 0$.

Associated plastic flow is assumed, so using Koiter's flow rule the plastic strain rate is given as the sum of contribution from all of the active surfaces,

$$\dot{\varepsilon}^p = \sum_{k=1}^3 \dot{\lambda}_k \frac{\partial f_k}{\partial \varepsilon}. \quad (23.25.14)$$

One of the major advantages of the cap model over other classical pressure-dependent plasticity models is the ability to control the amount of dilatency produced under shear loading. Dilatency is produced under shear loading as a result of the yield surface having a positive slope in $\sqrt{J_{2D}} - J_1$ space, so the assumption of plastic flow in the direction normal to the yield surface produces a plastic strain rate vector that has a component in the volumetric (hydrostatic) direction (see [Figure 23.25.1](#)). In models such as the Drucker-Prager and Mohr-Coulomb, this dilatency continues as long as shear loads are applied, and in many cases produces far more dilatency than is experimentally observed in material tests. In the cap model, when the failure surface is active, dilatency is produced just as with the Drucker-Prager and Mohr-Coulomb models. However, the hardening law permits the cap surface to contract until the cap intersects the failure envelope at the stress point, and the cap remains at that point. The local normal to the yield surface is now vertical, and therefore the normality rule assures that no further plastic volumetric strain (dilatency) is created. Adjustment of the parameters that control the

rate of cap contractions permits experimentally observed amounts of dilatency to be incorporated into the cap model, thus producing a constitutive law which better represents the physics to be modeled. Another advantage of the cap model over other models such as the Drucker-Prager and Mohr-Coulomb is the ability to model plastic compaction. In these models all purely volumetric response is elastic. In the cap model, volumetric response is elastic until the stress point hits the cap surface. Therefore, plastic volumetric strain (compaction) is generated at a rate controlled by the hardening law. Thus, in addition to controlling the amount of dilatency, the introduction of the cap surface adds another experimentally observed response characteristic of geological material into the model.

The inclusion of kinematic hardening results in hysteretic energy dissipation under cyclic loading conditions. Following the approach of Isenberg, et al., [1978] a nonlinear kinematic hardening law is used for the failure envelope surface when nonzero values of α and N are specified. In this case, the failure envelope surface is replaced by a family of yield surfaces bounded by an initial yield surface and a limiting failure envelope surface. Thus, the shape of the yield surfaces described above remains unchanged, but they may translate in a plane orthogonal to the J axis.

Translation of the yield surfaces is permitted through the introduction of a “back stress” tensor, α . The formulation including kinematic hardening is obtained by replacing the stress σ with the translated stress tensor $\eta \equiv \sigma - \alpha$ in all of the above equation. The history tensor α is assumed deviatoric, and therefore has only 5 unique components. The evolution of the back stress tensor is governed by the nonlinear hardening law

$$\dot{\alpha} = \bar{c}\bar{F}(\sigma, \alpha) \dot{\epsilon}^P, \quad (23.25.15)$$

where \bar{c} is a constant, \bar{F} is a scalar function of σ and α and $\dot{\epsilon}^P$ is the rate of deviator plastic strain. The constant may be estimated from the slope of the shear stress - plastic shear strain curve at low levels of shear stress.

The function \bar{F} is defined as

$$\bar{F} \equiv \max \left(0, 1 - \frac{(\sigma - \alpha)\alpha}{2NF_e(J_1)} \right), \quad (23.25.16)$$

where N is a constant defining the size of the yield surface. The value of N may be interpreted as the radial distant between the outside of the initial yield surface and the inside of the limit surface. In order for the limit surface of the kinematic hardening cap model to correspond with the failure envelope surface of the standard cap model, the scalar parameter a must be replaced $\alpha - N$ in the definition F_e .

The cap model contains a number of parameters which must be chosen to represent a particular material, and are generally based on experimental data. The parameters α , β , θ and γ are usually evaluated by fitting a curve through failure data taken from a set of triaxial compression tests. The parameters W , D , and X_0 define the cap hardening law. The value W represents the void fraction of the uncompressed sample and D governs the

slope of the initial loading curve in hydrostatic compression. The value of R is the ration of major to minor axes of the quarter ellipse defining the cap surface. Additional details and guidelines for fitting the cap model to experimental data are found in [Chen and Baladi, 1985].

23.26 Material Model 26: Crushable Foam

This orthotropic material model does the stress update in the local material system denoted by the subscripts, a , b , and c . The material model requires the following input parameters:

- E , Young's modulus for the fully compacted material;
- ν , Poisson's ratio for the compacted material;
- σ_y , yield stress for fully compacted honeycomb;
- LCA, load curve number for sigma-aa versus either relative volume or volumetric strain (see Figure 19.26.1.);
- LCB, load curve number for sigma-bb versus either relative volume or volumetric strain (default: LCB = LCA);
- LCC, the load curve number for sigma-cc versus either relative volume or volumetric strain (default: LCC = LCA);
- LCS, the load curve number for shear stress versus either relative volume or volumetric strain (default LCS = LCA);
- V_f , relative volume at which the honeycomb is fully compacted;
- E_{aaU} , elastic modulus in the uncompressed configuration;
- E_{bbU} , elastic modulus in the uncompressed configuration;
- E_{ccU} , elastic modulus in the uncompressed configuration;
- G_{abU} , elastic shear modulus in the uncompressed configuration;
- G_{bcU} , elastic shear modulus in the uncompressed configuration;
- G_{caU} , elastic shear modulus in the uncompressed configuration;
- LCAB, load curve number for sigma-ab versus either relative volume or volumetric strain (default: LCAB = LCS);
- LCBC, load curve number for sigma-bc versus either relative volume or volumetric strain default: LCBC = LCS);
- LCCA, load curve number for sigma-ca versus either relative volume or volumetric strain (default: LCCA = LCS);
- LCSR, optional load curve number for strain rate effects.

The behavior before compaction is orthotropic where the components of the stress tensor are uncoupled, i.e., an a component of strain will generate resistance in the local a

direction with no coupling to the local b and c directions. The elastic moduli vary linearly with the relative volume from their initial values to the fully compacted values:

$$\begin{aligned} E_{aa} &= E_{aau} + \beta(E - E_{aau}), \\ E_{bb} &= E_{bbu} + \beta(E - E_{bbu}), \\ E_{cc} &= E_{ccu} + \beta(E - E_{ccu}), \\ G_{ab} &= G_{abu} + \beta(G - G_{abu}), \\ G_{bc} &= G_{bcu} + \beta(G - G_{bcu}), \\ G_{ca} &= G_{cau} + \beta(G - G_{cau}), \end{aligned} \quad (23.26.1)$$

where

$$\beta = \max \left[\min \left(\frac{1 - V_{\min}}{1 - V_f}, 1 \right), 0 \right], \quad (23.26.2)$$

and G is the elastic shear modulus for the fully compacted honeycomb material

$$G = \frac{E}{2(1 + \nu)}. \quad (23.26.3)$$

The relative volume V is defined as the ratio of the current volume over the initial volume; typically, $V = 1$ at the beginning of a calculation. The relative volume, V_{\min} , is the minimum value reached during the calculation.

The load curves define the magnitude of the average stress as the material changes density (relative volume). Each curve related to this model must have the same number

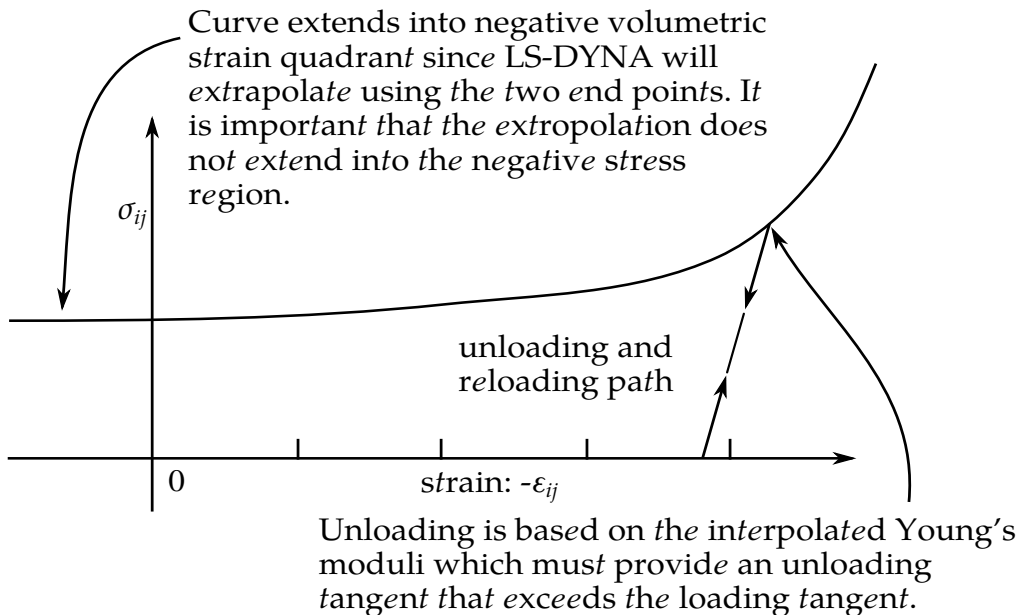


Figure 23.26.1. Stress quantity versus volumetric strain. Note that the “yield stress” at a volumetric strain of zero is nonzero. In the load curve definition, the “time” value is the volumetric strain and the “function” value is the yield stress.

of points and the same abscissa values. There are two ways to define these curves: as a function of relative volume V , or as a function of volumetric strain defined as:

$$\varepsilon_V = 1 - V. \quad (23.26.4)$$

In the former, the first value in the curve should correspond to a value of relative volume slightly less than the fully compacted value. In the latter, the first value in the curve should be less than or equal to zero corresponding to tension and should increase to full compaction. When defining the curves, care should be taken that the extrapolated values do not lead to negative yield stresses.

At the beginning of the stress update we transform each element's stresses and strain rates into the local element coordinate system. For the uncompacted material, the trial stress components are updated using the elastic interpolated moduli according to:

$$\begin{aligned} \sigma_{aa}^{n+1\text{trial}} &= \sigma_{aa}^n + E_{aa}\Delta\varepsilon_{aa}, \\ \sigma_{bb}^{n+1\text{trial}} &= \sigma_{bb}^n + E_{bb}\Delta\varepsilon_{bb}, \\ \sigma_{cc}^{n+1\text{trial}} &= \sigma_{cc}^n + E_{cc}\Delta\varepsilon_{cc}, \\ \sigma_{ab}^{n+1\text{trial}} &= \sigma_{ab}^n + 2G_{ab}\Delta\varepsilon_{ab}, \\ \sigma_{bc}^{n+1\text{trial}} &= \sigma_{bc}^n + 2G_{bc}\Delta\varepsilon_{bc}, \\ \sigma_{ca}^{n+1\text{trial}} &= \sigma_{ca}^n + 2G_{ca}\Delta\varepsilon_{ca} = 1. \end{aligned} \quad (23.26.5)$$

Then we independently check each component of the updated stresses to ensure that they do not exceed the permissible values determined from the load curves, e.g., if

$$|\sigma_{ij}^{n+1\text{trial}}| > \lambda\sigma_{ij}(V_{\min}), \quad (23.26.6)$$

then

$$\sigma_{ij}^{n+1} = \sigma_{ij}(V_{\min}) \frac{\lambda\sigma_{ij}^{n+1\text{trial}}}{|\sigma_{ij}^{n+1\text{trial}}|}. \quad (23.26.7)$$

The parameter λ is either unity or a value taken from the load curve number, LC-SR, that defines λ as a function of strain rate. Strain rate is defined here as the Euclidean norm of the deviatoric strain rate tensor.

For fully compacted material we assume that the material behavior is elastic-perfectly plastic and updated the stress components according to

$$s_{ij}^{\text{trial}} = s_{ij}^n + 2G\Delta\varepsilon_{ij}^{\text{dev}^{n+1/2}}, \quad (23.26.8)$$

where the deviatoric strain increment is defined as

$$\Delta \varepsilon_{ij}^{\text{dev}} = \Delta \varepsilon_{ij} - \frac{1}{3} \Delta \varepsilon_{kk} \delta_{ij}. \quad (23.26.9)$$

We next check to see if the yield stress for the fully compacted material is exceeded by comparing

$$s_{\text{eff}}^{\text{trial}} = \left(\frac{3}{2} s_{ij}^{\text{trial}} s_{ij}^{\text{trial}} \right)^{1/2}. \quad (23.26.10)$$

the effective trial stress, to the yield stress σ_y . If the effective trial stress exceeds the yield stress, we simply scale back the stress components to the yield surface:

$$s_{ij}^{n+1} = \frac{\sigma_y}{s_{\text{eff}}^{\text{trial}}} s_{ij}^{\text{trial}}. \quad (23.26.11)$$

We can now update the pressure using the elastic bulk modulus, K :

$$p^{n+1} = p^n - K \Delta \varepsilon_{kk}^{n+1/2},$$

$$K = \frac{E}{3(1-2\nu)}, \quad (23.26.12)$$

and obtain the final value for the Cauchy stress

$$\sigma_{ij}^{n+1} = s_{ij}^{n+1} - p^{n+1} \delta_{ij}. \quad (23.26.13)$$

After completing the stress update, we transform the stresses back to the global configuration.

23.27 Material Model 27: Incompressible Mooney-Rivlin Rubber

The Mooney-Rivlin material model is based on a strain energy function, W , as follows

$$W = A(I_1 - 3) + B(I_2 - 3) + C\left(\frac{1}{I_3^2} - 1\right) + D(I_3 - 1)^2. \quad (23.27.1)$$

A and B are user defined constants, whereas C and D are related to A and B as follows

$$\begin{aligned} C &= \frac{1}{2}A + B, \\ D &= \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}. \end{aligned} \quad (23.27.2)$$

The derivation of the constants C and D is straightforward [Feng, 1993] and is included here since we were unable to locate it in the literature. The principal components of Cauchy stress, σ_i , are given by [Ogden, 1984]

$$J\sigma_i = \lambda_i \frac{\partial W}{\partial \lambda_i}. \quad (23.27.3)$$

For uniform dilation

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda, \quad (23.27.4)$$

thus the pressure, p , is obtained (please note the sign convention),

$$p = \sigma_1 = \sigma_2 = \sigma_3 = \frac{2}{\lambda^3} \left(\lambda^2 \frac{\partial W}{\partial I_1} + 2\lambda^4 \frac{\partial W}{\partial I_2} + \lambda^6 \frac{\partial W}{\partial I_3} \right). \quad (23.27.5)$$

The relative volume, V , can be defined in terms of the stretches as:

$$V = \lambda^3 = \frac{\text{new volume}}{\text{old volume}}. \quad (23.27.6)$$

For small volumetric deformations the bulk modulus, K , can be defined as the ratio of the pressure over the volumetric strain as the relative volume approaches unity:

$$K = \lim_{V \rightarrow 1} \left(\frac{p}{V - 1} \right). \quad (23.27.7)$$

The partial derivatives of W lead to:

$$\begin{aligned}
\frac{\partial W}{\partial I_1} &= A, \\
\frac{\partial W}{\partial I_2} &= B, \\
\frac{\partial W}{\partial I_3} &= -2CI_3^{-3} + 2D(I_3 - 1) = -2C\lambda^{-18} + 2D(\lambda^6 - 1), \\
p &= \frac{2}{\lambda^3} \{A\lambda^2 + 2\lambda^4B + \lambda^6[-2C\lambda^{-18} + 2D(\lambda^6 - 1)]\} \\
&= \frac{2}{\lambda^3} \{A\lambda^2 + 2\lambda^4B - 2C\lambda^{-12} + 2D(\lambda^{12} - \lambda^6)\}.
\end{aligned} \tag{23.27.8}$$

In the limit as the stretch ratio approaches unity, the pressure must approach zero:

$$\lim_{\lambda \rightarrow 1} p = 0. \tag{23.27.9}$$

Therefore, $A + 2B - 2C = 0$ and

$$C = 0.5A + B. \tag{23.27.10}$$

To solve for D we note that:

$$\begin{aligned}
K &= \lim_{V \rightarrow 1} \left(\frac{p}{V - 1} \right) \\
&= \lim_{\lambda \rightarrow 1} \frac{\frac{2}{\lambda^3} \{A\lambda^2 + 2\lambda^4B - 2C\lambda^{-12} + 2D(\lambda^{12} - \lambda^6)\}}{\lambda^3 - 1} \\
&= 2 \lim_{\lambda \rightarrow 1} \frac{A\lambda^2 + 2\lambda^4B - 2C\lambda^{-12} + 2D(\lambda^{12} - \lambda^6)}{\lambda^6 - \lambda^3} \\
&= 2 \lim_{\lambda \rightarrow 1} \frac{2A\lambda + 8\lambda^3B + 24C\lambda^{-13} + 2D(12\lambda^{11} - 6\lambda^5)}{6\lambda^5 - 3\lambda^2} \\
&= \frac{2}{3} (2A + 8B + 24C + 12D) \\
&= \frac{2}{3} (14A + 32B + 12D).
\end{aligned} \tag{23.27.11}$$

We therefore obtain:

$$14A + 32B + 12D = \frac{3}{2}K = \frac{3}{2} \left(\frac{2G(1 + \nu)}{3(1 - 2\nu)} \right) = \frac{2(A + B)(1 + \nu)}{(1 - 2\nu)}. \tag{23.27.12}$$

Solving for D we obtain the desired equation:

$$D = \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}. \tag{23.27.13}$$

The invariants $I_1 - I_3$ are related to the right Cauchy-Green tensor \mathbf{C} as

$$\begin{aligned} I_1 &= C_{ii}, \\ I_2 &= \frac{1}{2} C_{ii}^2 - \frac{1}{2} C_{ij} C_{ij}, \\ I_3 &= \det(C_{ij}). \end{aligned} \quad (23.27.14)$$

The second Piola-Kirchhoff stress tensor, \mathbf{S} , is found by taking the partial derivative of the strain energy function with respect to the Green-Lagrange strain tensor, \mathbf{E} .

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} = 2 \frac{\partial W}{\partial C_{ij}} = 2 \left[A \frac{\partial I_1}{\partial C_{ij}} + B \frac{\partial I_2}{\partial C_{ij}} + \left(2D(I_3 - 1) - \frac{2C}{I_3^2} \right) \frac{\partial I_3}{\partial C_{ij}} \right]. \quad (23.27.15)$$

The derivatives of the invariants $I_1 - I_3$ are

$$\begin{aligned} \frac{\partial I_1}{\partial C_{ij}} &= \delta_{ij}, \\ \frac{\partial I_2}{\partial C_{ij}} &= I_1 \delta_{ij} - C_{ij}, \\ \frac{\partial I_3}{\partial C_{ij}} &= I_3 C_{ij}^{-1}. \end{aligned} \quad (23.27.16)$$

Inserting Equation (23.27.16) into Equation (23.27.15) yields the following expression for the second Piola-Kirchhoff stress:

$$S_{ij} = 2A \delta_{ij} + 2B(I_1 \delta_{ij} - C_{ij}) - 4C \frac{1}{I_3^2} C_{ij}^{-1} + 4D(I_3 - 1) I_3 C_{ij}^{-1}. \quad (23.27.17)$$

Equation (23.27.17) can be transformed into the Cauchy stress by using the push forward operation

$$\sigma_{ij} = \frac{1}{J} F_{ik} S_{kl} F_{jl}. \quad (23.27.18)$$

where $J = \det(F_{ij})$.

23.27.1 Stress Update for Shell Elements

As a basis for discussing the algorithmic tangent stiffness for shell elements in Section 19.27.3, the corresponding stress update as it is done in LS-DYNA is shortly recapitulated in this section. When dealing with shell elements, the stress (as well as constitutive matrix) is typically evaluated in corotational coordinates after which it is transformed back to the standard basis according to

$$\sigma_{ij} = R_{ik} R_{jl} \hat{\sigma}_{kl}. \quad (23.27.19)$$

Here R_{ij} is the rotation matrix containing the corotational basis vectors. The so-called corotated stress $\hat{\sigma}_{ij}$ is evaluated using Equation 19.27.21 with the exception that the deformation gradient is expressed in the corotational coordinates, i.e.,

$$\hat{\sigma}_{ij} = \frac{1}{J} \hat{F}_{ik} S_{kl} \hat{F}_{jl}, \quad (23.27.20)$$

where S_{ij} is evaluated using Equation (23.27.17). The corotated deformation gradient is incrementally updated with the aid of a time increment Δt , the corotated velocity gradient \hat{L}_{ij} , and the angular velocity $\hat{\Omega}_{ij}$ with which the embedded coordinate system is rotating.

$$\hat{F}_{ij} = (\delta_{ik} + \Delta t \hat{L}_{ik} - \Delta t \hat{\Omega}_{ik}) \hat{F}_{kj}. \quad (23.27.21)$$

The primary reason for taking a corotational approach is to facilitate the maintenance of a vanishing normal stress through the thickness of the shell, something that is achieved by adjusting the corresponding component of the corotated velocity gradient \hat{L}_{33} accordingly. The problem can be stated as to determine \hat{L}_{33} such that when updating the deformation gradient through Equation (23.27.21) and subsequently the stress through Equation (23.27.20), $\hat{\sigma}_{33} = 0$. To this end, it is assumed that

$$\hat{L}_{33} = \alpha (\hat{L}_{11} + \hat{L}_{22}), \quad (23.27.22)$$

for some parameter α that is determined in the following three step procedure. In the first two steps, $\alpha = 0$ and $\alpha = -1$, respectively, resulting in two trial normal stresses $\hat{\sigma}_{33}^{(0)}$ and $\hat{\sigma}_{33}^{(-1)}$. Then it is assumed that the actual normal stress depends linearly on α , meaning that the latter can be determined from

$$0 = \sigma_{33}^{(\alpha)} = \sigma_{33}^{(0)} + \alpha (\sigma_{33}^{(0)} - \sigma_{33}^{(-1)}). \quad (23.27.23)$$

In LS-DYNA, α is given by

$$\alpha = \begin{cases} \frac{\hat{\sigma}_{33}^{(0)}}{\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}} & |\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}| \geq 10^{-4} \\ -1 & \text{otherwise} \end{cases}, \quad (23.27.24)$$

and the stresses are determined from this value of α . Finally, to make sure that the normal stress through the thickness vanishes, it is set to 0 (zero) before exiting the stress update routine.

23.27.2 Derivation of the Continuum Tangent Stiffness

This section will describe the derivation of the continuum tangent stiffness for the Mooney-Rivlin material. For solid elements, the continuum tangent stiffness is chosen in favor of an algorithmic (consistent) tangential modulus as the constitutive equation at

hand is smooth and a consistent tangent modulus is not required for good convergence properties. For shell elements however, this stiffness must ideally be modified in order to account for the zero normal stress condition. This modification, and its consequences, are discussed in the next section.

The continuum tangent modulus in the reference configuration is per definition,

$$E_{ijkl}^{\text{PK}} = \frac{\partial S_{ij}}{\partial E_{kl}} = 2 \frac{\partial S_{ij}}{\partial C_{kl}}. \quad (23.27.25)$$

Splitting up the differentiation of Equation (23.27.17) we get

$$\frac{\partial(I_1 \delta_{ij} - C_{ij})}{\partial C_{kl}} = \delta_{kl} \delta_{ij} - \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad (23.27.26)$$

$$\frac{\partial\left(\frac{1}{I_3^2} C_{ij}^{-1}\right)}{\partial C_{kl}} = -\frac{2}{I_3^2} C_{kl}^{-1} C_{ij}^{-1} - \frac{1}{2I_3^2} (C_{kj}^{-1} C_{il}^{-1} + C_{lj}^{-1} C_{ik}^{-1}) \quad (23.27.27)$$

$$\frac{\partial(I_3(I_3 - 1)C_{ij}^{-1})}{\partial C_{kl}} = I_3(2I_3 - 1)C_{kl}^{-1}C_{ij}^{-1} - \frac{1}{2}I_3(I_3 - 1)(C_{kj}^{-1}C_{il}^{-1} + C_{lj}^{-1}C_{ik}^{-1}). \quad (23.27.28)$$

Since LS-DYNA needs the tangential modulus for the Cauchy stress, it is a good idea to transform the terms in Equation (23.27.27) before summing them up. The push forward operation for the fourth-order tensor E_{ijkl}^{PK} is

$$E_{ijkl}^{\text{TC}} = \frac{1}{J} F_{ia} F_{jb} F_{kc} F_{ld} E_{abcd}^{\text{PK}}. \quad (23.27.29)$$

Since the right Cauchy-Green tensor is $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ and the left Cauchy-Green tensor is $\mathbf{b} = \mathbf{F}^T \mathbf{F}$, and the determinant and trace of the both stretches are equal, the transformation is in practice carried out by interchanging

$$C_{ij}^{-1} \rightarrow \delta_{ij}, \quad (23.27.30)$$

$$\delta_{ij} \rightarrow b_{ij}. \quad (23.27.31)$$

The end result is then

$$\begin{aligned} J E_{ijkl}^{\text{TC}} = & 4B \left[b_{kl} b_{ij} - \frac{1}{2} (b_{ik} b_{jl} + b_{il} b_{jk}) \right] + \frac{4C}{I_3^2} [4\delta_{ij} \delta_{kl} + (\delta_{kj} \delta_{il} + \delta_{lj} \delta_{ik})] + \\ & 8DI_3 \left[(2I_3 - 1) \delta_{ij} \delta_{kl} - \frac{1}{2} (I_3 - 1) (\delta_{kj} \delta_{il} + \delta_{lj} \delta_{ik}) \right]. \end{aligned} \quad (23.27.32)$$

23.27.3 The Algorithmic Tangent Stiffness for Shell Elements

The corotated tangent stiffness matrix is given by Equation (23.27.32) with the exception that the left Cauchy-Green tensor and deformation gradient are given in corotational coordinates, i.e.,

$$J\hat{E}_{ijkl}^{\text{TC}} = 4B \left[\hat{b}_{kl}\hat{b}_{ij} - \frac{1}{2}(\hat{b}_{ik}\hat{b}_{jl} + \hat{b}_{il}\hat{b}_{jk}) \right] + \frac{4C}{I_3^2} [4\delta_{ij}\delta_{kl} + (\delta_{kj}\delta_{il} + \delta_{lj}\delta_{ik})] \\ + 8DI_3 \left[(2I_3 - 1)\delta_{ij}\delta_{kl} - \frac{1}{2}(I_3 - 1)(\delta_{kj}\delta_{il} + \delta_{lj}\delta_{ik}) \right]. \quad (23.27.33)$$

Using this exact expression for the tangent stiffness matrix in the context of shell elements is not adequate since it does not take into account that the normal stress is zero and it must be modified appropriately. To this end, we assume that the tangent moduli in Equation (23.27.33) relates the corotated rate-of-deformation tensor \hat{D}_{ij} to the corotated rate of stress $\hat{\sigma}_{ij}^\bullet$,

$$\hat{\sigma}_{ij}^\bullet = \hat{E}_{ijkl}^{\text{TC}} \hat{D}_{kl}. \quad (23.27.34)$$

Even though this is not completely true, we believe that attempting a more thorough treatment would hardly be worth the effort. The objective can now be stated as to find a modified tangent stiffness matrix $\hat{E}_{ijkl}^{\text{TCalg}}$ such that

$$\hat{\sigma}_{ij}^{\bullet\text{alg}} = \hat{E}_{ijkl}^{\text{TCalg}} \hat{D}_{kl}, \quad (23.27.35)$$

where $\hat{\sigma}_{ij}^{\text{alg}}$ is the stress as it is evaluated in LS-DYNA. The stress update, described in Section 19.27.1, is performed in a rather ad hoc way which probably makes the stated objective unachievable. Still we attempt to extract relevant information from it that enables us to come somewhat close.

An example of a modification of this tangent moduli is due to Hughes and Liu [1981] and given by

$$\hat{E}_{ijkl}^{\text{TCalg}} = \hat{E}_{ijkl}^{\text{TC}} - \frac{\hat{E}_{ij33}^{\text{TC}} \hat{E}_{33kl}^{\text{TC}}}{\hat{E}_{3333}^{\text{TC}}}. \quad (23.27.36)$$

This matrix is derived by eliminating the thickness strain \hat{D}_{33} from the equation $\hat{\sigma}_{33}^\bullet = 0$ in Equation (23.27.35) as an unknown. This modification is unfortunately not consistent with how the stresses are updated in LS-DYNA. When consulting Section 19.27.1, it is suggested that \hat{D}_{33} instead can be eliminated from

$$\hat{D}_{33} = \alpha(\hat{D}_{11} + \hat{D}_{22}), \quad (23.27.37)$$

using the α determined from the stress update. Unfortunately, by the time when the tangent stiffness matrix is calculated, the exact value of α is not known. From experimental observations however, we have found that α is seldom far from being equal to -1 . The fact that $\alpha = -1$ represents incompressibility strengthen this hypothesis. This leads to a modified tangent stiffness $\hat{E}_{ijkl}^{\text{TCalg}}$ that is equal to $\hat{E}_{ijkl}^{\text{TC}}$ except for the following modifications,

$$\begin{aligned}\hat{E}_{ijjj}^{\text{TCalg}} &= \hat{E}_{ijjj}^{\text{TC}} - \hat{E}_{ii33}^{\text{TC}} - \hat{E}_{33jj}^{\text{TC}} + \hat{E}_{3333}^{\text{TC}}, \\ \hat{E}_{33ij}^{\text{TCalg}} &= \hat{E}_{ij33}^{\text{TCalg}} = 0, i \neq j.\end{aligned}\tag{23.27.38}$$

To preclude the obvious singularity, a small positive value is assigned to $\hat{E}_{3333}^{\text{TCalg}}$,

$$\hat{E}_{3333}^{\text{TCalg}} = 10^{-4} (|\hat{E}_{1111}^{\text{TCalg}}| + |\hat{E}_{2222}^{\text{TCalg}}|).\tag{23.27.39}$$

As with the Hughes-Liu modification, this modification preserves symmetry and positive definiteness of the tangent moduli, which together with the stress update “consistency” makes it intuitively attractive.

23.28 Material Model 28: Resultant Plasticity

This plasticity model, based on resultants as illustrated in [Figure 23.28.1](#), is very cost effective but not as accurate as through-thickness integration. This model is available only with the C^0 triangular, Belytschko-Tsay shell, and the Belytschko beam element since these elements, unlike the Hughes-Liu elements, lend themselves very cleanly to a resultant formulation.

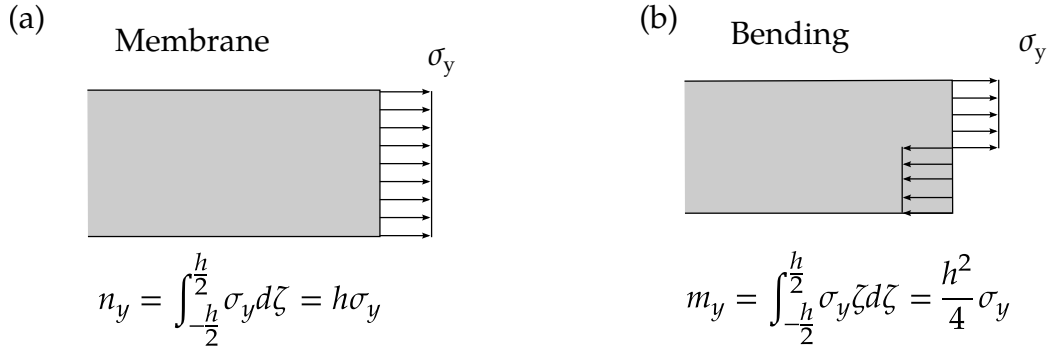


Figure 23.28.1. Full section yield using resultant plasticity.

In applying this model to shell elements the resultants are updated incrementally using the midplane strains ϵ^m and curvatures κ :

$$\Delta n = \Delta t C \epsilon^m \quad (23.28.1)$$

$$\Delta m = \Delta t \frac{h^3}{12} C \kappa, \quad (23.28.2)$$

where the plane stress constitutive matrix is given in terms of Young's Modulus E and Poisson's ratio ν as:

$$\bar{m} = m_{xx}^2 - m_{xx}m_{yy} + m_{yy}^2 + 3m_{xy}^2. \quad (23.28.3)$$

Defining

$$\bar{n} = n_{xx}^2 - n_{xx}n_{yy} + n_{yy}^2 + 3n_{xy}^2, \quad (23.28.4)$$

$$\bar{m} = m_{xx}^2 - m_{xx}m_{yy} + m_{yy}^2 + 3m_{xy}^2, \quad (23.28.5)$$

$$\bar{m}\bar{n} = m_{xx}n_{xx} - \frac{1}{2}m_{xx}n_{yy} - \frac{1}{2}n_{xx}m_{yy} + m_y n_y + 3m_{xy}n_{xy}, \quad (23.28.6)$$

the Ilyushin yield function becomes

$$f(m, n) = \bar{n} + \frac{4|\bar{m}\bar{n}|}{h\sqrt{3}} + \frac{16\bar{m}}{h^2} \leq n_y^2 = h^2\sigma_y^2. \quad (23.28.7)$$

In our implementation we update the resultants elastically and check to see if the yield condition is violated:

$$f(m, n) > n_y^2. \quad (23.28.8)$$

If so, the resultants are scaled by the factor α :

$$\alpha = \sqrt{\frac{n_y^2}{f(m, n)}}. \quad (23.28.9)$$

We update the yield stress incrementally:

$$\sigma_y^{n+1} = \sigma_y^n + E^P \Delta \varepsilon_{\text{plastic}}^{\text{eff}}, \quad (23.28.10)$$

where E^P is the plastic hardening modulus which in incremental plastic strain is approximated by

$$\Delta \varepsilon_{\text{plastic}}^{\text{eff}} = \frac{\sqrt{f(m, n)} - n_y}{h(3G + E^P)}. \quad (23.28.11)$$

Kennedy, et. al., report that this model predicts results that may be too stiff; users of this model should proceed cautiously.

In applying this material model to the Belytschko beam, the flow rule changes to

$$f(m, n) = \hat{f}_x^2 + \frac{4\hat{m}_y^2 A}{3I_{yy}} + \frac{4\hat{m}_z^2 A}{3I_{zz}} \leq n_y^2 = A^2 \sigma_y^2, \quad (23.28.12)$$

have been updated elastically according to Equations (4.16)-(4.18). The yield condition is checked with [Equation \(23.28.8\)](#), and if it is violated, the resultants are scaled as described above.

This model is frequently applied to beams with non-rectangular cross sections. The accuracy of the results obtained should be viewed with some healthy suspicion. No work hardening is available with this model.

23.29 Material Model 29: FORCE LIMITED Resultant Formulation

This material model is available for the Belytschko beam element only. Plastic hinges form at the ends of the beam when the moment reaches the plastic moment. The moment-versus-rotation relationship is specified by the user in the form of a load curve and scale factor. The point pairs of the load curve are (*plastic rotation in radians, plastic moment*). Both quantities should be positive for all points, with the first point pair being (*zero, initial plastic moment*). Within this constraint any form of characteristic may be used including flat or falling curves. Different load curves and scale factors may be specified at each node and about each of the local s and t axes.

Axial collapse occurs when the compressive axial load reaches the collapse load. The collapse load-versus-collapse deflection is specified in the form of a load curve. The points of the load curve are (*true strain, collapse force*). Both quantities should be entered as positive for all points, and will be interpreted as compressive i.e., collapse does not occur in tension. The first point should be the pair (*zero, initial collapse load*).

The collapse load may vary with end moment and with deflection. In this case, several load-deflection curves are defined, each corresponding to a different end moment. Each load curve should have the same number of point pairs and the same deflection values. The end moment is defined as the average of the absolute moments at each end of the beam, and is always positive. It is not possible to make the plastic moment vary with axial load.

A co-rotational technique and moment-curvature relations are used to compute the internal forces. The co-rotational technique is treated in Section 4 in and will not be treated here as we will focus solely on the internal force update and computing the tangent stiffness. For this we use the notation

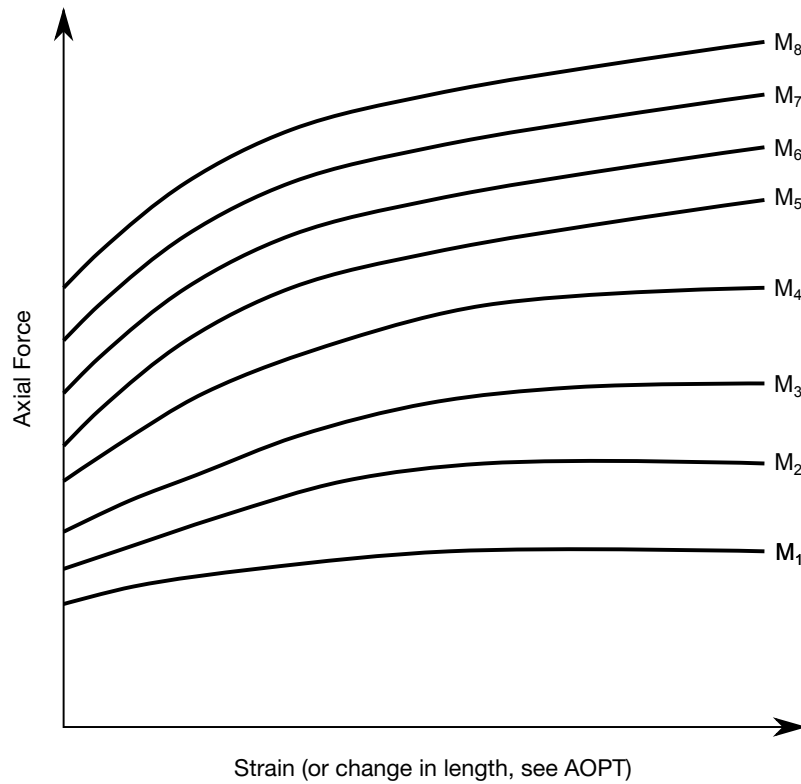


Figure 23.29.1. The force magnitude is limited by the applied end moment. For an intermediate value of the end moment, LS-DYNA interpolates between the curves to determine the allowable force.

$$\begin{aligned}
 E &= \text{Young's modulus} \\
 G &= \text{Shear modulus} \\
 A &= \text{Cross sectional area} \\
 A_s &= \text{Effective area in shear} \\
 l^n &= \text{Reference length of beam} \\
 l^{n+1} &= \text{Current length of beam} \\
 I_{yy} &= \text{Second moment of inertia about } y \\
 I_{zz} &= \text{Second moment of inertia about } z \\
 J &= \text{Polar moment of inertia} \\
 e_i &= i\text{th local base vector in the current configuration} \\
 y_I &= \text{nodal vector in } y \text{ direction at node } I \text{ in the current configuration} \\
 z_I &= \text{nodal vector in } z \text{ direction at node } I \text{ in the current configuration}
 \end{aligned} \tag{23.29.1}$$

We emphasize that the local y and z base vectors in the reference configuration always coincide with the corresponding nodal vectors. The nodal vectors in the current configuration are updated using the Hughes-Winget formula while the base vectors are computed from the current geometry of the element and the current nodal vectors.

23.29.1 Internal Forces

Elastic Update

In the local system for a beam connected by nodes I and J , the axial force is updated as

$$\mathbf{f}_a^{\text{el}} = \mathbf{f}_a^{\text{n}} + K_a^{\text{el}} \delta, \quad (23.29.2)$$

where

$$\begin{aligned} K_a^{\text{el}} &= \frac{EA}{l^n}, \\ \delta &= l^{n+1} - l^n. \end{aligned} \quad (23.29.3)$$

The torsional moment is updated as

$$m_t^{\text{el}} = m_t^{\text{n}} + K_t^{\text{el}} \theta_t, \quad (23.29.4)$$

where

$$\begin{aligned} K_t^{\text{el}} &= \frac{GJ}{l^n}, \\ \theta_t &= \frac{1}{2} \mathbf{e}_1^T (\mathbf{y}_I \times \mathbf{y}_J + \mathbf{z}_I \times \mathbf{z}_J). \end{aligned} \quad (23.29.5)$$

The bending moments are updated as

$$\mathbf{m}_y^{\text{el}} = \mathbf{m}_y^{\text{n}} + \mathbf{A}_y^{\text{el}} \theta_y \quad (23.29.6)$$

$$\mathbf{m}_z^{\text{el}} = \mathbf{m}_z^{\text{n}} + \mathbf{A}_z^{\text{el}} \theta_z, \quad (23.29.7)$$

where

$$\mathbf{A}_*^{\text{el}} = \frac{1}{1 + \varphi_*} \frac{EI_{**}}{l^n} \begin{bmatrix} 4 + \varphi_* & 2 - \varphi_* \\ 2 - \varphi_* & 4 + \varphi_* \end{bmatrix} \quad (23.29.8)$$

$$\varphi_* = \frac{12EI_{**}}{GA_s l^n} \quad (23.29.9)$$

$$\theta_y^T = -\mathbf{e}_3^T (\mathbf{y}_I \times \mathbf{z}_I \quad \mathbf{y}_J \times \mathbf{z}_J) \quad (23.29.10)$$

$$\theta_z^T = \mathbf{e}_2^T (\mathbf{y}_I \times \mathbf{z}_I \quad \mathbf{y}_J \times \mathbf{z}_J). \quad (23.29.11)$$

In the following we refer to \mathbf{A}_*^{el} as the (elastic) moment-rotation matrix.

Plastic Correction

After the elastic update the state of force is checked for yielding as follows. As a preliminary note we emphasize that whenever yielding does not occur the elastic stiffnesses and forces are taken as the new stiffnesses and forces.

The yield moments in direction i at node I as functions of plastic rotations are denoted $m_{il}^Y(\theta_{il}^P)$. This function is given by the user but also depends on whether a plastic hinge has been created. The theory for plastic hinges is given in the LS-DYNA Keyword User's Manual [Hallquist 2003] and is not treated here. Whenever the elastic moment exceeds the plastic moment, the plastic rotations are updated as

$$\theta_{il}^{P(n+1)} = \theta_{il}^{P(n)} + \frac{|m_{il}^{el}| - m_{il}^Y}{\max\left(0.001, A_{i(II)}^{el} + \frac{\partial m_{il}^Y}{\partial \theta_{il}^P}\right)}, \quad (23.29.12)$$

and the moment is reduced to the yield moment

$$m_{il}^{n+1} = m_{il}^Y(\theta_{il}^{P(n+1)}) \operatorname{sgn}(m_{il}^{el}). \quad (23.29.13)$$

The corresponding diagonal component in the moment-rotation matrix is reduced as

$$A_{i(II)}^{n+1} = A_{i(II)}^{el} \left(1 - \alpha \frac{A_{i(II)}^{el}}{\max\left(0.001, A_{i(II)}^{el} + \frac{\partial m_{il}^Y}{\partial \theta_{il}^P}\right)} \right), \quad (23.29.14)$$

where $\alpha \leq 1$ is a parameter chosen such that the moment-rotation matrix remains positive definite.

The yield moment in torsion is given by $m_t^Y(\theta_t^P)$ and is provided by the user. If the elastic torsional moment exceeds this value, the plastic torsional rotation is updated as

$$\theta_t^{P(n+1)} = \theta_t^{P(n)} + \frac{|m_t^{el}| - m_t^Y}{\max\left(0.001, K_t^{el} + \frac{\partial m_t^Y}{\partial \theta_t^P}\right)}, \quad (23.29.15)$$

and the moment is reduced to the yield moment

$$m_t^{n+1} = m_t^Y(\theta_t^{P(n+1)}) \operatorname{sgn}(m_t^{el}). \quad (23.29.16)$$

The torsional stiffness is modified as

$$K_t^{n+1} = K_t^{\text{el}} \left(1 - \alpha \frac{K_t^{\text{el}}}{K_t^{\text{el}} + \frac{\partial m_t^Y}{\partial \theta_t^P}} \right), \quad (23.29.17)$$

where again $\alpha \leq 1$ is chosen so that the stiffness is positive.

Axial collapse is modeled by limiting the axial force by $f_a^Y(\varepsilon, m)$, i.e., a function of the axial strains and the magnitude of bending moments. If the axial elastic force exceeds this value it is reduced to yield

$$f_a^{n+1} = f_a^Y(\varepsilon^{n+1}, m^{n+1}) \text{sgn}(f_a^{\text{el}}), \quad (23.29.18)$$

and the axial stiffness is given by

$$K_a^{n+1} = \max \left(0.05 K_a^{\text{el}}, \frac{\partial f_a^Y}{\partial \varepsilon} \right). \quad (23.29.19)$$

We neglect the influence of change in bending moments when computing this parameter.

Damping

Damping is introduced by adding a viscous term to the internal force on the form

$$\mathbf{f}_v = \mathbf{D} \frac{d}{dt} \begin{bmatrix} \delta \\ \theta_t \\ \theta_y \\ \theta_z \end{bmatrix}, \quad (23.29.20)$$

$$\mathbf{D} = \gamma \begin{bmatrix} K_a^{\text{el}} & & & \\ & K_t^{\text{el}} & & \\ & & A_y^{\text{el}} & \\ & & & A_z^{\text{el}} \end{bmatrix}, \quad (23.29.21)$$

where γ is a damping parameter.

Transformation

The internal force vector in the global system is obtained through the transformation

$$\mathbf{f}_g^{n+1} = \mathbf{S} \mathbf{f}_l^{n+1}, \quad (23.29.22)$$

where

$$\mathbf{S} = \begin{bmatrix} -e_1 & 0 & -e_3/l^{n+1} & -e_3/l^{n+1} & e_2/l^{n+1} & e_2/l^{n+1} \\ 0 & -e_1 & e_2 & 0 & e_3 & 0 \\ e_1 & 0 & e_3/l^{n+1} & e_3/l^{n+1} & -e_2/l^{n+1} & -e_2/l^{n+1} \\ 0 & e_1 & 0 & e_2 & 0 & e_3 \end{bmatrix}, \quad (23.29.23)$$

$$\mathbf{f}_l^{n+1} = \begin{bmatrix} f_a^{n+1} \\ m_t^{n+1} \\ m_y^{n+1} \\ m_z^{n+1} \end{bmatrix}. \quad (23.29.24)$$

23.29.2 Tangent Stiffness

Derivation

The tangent stiffness is derived from taking the variation of the internal force

$$\delta \mathbf{f}_g^{n+1} = \delta \mathbf{S} \mathbf{f}_l^{n+1} + \mathbf{S} \delta \mathbf{f}_l^{n+1}, \quad (23.29.25)$$

which can be written

$$\delta \mathbf{f}_g^{n+1} = \mathbf{K}^{\text{geo}} \delta \mathbf{u} + \mathbf{K}^{\text{mat}} \delta \mathbf{u}, \quad (23.29.26)$$

where

$$\delta \mathbf{u} = [\delta x_I^T \quad \delta \omega_I^T \quad \delta x_J^T \quad \delta \omega_J^T]^T. \quad (23.29.27)$$

There are two contributions to the tangent stiffness, one geometrical and one material contribution. The geometrical contribution is given (approximately) by

$$\mathbf{K}^{\text{geo}} = \mathbf{R}(\mathbf{f}_l^{n+1} \otimes \mathbf{I})\mathbf{W} - \frac{1}{l^{n+1}l^{n+1}} \mathbf{T} \mathbf{f}_l^{n+1} \mathbf{L}, \quad (23.29.28)$$

where

$$\mathbf{R} = \begin{bmatrix} R_1 & 0 & R_3/l^{n+1} & R_3/l^{n+1} & -R_2/l^{n+1} & -R_2/l^{n+1} \\ 0 & R_1 & -R_2 & 0 & -R_3 & 0 \\ -R_1 & 0 & -R_3/l^{n+1} & -R_3/l^{n+1} & R_2/l^{n+1} & R_2/l^{n+1} \\ 0 & -R_1 & 0 & -R_2 & 0 & -R_3 \end{bmatrix}, \quad (23.29.29)$$

$$\mathbf{W} = [-R_1/l^{n+1} \quad \mathbf{e}_1 \mathbf{e}_1^T/2 \quad R_1/l^{n+1} \quad \mathbf{e}_1 \mathbf{e}_1^T/2], \quad (23.29.30)$$

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & -e_3 & -e_3 & e_2 & e_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_3 & e_3 & -e_2 & -e_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (23.29.31)$$

$$\mathbf{L} = [-\mathbf{e}_1^T \quad 0 \quad \mathbf{e}_1^T \quad 0], \quad (23.29.32)$$

and \mathbf{I} is the 3 by 3 identity matrix. We use \otimes as the outer matrix product and define

$$\mathbf{R}_i \mathbf{v} = \mathbf{e}_i \times \mathbf{v}. \quad (23.29.33)$$

The material contribution can be written as

$$\mathbf{K}^{\text{mat}} = \mathbf{S} \mathbf{K} \mathbf{S}^T, \quad (23.29.34)$$

where

$$\mathbf{K} = \begin{bmatrix} K_a^{n+1} & & & \\ & K_t^{n+1} & & \\ & & A_y^{n+1} & \\ & & & A_z^{n+1} \end{bmatrix} + \frac{1}{\Delta t} \mathbf{D}. \quad (23.29.35)$$

23.30 Material Model 30: Closed-Form Update Shell Plasticity

This section presents the mathematical details of the shape memory alloy material in LS-DYNA. The description closely follows the one of Auricchio and Taylor [1997] with appropriate modifications for this particular implementation.

23.30.1 Mathematical Description of the Material Model

The Kirchhoff stress $\boldsymbol{\tau}$ in the shape memory alloy can be written

$$\boldsymbol{\tau} = p\mathbf{i} + \mathbf{t}, \quad (23.30.1)$$

where \mathbf{i} is the second order identity tensor and

$$p = K(\theta - 3\alpha\zeta_S\epsilon_L), \quad (23.30.2)$$

$$\mathbf{t} = 2G(\mathbf{e} - \zeta_S\epsilon_L\mathbf{n}). \quad (23.30.3)$$

Here K and G are bulk and shear modulii, θ and \mathbf{e} are volumetric and shear logarithmic strains and α and ϵ_L are constant material parameters. There is an option to define the bulk and shear modulii as functions of the martensite fraction according to

$$\begin{aligned} K &= K_A + \zeta_S(K_S - K_A), \\ G &= G_A + \zeta_S(G_S - G_A), \end{aligned} \quad (23.30.4)$$

in case the stiffness of the martensite differs from that of the austenite. Furthermore, the unit vector \mathbf{n} is defined as

$$\mathbf{n} = \mathbf{e}/(\|\mathbf{e}\| + 10^{-12}), \quad (23.30.5)$$

and a loading function is introduced as

$$F = 2G\|\mathbf{e}\| + 3\alpha K\theta - \beta\zeta_S, \quad (23.30.6)$$

where

$$\beta = (2G + 9\alpha^2 K)\epsilon_L. \quad (23.30.7)$$

For the evolution of the martensite fraction ζ_S in the material, the following rule is adopted

$$\left. \begin{aligned} F - R_s^{AS} &> 0 \\ \dot{F} &> 0 \\ \zeta_S &< 1 \end{aligned} \right\} \Rightarrow \dot{\zeta}_S = -(1 - \zeta_S) \frac{\dot{F}}{F - R_f^{AS}} \quad (23.30.8)$$

$$\left. \begin{array}{l} F - R_s^{SA} < 0 \\ \dot{F} < 0 \\ \tilde{\zeta}_S > 0 \end{array} \right\} \Rightarrow \dot{\tilde{\zeta}}_S = \tilde{\zeta}_S \frac{\dot{F}}{F - R_f^{SA}}. \quad (23.30.9)$$

Here R_s^{AS} , R_f^{AS} , R_s^{SA} and R_f^{SA} are constant material parameters. The Cauchy stress is finally obtained as

$$\sigma = \frac{\tau}{J}, \quad (23.30.10)$$

where J is the Jacobian of the deformation.

23.30.2 Algorithmic Stress Update

For the stress update we assume that the martensite fraction $\tilde{\zeta}_S^n$ and the value of the loading function F^n is known from time t_n and the deformation gradient at time t_{n+1} , F , is known. We form the left Cauchy-Green tensor as $\mathbf{B} = \mathbf{F}\mathbf{F}^T$ which is diagonalized to obtain the principal values and directions $\mathbf{\Lambda} = \text{diag}(\lambda_i)$ and \mathbf{Q} . The volumetric and principal shear logarithmic strains are given by

$$\begin{aligned} \theta &= \log(J), \\ e_i &= \log\left(\frac{\lambda_i}{J^{1/3}}\right), \end{aligned} \quad (23.30.11)$$

where

$$J = \lambda_1 \lambda_2 \lambda_3. \quad (23.30.12)$$

is the total Jacobian of the deformation. Using Equation (23.30.6) with $\tilde{\zeta}_S = \tilde{\zeta}_S^n$, a value F^{trial} of the loading function can be computed. The discrete counterpart of Equation (23.30.8) becomes

$$\left. \begin{array}{l} F^{\text{trial}} - R_s^{AS} > 0 \\ F^{\text{trial}} - F^n > 0 \\ \tilde{\zeta}_S^n < 1 \end{array} \right\} \Rightarrow \Delta \tilde{\zeta}_S = - (1 - \tilde{\zeta}_S^n - \Delta \tilde{\zeta}_S) \frac{F^{\text{trial}} - \beta \Delta \tilde{\zeta}_S - \min(\max(F^n, R_s^{AS}), R_f^{AS})}{F^{\text{trial}} - \beta \Delta \tilde{\zeta}_S - R_f^{AS}} \quad (23.30.13)$$

$$\left. \begin{array}{l} F^{\text{trial}} - R_s^{SA} < 0 \\ F^{\text{trial}} - F^n < 0 \\ \tilde{\zeta}_S^n > 0 \end{array} \right\} \Rightarrow \Delta \tilde{\zeta}_S = (\tilde{\zeta}_S^n + \Delta \tilde{\zeta}_S) \frac{F^{\text{trial}} - \beta \Delta \tilde{\zeta}_S - \min(\max(F^n, R_f^{SA}), R_s^{SA})}{F^{\text{trial}} - \beta \Delta \tilde{\zeta}_S - R_f^{SA}}. \quad (23.30.14)$$

If none of the two conditions to the left are satisfied, set $\tilde{\zeta}_S^{n+1} = \tilde{\zeta}_S^n$, $F^{n+1} = F^{\text{trial}}$ and compute the stress σ^{n+1} using Equations (23.30.1), (23.30.2), (23.30.5), (23.30.10) and $\tilde{\zeta}_S = \tilde{\zeta}_S^n$. When phase transformation occurs according to a condition to the left, the

corresponding equation to the right is solved for $\Delta\tilde{\zeta}_S$. If the bulk and shear moduli are constant this is an easy task. Otherwise F^{trial} as well as β depends on this parameter and makes things a bit more tricky. We have that

$$\begin{aligned} F^{\text{trial}} &= F_n^{\text{trial}} \left(1 + \frac{E_S - E_A}{E_n} \Delta\tilde{\zeta}_S \right), \\ \beta &= \beta_n \left(1 + \frac{E_S - E_A}{E_n} \Delta\tilde{\zeta}_S \right), \end{aligned} \quad (23.30.15)$$

where E_S and E_A are Young's moduli for martensite and austenite, respectively. The subscript n is introduced for constant quantities evaluated at time t_n . To simplify the upcoming expressions, these relations are written

$$\begin{aligned} F^{\text{trial}} &= F_n^{\text{trial}} + \Delta F^{\text{trial}} \Delta\tilde{\zeta}_S, \\ \beta &= \beta_n + \Delta\beta \Delta\tilde{\zeta}_S. \end{aligned} \quad (23.30.16)$$

Inserting these expressions into Equation (19.30.7) results in

$$\begin{aligned} f(\Delta\tilde{\zeta}_S) &= \Delta\beta(1 - \tilde{\zeta}_S^n) \Delta\tilde{\zeta}_S^2 + \left(R_f^{\text{AS}} - \tilde{F}_{\text{AS}}^n + (\beta_n - \Delta F^{\text{trial}})(1 - \tilde{\zeta}_S^n) \right) \Delta\tilde{\zeta}_S + \\ &(1 - \tilde{\zeta}_S^n)(\tilde{F}_{\text{AS}}^n - F_n^{\text{trial}}) = 0. \end{aligned} \quad (23.30.17)$$

and

$$\begin{aligned} f(\Delta\tilde{\zeta}_S) &= \Delta\beta\tilde{\zeta}_S^n \Delta\tilde{\zeta}_S^2 + (\tilde{F}_{\text{SA}}^n - R_f^{\text{SA}} + (\beta_n - \Delta F^{\text{trial}})\tilde{\zeta}_S^n) \Delta\tilde{\zeta}_S + \\ &\tilde{\zeta}_S^n(\tilde{F}_{\text{SA}}^n - F_n^{\text{trial}}) = 0. \end{aligned} \quad (23.30.18)$$

respectively, where we have for simplicity set

$$\begin{aligned} \tilde{F}_{\text{AS}}^n &= \min(\max(F^n, R_s^{\text{AS}}), R_f^{\text{AS}}), \\ \tilde{F}_{\text{SA}}^n &= \min(\max(F^n, R_f^{\text{SA}}), R_s^{\text{SA}}). \end{aligned} \quad (23.30.19)$$

The solutions to these equations are approximated with two Newton iterations starting in the point $\Delta\tilde{\zeta}_S = 0$. Now set $\tilde{\zeta}_S^{n+1} = \min(1, \max(0, \tilde{\zeta}_S^n + \Delta\tilde{\zeta}_S))$ and compute σ^{n+1} and F^{n+1} according to [Equations \(23.30.1\), \(23.30.2\), \(23.30.5\), \(23.30.6\), \(23.30.10\)](#) and $\tilde{\zeta}_S = \tilde{\zeta}_S^{n+1}$.

23.30.3 Tangent Stiffness Matrix

An algorithmic tangent stiffness matrix relating a change in true strain to a corresponding change in Kirchhoff stress is derived in the following. Taking the variation of [Equation \(23.30.2\)](#) results in

$$\delta p = K(\delta\theta - 3\alpha\delta\tilde{\zeta}_S\varepsilon_L) + \delta K(\theta - 3\alpha\tilde{\zeta}_S\varepsilon_L), \quad (23.30.20)$$

$$\delta \mathbf{t} = 2G(\delta \mathbf{e} - \delta\tilde{\zeta}_S\varepsilon_L \mathbf{n} - \tilde{\zeta}_S\varepsilon_L \delta \mathbf{n}) + 2\delta G(\mathbf{e} - \tilde{\zeta}_S\varepsilon_L \mathbf{n}). \quad (23.30.21)$$

The variation of the unit vector in Equation (23.30.5) can be written

$$\delta \mathbf{n} = \frac{1}{\|\mathbf{e}\| + 10^{-12}} (\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) \delta \mathbf{e}, \quad (23.30.22)$$

where \mathbf{I} is the fourth order identity tensor. For the variation of martensite fraction we introduce the indicator parameters H^{AS} and H^{SA} that should give information of the probability of phase transformation occurring in the next stress update. Set initially $H^{\text{AS}} = H^{\text{SA}} = 0$ and change them according to

$$\left. \begin{array}{l} F^{\text{trial}} - R_s^{\text{AS}} > 0 \\ F^{\text{trial}} - F^n > 0 \\ \zeta_S^n + \Delta \zeta_S \leq 1 \end{array} \right\} \Rightarrow H^{\text{AS}} = 1, \quad (23.30.23)$$

$$\left. \begin{array}{l} F^{\text{trial}} - R_s^{\text{SA}} < 0 \\ F^{\text{trial}} - F^n < 0 \\ \zeta_S^n + \Delta \zeta_S \geq 0 \end{array} \right\} \Rightarrow H^{\text{SA}} = 1, \quad (23.30.24)$$

using the quantities computed in the previous stress update. For the variation of the martensite fraction we take variations of Equations (23.30.17) and (23.30.18) with

$$\delta F_n^{\text{trial}} = 2G\mathbf{n} : \delta \mathbf{e} + 3\alpha K \delta \theta, \quad (23.30.25)$$

which results in

$$\delta \zeta_S = \gamma (2G\mathbf{n} : \delta \mathbf{e} + 3\alpha K \delta \theta), \quad (23.30.26)$$

where

$$\gamma = \frac{(1 - \zeta_S^n) H^{\text{AS}}}{R_f^{\text{AS}} - \tilde{F}_{\text{AS}}^n + (\beta_n - \Delta F_n^{\text{trial}})(1 - \zeta_S^n)} + \frac{\zeta_S^n H^{\text{SA}}}{\tilde{F}_{\text{SA}}^n - R_f^{\text{SA}} + (\beta_n - \Delta F_n^{\text{trial}})\zeta_S^n}. \quad (23.30.27)$$

As can be seen, we use the value of γ obtained in the previous stress update since this is easier to implement and will probably give a good indication of the current value of this parameter.

The variation of the material parameters K and G results in

$$\begin{aligned} \delta K &= (K_S - K_A) \delta \zeta_S, \\ \delta G &= (G_S - G_A) \delta \zeta_S, \end{aligned} \quad (23.30.28)$$

and, finally, using the identities

$$\mathbf{n} : \delta \mathbf{e} = \mathbf{n} : \delta \boldsymbol{\varepsilon}, \quad (23.30.29)$$

$$\delta \theta = \mathbf{i} : \delta \boldsymbol{\varepsilon}, \quad (23.30.30)$$

$$\delta \boldsymbol{\tau} = \mathbf{i} \delta p + \delta \mathbf{t}, \quad (23.30.31)$$

results in

$$\begin{aligned}
\delta \boldsymbol{\tau} = & \left\{ 2G \left(1 - \frac{\tilde{\zeta}_S \varepsilon_L}{\|\mathbf{e}\| + 10^{-12}} \right) \mathbf{I}^{\text{dev}} \right. \\
& + K \left[1 - 9\alpha^2 K \gamma \varepsilon_L + 3\alpha \gamma (K_S - K_A) (\theta - 3\alpha \tilde{\zeta}_S \varepsilon_L) \right] \mathbf{i} \otimes \mathbf{i} \\
& + 2\gamma G (K_S - K_A) (\theta - 3\alpha \tilde{\zeta}_S \varepsilon_L) \mathbf{i} \otimes \mathbf{n} \\
& + 6\gamma \alpha K (G_S - G_A) (\|\mathbf{e}\| - \tilde{\zeta}_S \varepsilon_L) \mathbf{n} \otimes \mathbf{i} \\
& + 2G \left[\frac{\tilde{\zeta}_S \varepsilon_L}{\|\mathbf{e}\| + 10^{-12}} - 2G \gamma \varepsilon_L + 2\gamma (G_S - G_A) (\|\mathbf{e}\| - \tilde{\zeta}_S \varepsilon_L) \right] \mathbf{n} \otimes \mathbf{n} \\
& \left. - 6KG\alpha \gamma \varepsilon_L (\mathbf{i} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{i}) \right\} \delta \varepsilon.
\end{aligned} \tag{23.30.32}$$

where \mathbf{I}^{dev} is the fourth order deviatoric identity tensor. In general this tangent is not symmetric because of the terms on the second line in the expression above. We simply use a symmetrization of the tangent stiffness above in the implementation. Furthermore, we transform the tangent to a tangent closer related to the one that should be used in the LS-DYNA implementation,

$$\begin{aligned}
\mathbf{C} = J^{-1} & \left\{ 2G \left(1 - \frac{\tilde{\zeta}_S \varepsilon_L}{\|\mathbf{e}\| + 10^{-12}} \right) \mathbf{I}^{\text{dev}} \right. \\
& + K \left[1 - 9\alpha^2 K \gamma \varepsilon_L + 3\alpha \gamma (K_S - K_A) (\theta - 3\alpha \tilde{\zeta}_S \varepsilon_L) \right] \mathbf{i} \otimes \mathbf{i} \\
& + 3\gamma \alpha K (G_S - G_A) (\|\mathbf{e}\| - \tilde{\zeta}_S \varepsilon_L) - 6KG\alpha \gamma \varepsilon_L \mathbf{i} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{i} \\
& \left. + 2G \left[\frac{\tilde{\zeta}_S \varepsilon_L}{\|\mathbf{e}\| + 10^{-12}} - 2G \gamma \varepsilon_L + 2\gamma (G_S - G_A) (\|\mathbf{e}\| - \tilde{\zeta}_S \varepsilon_L) \right] \mathbf{n} \otimes \mathbf{n} \right\} \delta \varepsilon.
\end{aligned} \tag{23.30.33}$$

23.31 Material Model 31: Slightly Compressible Rubber Model

This model implements a modified form of the hyperelastic constitutive law first described in [Kenchington 1988].

The strain energy functional, U , is defined in terms of the input constants as:

$$U = C_{100}I_1 + C_{200}I_1^2 + C_{300}I_1^3 + C_{400}I_1^4 + C_{110}I_1I_2 + C_{210}I_1^2I_2 + C_{010}I_2 + C_{020}I_2^2 + f(J), \quad (23.31.1)$$

where the strain invariants can be expressed in terms of the deformation gradient matrix, F_{ij} , and the Green-St. Venant strain tensor, E_{ij} :

$$\begin{aligned} J &= |F_{ij}| \\ I_1 &= E_{ii} \\ I_2 &= \frac{1}{2!} \delta_{pq}^{ij} E_{pi} E_{qj}. \end{aligned} \quad (23.31.2)$$

The derivative of U with respect to a component of strain gives the corresponding component of stress

$$S_{ij} = \frac{\partial U}{\partial E_{ij}}, \quad (23.31.3)$$

where, S_{ij} , is the second Piola-Kirchhoff stress tensor which is transformed into the Cauchy stress tensor:

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl}, \quad (23.31.4)$$

where ρ_0 and ρ are the initial and current density, respectively.

23.32 Material Model 32: Laminated Glass Model

This model is available for modeling safety glass. Safety glass is a layered material of glass bonded to a polymer material which can undergo large strains.

The glass layers are modeled by isotropic hardening plasticity with failure based on exceeding a specified level of plastic strain. Glass is quite brittle and cannot withstand large strains before failing. Plastic strain was chosen for failure since it increases monotonically and, therefore, is insensitive to spurious numerical noise in the solution.

The material to which the glass is bonded is assumed to stretch plastically without failure. The user defined integration rule option must be used with this material. The user defined rule specifies the thickness of the layers making up the safety glass. Each integration point is flagged with a zero if the layer is glass and with a one if the layer is polymer.

An iterative plane stress plasticity algorithm is used to enforce the plane stress condition.

23.33 Material Model 33: Barlat's Anisotropic Plasticity Model

This model was developed by Barlat, Lege, and Brem [1991] for modeling material behavior in forming processes. The finite element implementation of this model is described in detail by Chung and Shah [1992] and is used here.

The yield function Φ is defined as

$$\Phi = |S_1 - S_2|^m + |S_2 - S_3|^m + |S_3 - S_1|^m = 2^m, \quad (23.33.1)$$

where $\bar{\sigma}$ is the effective stress, and S_i for $i = 1, 2, 3$ are the principal values of the symmetric matrix $S_{\alpha\beta}$,

$$\begin{aligned} S_{xx} &= \frac{[c(\sigma_{xx} - \sigma_{yy}) - b(\sigma_{zz} - \sigma_{xx})]}{3} \\ S_{yy} &= \frac{[a(\sigma_{yy} - \sigma_{zz}) - c(\sigma_{xx} - \sigma_{yy})]}{3} \\ S_{zz} &= \frac{[b(\sigma_{zz} - \sigma_{xx}) - a(\sigma_{yy} - \sigma_{zz})]}{3} \\ S_{yz} &= f\sigma_{yz} \\ S_{zx} &= g\sigma_{zx} \\ S_{xy} &= h\sigma_{xy}. \end{aligned} \quad (23.33.2)$$

The material constants a, b, c, f, g and h represent anisotropic properties. When $a = b = c = f = g = h = 1$, the material is isotropic and the yield surface reduces to the Tresca yield surface for $m = 1$ and von Mises yield surface for $m = 2$ or 4 . For face-centered-cubic (FCC) materials $m = 8$ is recommended and for body-centered-cubic (BCC) materials $m = 6$ is used.

The yield strength of the material is

$$\sigma_y = k(1 + \varepsilon_0)^n. \quad (23.33.3)$$

23.34 Material Model 34: Fabric

The fabric model is a variation on the Layered Orthotropic Composite material model (Material 22) and is valid for only 3 and 4 node membrane elements. This material model is strongly recommended for modeling airbags and seatbelts. In addition to being a constitutive model, this model also invokes a special membrane element formulation that is better suited to the large deformations experienced by fabrics. For thin fabrics, buckling (wrinkling) can occur with the associated inability of the structure to support compressive stresses; a material parameter flag is included for this option. A linear elastic liner is also included which can be used to reduce the tendency for these material/elements to be crushed when the no-compression option is invoked.

If the airbag material is to be approximated as an isotropic elastic material, then only one Young's modulus and Poisson's ratio should be defined. The elastic approximation is very efficient because the local transformations to the material coordinate system may be skipped. If orthotropic constants are defined, it is very important to consider the orientation of the local material system and use great care in setting up the finite element mesh.

If the reference configuration of the airbag is taken as the folded configuration, the geometrical accuracy of the deployed bag will be affected by both the stretching and the compression of elements during the folding process. Such element distortions are very difficult to avoid in a folded bag. By reading in a reference configuration such as the final unstretched configuration of a deployed bag, any distortions in the initial geometry of the folded bag will have no effect on the final geometry of the inflated bag. This is because the stresses depend only on the deformation gradient matrix:

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \quad (23.34.1)$$

where the choice of X_j may coincide with the folded or unfold configurations. It is this unfolded configuration which may be specified here. When the reference geometry is used then the no-compression option should be active. With the reference geometry it is possible to shrink the airbag and then perform the inflation. Although the elements in the shrunken bag are very small, the time step can be based on the reference geometry so a very reasonable time step size is obtained. The reference geometry based time step size is optional in the input.

The parameters fabric leakage coefficient, FLC, fabric area coefficient, FAC, and effective leakage area, ELA, for the fabric in contact with the structure are optional for the Wang-Nefske and hybrid inflation models. It is possible for the airbag to be constructed of multiple fabrics having different values of porosity and permeability. The gas leakage through the airbag fabric then requires an accurate determination of the areas by part ID

available for leakage. The leakage area may change over time due to stretching of the airbag fabric or blockage when the outer surface of the bag is in contact with the structure. LS-DYNA can check the interaction of the bag with the structure and split the areas into regions that are blocked and unblocked depending on whether the regions are in contact or not, respectively. Typically, the parameters, FLC and FAC, must be determined experimentally and their variation with time and pressure are optional inputs that allow for maximum modeling flexibility.

23.35 Material Model 35: Kinematic/Isotropic Plastic Green-Naghdi Rate

The reader interested in an detailed discussion of the relative merits of various stress rates, especially Jaumann [1911] and Green-Naghdi [1965], is urged to read the work of Johnson and Bammann [1984]. A mathematical description of these two stress rates, and how they are implemented in LS-DYNA, is given in the section entitled *Stress Update Overview* in this manual.

In the cited work by Johnson and Bammann, they conclude that the Green-Naghdi stress rate is to be preferred over all other proposed stress rates, including the most widely used Jaumann rate, because the Green-Naghdi stress rate is based on the notions of invariance under superimposed rigid-body motions. However, implementation of the Green-Naghdi stress rate comes at a significant computational cost compared to the Jaumann stress rate, e.g., see the discussion in this manual in the section entitled *Green-Naghdi Stress Rate*.

Perhaps more importantly, in practical applications there is little if any noted difference in the results generated using either Jaumann or Green-Naghdi stress rate. This is in part due to the fact that the Jaumann stress rate only produces erroneous results³ when linear kinematic hardening is used; the results for isotropic hardening are not affected by the choice of either of these two stress rates. Also in practical applications, the shear strains are rather small, compared to the extensional strains, and if they are not small it is usually the case that the material description, i.e., constitutive model, is not valid for such large shear strains.

³ The results of a simple shear simulation, monotonically increasing shear deformation, produce sinusoidal stress response.

23.36 Material Model 36: Barlat's 3-Parameter Plasticity Model

Material model 36 in LS-DYNA aims at modeling sheets with anisotropy under plane stress conditions. It allows the use of the Lankford parameters for the definition of the anisotropic yield surface. The yield condition can be written

$$f(\sigma, \varepsilon_p) = \sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) - \sigma_Y(\varepsilon_p) \leq 0, \quad (23.36.1)$$

where

$$\begin{aligned} \sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) &= \left(\frac{a}{2} |K_1 + K_2|^m + \frac{a}{2} |K_1 - K_2|^m + \frac{c}{2} |2K_2|^m \right)^{1/m} \\ K_1 &= K_1(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \frac{\sigma_{11} + h\sigma_{22}}{2} \\ K_2 &= K_2(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \sqrt{\left(\frac{\sigma_{11} - h\sigma_{22}}{2} \right)^2 + p^2 \sigma_{12}^2}, \end{aligned} \quad (23.36.2)$$

and the hardening of the yield surface is either linear, exponential or determined by a load curve. In the above, the stress components σ_{11} , σ_{22} and σ_{12} are with respect to the material coordinate system and ε_p denotes the effective plastic strain. The material parameters a , c , h and p can be determined from the Lankford parameters as described in the LS-DYNA Keyword User's Manual [Hallquist 2003]. The Lankford parameters, or R-values, are defined as the ratio of instantaneous width change to instantaneous thickness change. That is, assume that the width W and thickness T are measured as function of strain. Then the corresponding R-value is given by

$$R = \frac{\frac{dW}{d\varepsilon}/W}{\frac{dT}{d\varepsilon}/T}.$$

The gradient of the yield surface is denoted

$$\frac{\partial f}{\partial \sigma}(\sigma) = \begin{pmatrix} \frac{\partial f}{\partial \sigma_{11}}(\sigma) \\ \frac{\partial f}{\partial \sigma_{22}}(\sigma) \\ \frac{\partial f}{\partial \sigma_{12}}(\sigma) \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial \sigma_{11}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) \\ \frac{\partial f}{\partial \sigma_{22}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) \\ \frac{\partial f}{\partial \sigma_{12}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) \\ 0 \\ 0 \end{pmatrix}, \quad (23.36.3)$$

where

$$\begin{aligned} \frac{\partial f}{\partial \sigma_{11}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) &= \frac{\partial \sigma_{\text{eff}}}{\partial \sigma_{11}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \frac{\sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12})^{1-m}}{2} \cdot \\ &\left\{ a(K_1 - K_2)|K_1 - K_2|^{m-2} \left(\frac{1}{2} - \frac{\sigma_{11} - h\sigma_{22}}{4K_2} \right) + \right. \\ &a(K_1 + K_2)|K_1 + K_2|^{m-2} \left(\frac{1}{2} + \frac{\sigma_{11} - h\sigma_{22}}{4K_2} \right) + \\ &\left. c2^m K_2^{m-1} \frac{\sigma_{11} - h\sigma_{22}}{4K_2} \right\}, \end{aligned} \quad (23.36.4)$$

$$\begin{aligned} \frac{\partial f}{\partial \sigma_{22}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) &= \frac{\partial \sigma_{\text{eff}}}{\partial \sigma_{22}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \frac{\sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12})^{1-m}}{2} h \cdot \\ &\left\{ a(K_1 - K_2)|K_1 - K_2|^{m-2} \left(\frac{1}{2} + \frac{\sigma_{11} - h\sigma_{22}}{4K_2} \right) + \right. \\ &a(K_1 + K_2)|K_1 + K_2|^{m-2} \left(\frac{1}{2} - \frac{\sigma_{11} - h\sigma_{22}}{4K_2} \right) - \\ &\left. c2^m K_2^{m-1} \frac{\sigma_{11} - h\sigma_{22}}{4K_2} \right\}, \end{aligned} \quad (23.36.5)$$

and

$$\begin{aligned} \frac{\partial f}{\partial \sigma_{12}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) &= \frac{\partial \sigma_{\text{eff}}}{\partial \sigma_{12}}(\sigma_{11}, \sigma_{22}, \sigma_{12}) = \frac{\sigma_{\text{eff}}(\sigma_{11}, \sigma_{22}, \sigma_{12})^{1-m}}{2} \frac{p^2 \sigma_{12}}{K_2} \cdot \\ &\left\{ -a(K_1 - K_2)|K_1 - K_2|^{m-2} + a(K_1 + K_2)|K_1 + K_2|^{m-2} + c2^m K_2^{m-1} \right\}. \end{aligned} \quad (23.36.6)$$

23.36.1 Material Tangent Stiffness

Since the plastic model is associative, the general expression for tangent relating the total strain rate to total stress rate can be found in standard textbooks. Since this situation is rather special we derive it here for the plane stress model presented in the previous section. The elastic stress-strain relation can be written

$$\begin{aligned} \dot{\sigma} &= \begin{pmatrix} \dot{\sigma}_{11} \\ \dot{\sigma}_{22} \\ \dot{\sigma}_{12} \\ \dot{\sigma}_{23} \\ \dot{\sigma}_{13} \end{pmatrix} = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & & & \\ \nu & 1 & & & \\ & & \frac{1-\nu}{2} & & \\ & & & \frac{1-\nu}{2} & \\ & & & & \frac{1-\nu}{2} \end{pmatrix} \begin{pmatrix} \dot{\epsilon}_{11} - \dot{\epsilon}_{11}^P \\ \dot{\epsilon}_{22} - \dot{\epsilon}_{22}^P \\ 2(\dot{\epsilon}_{12} - \dot{\epsilon}_{12}^P) \\ 2(\dot{\epsilon}_{23} - \dot{\epsilon}_{23}^P) \\ 2(\dot{\epsilon}_{13} - \dot{\epsilon}_{13}^P) \end{pmatrix} \\ &= \mathbf{C}^{\text{PS}}(\dot{\epsilon} - \dot{\epsilon}^P). \end{aligned} \quad (23.36.7)$$

where E is the Young's modulus, ν is the Poisson's ratio and \mathbf{C}^{PS} denotes the plane stress elastic tangential stiffness matrix. The associative flow rule for the plastic strain can be written

$$\dot{\epsilon}^P = \dot{\lambda} \frac{\partial f}{\partial \sigma'} \quad (23.36.8)$$

and the consistency condition results in

$$\frac{\partial f^T}{\partial \sigma} \dot{\sigma} + \frac{\partial f}{\partial \epsilon_p} \dot{\epsilon}_p = 0. \quad (23.36.9)$$

For algorithmic consistency, the effective plastic strain rate is defined as $\dot{\epsilon}_p = \dot{\lambda}$. Multiplying Equation (23.36.7) with $\frac{\partial f}{\partial \sigma}$ and using Equation (23.36.8) and Equation (23.36.9) gives

$$\dot{\lambda} = \frac{\frac{\partial f^T}{\partial \sigma} \mathbf{C}^{ps} \dot{\epsilon}}{\frac{\partial f^T}{\partial \sigma} \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} - \frac{\partial f}{\partial \epsilon_p}}. \quad (23.36.10)$$

Inserting

$$\dot{\epsilon}^P = \frac{\frac{\partial f^T}{\partial \sigma} \mathbf{C}^{ps} \dot{\epsilon}}{\frac{\partial f^T}{\partial \sigma} \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} - \frac{\partial f}{\partial \epsilon_p}} \frac{\partial f}{\partial \sigma'} \quad (23.36.11)$$

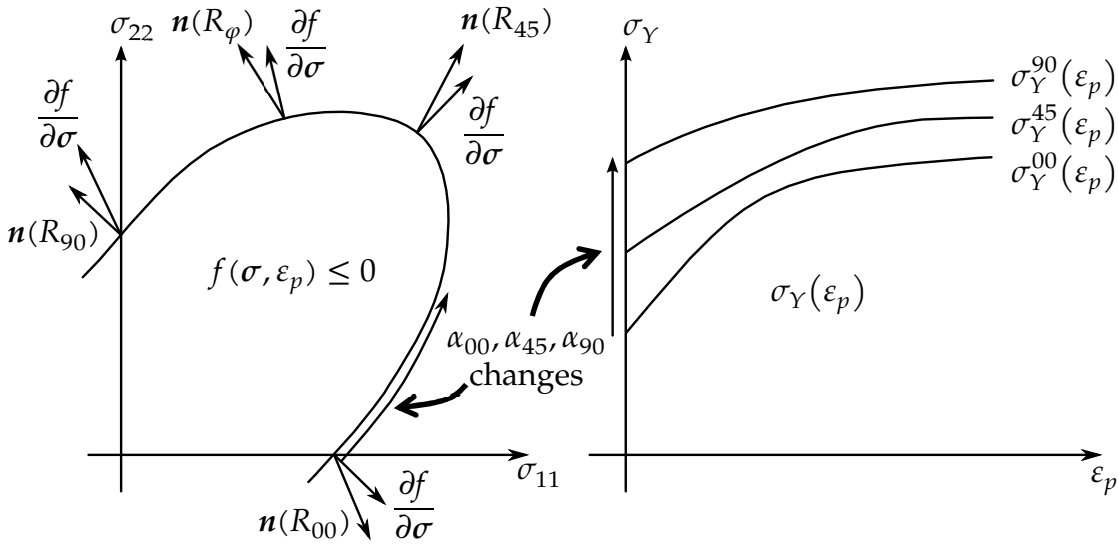
into Equation (23.36.7) results in

$$\dot{\sigma} = \left(\mathbf{C}^{ps} - \frac{\left\{ \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} \right\} \left\{ \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} \right\}^T}{\frac{\partial f^T}{\partial \sigma} \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} - \frac{\partial f}{\partial \epsilon_p}} \right) \dot{\epsilon}. \quad (23.36.12)$$

To get the elastic-plastic tangent stiffness tensor in the element coordinate system it needs to be transformed back. Since the elastic tangential stiffness tensor is isotropic with respect to the axis of rotation, the plastic tangent stiffness tensor can be written

$$\mathbf{C}_{\text{plastic}}^{ps} = \left(\mathbf{C}^{ps} - \frac{\left\{ \mathbf{Q} \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} \right\} \left\{ \mathbf{Q} \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} \right\}^T}{\frac{\partial f^T}{\partial \sigma} \mathbf{C}^{ps} \frac{\partial f}{\partial \sigma} - \frac{\partial f}{\partial \epsilon_p}} \right), \quad (23.36.13)$$

where \mathbf{Q} is the rotation matrix in Voigt form.



23-1 Plastic flow direction (left) and hardening (right) illustrated for variable R-values and hardening. Changes in α_{00} , α_{45} and α_{90} come from changes

23.36.2 Load curves in different directions

A project by Fleischer et.al. [2007] resulted in the option HR = 7, see *MAT_036 in LS-DYNA Keyword User's Manual, which allows for specification of different hardening curves in the three directions corresponding to 0, 45 and 90 degrees. In addition, the R-values in these directions can be specified as functions of plastic strains, both features make up an interesting extension to the standard form of the material model. This section is devoted to the theory of the hardening option while the theory for the R-values follows in the next section.

23.36.2.1 An introductory remark

This material typically fits three Lankford parameters and the yield stress in the 00 direction. This fit will result in a non-intuitive effective stress-strain relationship for uniaxial tension in other directions. To explain this we assume that we pull in the φ direction giving a uniaxial stress value of σ_φ and a corresponding plastic strain component ε_φ^p . The relation between the stress value σ_φ and effective stress σ_{eff} is given by

$$\sigma_{\text{eff}} = k_\varphi \sigma_\varphi$$

where $k_\varphi = 1$ if $\varphi = 0$ but not in general. The plastic work relation, which defines the effective plastic strain for the current material, gives the following expression for the effective plastic strain

$$\varepsilon_p = \frac{\varepsilon_\varphi^p \sigma_\varphi}{\sigma_{\text{eff}}}.$$

This means that there is a relationship with a stress-strain hardening curve using the effective stress and strain and a corresponding stress-strain hardening curve using the actual stress and strain values. Assume that a test reveals that the hardening is given by the curve

$$\sigma_\varphi = \sigma_\varphi(\varepsilon_\varphi^p)$$

and we want to determine the hardening curve used by LS-DYNA

$$\sigma_{\text{eff}} = \sigma_{\text{eff}}(\varepsilon_p),$$

then using the relationships above yields

$$\sigma_{\text{eff}}(\varepsilon_p) = k_\varphi \sigma_\varphi(k_\varphi \varepsilon_p).$$

Consequently a user input hardening curve must internally be transformed to an effective hardening curve to be used in the material model to get the desired behavior. Still, the effective plastic strain is not going to be equal to the plastic strain component in the tensile direction and validation of the hardening behavior is not straightforward. Therefore we introduce a new effective plastic strain $\tilde{\varepsilon}_p$ with evolution given by

$$\frac{d\tilde{\varepsilon}_p}{dt} = k_\varphi \frac{d\varepsilon_p}{dt}$$

that can be used to verify the hardening relationship. This is actually the von Mises plastic strain in the work hardening sense and is output to the d3plot database as history variable #2 for post-processing.

23.36.2.2 The model

The load curve hardening option can be generalized to allow different hardening curves in the 00, 45 and 90 directions, as well as balanced biaxial and shear loading. To this end we let the yield value be given as a convex combination of the hardening curves in each direction as

$$\sigma_Y(\sigma, \varepsilon_p) = \alpha_{00}\sigma_Y^{00}(\varepsilon_p) + \alpha_{45}\sigma_Y^{45}(\varepsilon_p) + \alpha_{90}\sigma_Y^{90}(\varepsilon_p) + \alpha_{\text{bi}}\sigma_Y^{\text{bi}}(\varepsilon_p) + \alpha_{\text{sh}}\sigma_Y^{\text{sh}}(\varepsilon_p)$$

where $\sigma_Y^{00}(\varepsilon_p)$ is the user defined hardening curve in direction 00, and the others are the internal hardening curves in the other directions (see the section 23.36.2.1 and figure 23-1). The convex parameters must fulfill

$$0 \leq \alpha_{00} \leq 1, 0 \leq \alpha_{45} \leq 1, 0 \leq \alpha_{90} \leq 1, 0 \leq \alpha_{\text{bi}} \leq 1, 0 \leq \alpha_{\text{sh}} \leq 1,$$

$$\alpha_{00} + \alpha_{45} + \alpha_{90} + \alpha_{\text{bi}} + \alpha_{\text{sh}} = 1,$$

and depend on the stress state. Furthermore $\alpha_{00} = 1$ must mean that the stress is uniaxial and is directed in the 00 direction, and that the same thing holds for the other directions. To accomplish this we reason as follows.

Let $\hat{\sigma}_{ij}$ be the normalized in-plane stress components, i.e.,

$$\hat{\sigma}_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{11}^2 + \sigma_{22}^2 + 2\sigma_{12}^2}}$$

and let σ be the largest eigenvalue to this matrix and q_i the associated eigenvector components. Furthermore we define $\sigma_v = (\hat{\sigma}_{11} + \hat{\sigma}_{22})/2$ as the normalized volumetric stress

and $\sigma_d = \sqrt{\frac{(\hat{\sigma}_{11} - \hat{\sigma}_{22})^2}{4} + \hat{\sigma}_{12}^2}$ the normalized shear stress, and make a note that $0 \leq \sigma_d \leq 1/\sqrt{2}$ and

$$\sigma_d = 0 \rightarrow \text{biaxial stress state}$$

$$\sigma_d = 1/2 \rightarrow \text{uniaxial stress state}$$

$$\sigma_d = 1/\sqrt{2} \rightarrow \text{shear stress state.}$$

If $a = 2\sigma_v^2$ is the fraction of stress that is volumetric and $b = \sigma^2$ an indicator of uniaxial stress state, then $c = a(1 - 4\{b - 1/2\}^2)$ is a normalized measure that indicates when the stress is deviatoric/uniaxial or volumetric. That is, $c = 0$ means that the stress is deviatoric or uniaxial and $c = 1$ means that it is volumetric.

Now, let $q = 4q_1^2(1 - q_1^2)$ be the fraction of the eigenvector q_i that points in the 00 or 90 direction. That is, $q = 1$ means that the eigenvector points in the 45 direction and $q = 0$ means that it is pointing in either the 00 or 90 direction. Moreover, the same way of reasoning is valid for the eigenvector associated to the smallest eigenvalue.

To determine in which direction of 00 or 90 a certain eigenvector is pointing we introduce $d = bq_1^2 + (1 - b)(1 - q_1^2)$, and deduce that $d = 1$ means that the eigenvector q_i points in the 00 direction and $d = 0$ means that it is pointing in the 90 direction.

We are now ready to give partial expressions for the three uniaxial convex parameters

$$\tilde{\alpha}_{00} = (1 - c)d(1 - q) + c/4$$

$$\tilde{\alpha}_{45} = (1 - c)q + c/2$$

$$\tilde{\alpha}_{90} = (1 - c)(1 - d)(1 - q) + c/4$$

and these are completed by means of adding the biaxial and shear parts

$$\alpha_{bi} = \max(0, 1 - 4\sigma_d^2)$$

$$\alpha_{sh} = \max(0, 4\sigma_d^2 - 1)$$

$$\alpha_{00} = \tilde{\alpha}_{00}(1 - \alpha_{bi} - \alpha_{sh})$$

$$\alpha_{45} = \tilde{\alpha}_{45}(1 - \alpha_{bi} - \alpha_{sh})$$

$$\alpha_{90} = \tilde{\alpha}_{90}(1 - \alpha_{bi} - \alpha_{sh})$$

This set of parameters fulfills the requirements mentioned above and allows for a decent expression for a directional dependent yield stress.

In the consistency condition we do not consider the derivatives of the convex parameters with respect to the stress, as we assume that these will not have a major impact on convergence.

23.36.3 Variable Lankford parameters

The R-values are supposed to be variable with deformation, and we let $R_{00}(\varepsilon_p)$, $R_{45}(\varepsilon_p)$ and $R_{90}(\varepsilon_p)$ be the internal load curves that are transformed from the ones given by the user. Then we define the directional dependent R-value according to

$$R(\sigma, \varepsilon_p) = \alpha_{00}R_{00}(\varepsilon_p) + \alpha_{45}R_{45}(\varepsilon_p) + \alpha_{90}R_{90}(\varepsilon_p)$$

using the same set of convex parameters as in the previous section. A generalized relation for the R-value in terms of the stress can be given as

$$(\hat{\sigma}_{22} + \{\hat{\sigma}_{11} + \hat{\sigma}_{22}\}R)n_1 + (\hat{\sigma}_{11} + \{\hat{\sigma}_{11} + \hat{\sigma}_{22}\}R)n_2 - \hat{\sigma}_{12}n_4 = 0,$$

where $\hat{\sigma}_{ij}$ are the normalized stress components, n_i is the direction of plastic flow and where we have suppressed the dependence of stress and plastic strain in R . By setting

$$\Delta n_1 = n_1 - \frac{\partial f}{\partial \sigma_{11}}, \Delta n_2 = n_2 - \frac{\partial f}{\partial \sigma_{22}}, \Delta n_4 = n_4 - \frac{\partial f}{\partial \sigma_{12}}$$

and

$$\Delta R = R(\varepsilon_p) - R(0)$$

we can simplify this equation as

$$\begin{aligned} &(\hat{\sigma}_{22} + \{\hat{\sigma}_{11} + \hat{\sigma}_{22}\}R)\Delta n_1 + (\hat{\sigma}_{11} + \{\hat{\sigma}_{11} + \hat{\sigma}_{22}\}R)\Delta n_2 - \hat{\sigma}_{12}\Delta n_4 = \\ &\quad - \left(\frac{\partial f}{\partial \sigma_{11}} + \frac{\partial f}{\partial \sigma_{22}} \right) \{\hat{\sigma}_{11} + \hat{\sigma}_{22}\} \Delta R \end{aligned}$$

assuming that the relation already holds for the yield surface normal and R-value in the reference configuration.

This equation is complemented with a consistency condition of the plastic flow

$$\hat{\sigma}_{11}\Delta n_1 + \hat{\sigma}_{22}\Delta n_2 + \hat{\sigma}_{12}\Delta n_4 = 0.$$

These two equations are linearly independent if and only if

$$\{\hat{\sigma}_{11} + \hat{\sigma}_{22}\} \sqrt{\frac{(\hat{\sigma}_{11} - \hat{\sigma}_{22})^2}{4}} + \hat{\sigma}_{12}^2 \neq 0$$

and then the equation

$$-\hat{\sigma}_{12}\Delta n_1 + \hat{\sigma}_{12}\Delta n_2 + (\hat{\sigma}_{11} - \hat{\sigma}_{22})\Delta n_4 = 0$$

can be used to complement the previous two. This defines a system of equations that can be used to solve in least square sense for the perturbation Δn_i of the yield surface normal to get the R-value of interest. To avoid numerical problems and make the perturbation continuous with respect to the stress, the right hand side of the first equation is changed to

$$-\left(\frac{\partial f}{\partial \sigma_{11}} + \frac{\partial f}{\partial \sigma_{22}}\right)\{\hat{\sigma}_{11} + \hat{\sigma}_{22}\}((\hat{\sigma}_{11} - \hat{\sigma}_{22})^2 + 4\hat{\sigma}_{12}^2)\Delta R.$$

This results in a non-associated flow rule, meaning that the plastic flow is not in the direction of the yield surface normal. Again, we don't take any special measures into account for the stress return algorithm as we believe that the perturbation of the normal is small enough not to deteriorate convergence. In [figure 23-1](#) the plastic flow direction is illustrated as function of the stress on the yield surface.

23.37 Material Model 37: Transversely Anisotropic Elastic-Plastic

This fully iterative plasticity model is available only for shell elements. The input parameters for this model are: Young's modulus E ; Poisson's ratio ν ; the yield stress; the tangent modulus E_t ; and the anisotropic hardening parameter R .

Consider Cartesian reference axes which are parallel to the three symmetry planes of anisotropic behavior. Then the yield function suggested by Hill [1948] can be written

$$F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 - 1 = 0, \quad (23.37.1)$$

where σ_{y1} , σ_{y2} , and σ_{y3} , are the tensile yield stresses and σ_{y12} , σ_{y23} , and σ_{y31} are the shear yield stresses. The constants F , G , H , L , M , and N are related to the yield stress by

$$\begin{aligned} 2L &= \frac{1}{\sigma_{y23}^2} \\ 2M &= \frac{1}{\sigma_{y31}^2} \\ 2N &= \frac{1}{\sigma_{y12}^2} \\ 2F &= \frac{1}{\sigma_{y2}^2} + \frac{1}{\sigma_{y3}^2} - \frac{1}{\sigma_{y1}^2} \\ 2G &= \frac{1}{\sigma_{y3}^2} + \frac{1}{\sigma_{y1}^2} - \frac{1}{\sigma_{y2}^2} \\ 2H &= \frac{1}{\sigma_{y1}^2} + \frac{1}{\sigma_{y2}^2} - \frac{1}{\sigma_{y3}^2}. \end{aligned} \quad (23.37.2)$$

The isotropic case of von Mises plasticity can be recovered by setting

$$F = G = H = \frac{1}{2\sigma_y^2}, \quad (23.37.3)$$

and

$$L = M = N = \frac{1}{2\sigma_y^2}. \quad (23.37.4)$$

For the particular case of transverse anisotropy, where properties do not vary in the $x_1 - x_2$ plane, the following relations hold:

$$\begin{aligned}
F &= 2G = \frac{1}{\sigma_{y3}^2} \\
2H &= \frac{2}{\sigma_y^2} - \frac{1}{\sigma_{y3}^2} \\
N &= \frac{2}{\sigma_y^2} - \frac{1}{2} \frac{1}{\sigma_{y3}^2},
\end{aligned} \tag{23.37.5}$$

where it has been assumed that $\sigma_{y1} = \sigma_{y2} = \sigma_y$.

Letting $K = \frac{\sigma_y}{\sigma_{y3}}$, the yield criterion can be written

$$F(\sigma) = \sigma_e = \sigma_y, \tag{23.37.6}$$

where

$$\begin{aligned}
F(\sigma) \equiv & \left[\sigma_{11}^2 + \sigma_{22}^2 + K^2 \sigma_{33}^2 - K^2 \sigma_{33} (\sigma_{11} + \sigma_{22}) - (2 - K^2) \sigma_{11} \sigma_{22} \right. \\
& \left. + 2L\sigma_y^2 (\sigma_{23}^2 + \sigma_{31}^2) + 2 \left(2 - \frac{1}{2} K^2 \right) \sigma_{12}^2 \right]^{1/2}.
\end{aligned} \tag{23.37.7}$$

The rate of plastic strain is assumed to be normal to the yield surface so $\dot{\epsilon}_{ij}^P$ is found from

$$\dot{\epsilon}_{ij}^P = \lambda \frac{\partial F}{\partial \sigma_{ij}}. \tag{23.37.8}$$

Now consider the case of plane stress, where $\sigma_{33} = 0$. Also, define the anisotropy input parameter R as the ratio of the in-plane plastic strain rate to the out-of-plane plastic strain rate:

$$R = \frac{\dot{\epsilon}_{22}^P}{\dot{\epsilon}_{33}^P}. \tag{23.37.9}$$

It then follows that

$$R = \frac{2}{K^2} - 1. \tag{23.37.10}$$

Using the plane stress assumption and the definition of R , the yield function may now be written

$$F(\sigma) = \left[\sigma_{11}^2 + \sigma_{22}^2 - \frac{2R}{R+1} \sigma_{11} \sigma_{22} + 2 \frac{2R+1}{R+1} \sigma_{12}^2 \right]^{1/2}. \tag{23.37.11}$$

23.38 Material Model 38: Blatz-Ko Compressible Foam

$$W(I_1, I_2, I_3) = \frac{\mu}{2} \left(\frac{I_2}{I_3} + 2\sqrt{I_3} - 5 \right), \quad (23.38.1)$$

where μ is the shear modulus and I_1 , I_2 , and I_3 are the strain invariants. Blatz and Ko [1962] suggested this form for a 47 percent volume polyurethane foam rubber with a Poisson's ratio of 0.25. The second Piola-Kirchhoff stresses are given as

$$S^{ij} = \mu \left[(I\delta_{ij} - G_{ij}) \frac{1}{I_3} + \left(\sqrt{I_3} - \frac{I_2}{I_3} \right) G^{ij} \right], \quad (23.38.2)$$

where $G_{ij} = \frac{\partial x_k}{\partial X_i} \frac{\partial x_k}{\partial X_j}$, $G^{ij} = \frac{\partial X_i}{\partial x_k} \frac{\partial X_j}{\partial x_k}$, after determining S_{ij} , it is transformed into the Cauchy stress tensor:

$$\sigma_{ij} = \frac{\rho}{\rho_0} \frac{\partial x_i}{\partial X_k} \frac{\partial x_j}{\partial X_l} S_{kl}, \quad (23.38.3)$$

where ρ_0 and ρ are the initial and current density, respectively.

23.39 Material Model 39: Transversely Anisotropic Elastic-Plastic With FLD

See Material Model 37 for the similar model theoretical basis. The first history variable is the maximum strain ratio defined by:

$$\frac{\epsilon_{\text{major_workpiece}}}{\epsilon_{\text{major_fld}}} \quad (23.39.1)$$

corresponding to $\epsilon_{\text{minor_workpiece}}$. This history variable replaces the effective plastic strain in the output. Plastic strains can still be obtained but one additional history variable must be written into the D3PLOT database.

The strains on which these calculations are based are integrated in time from the strain rates:

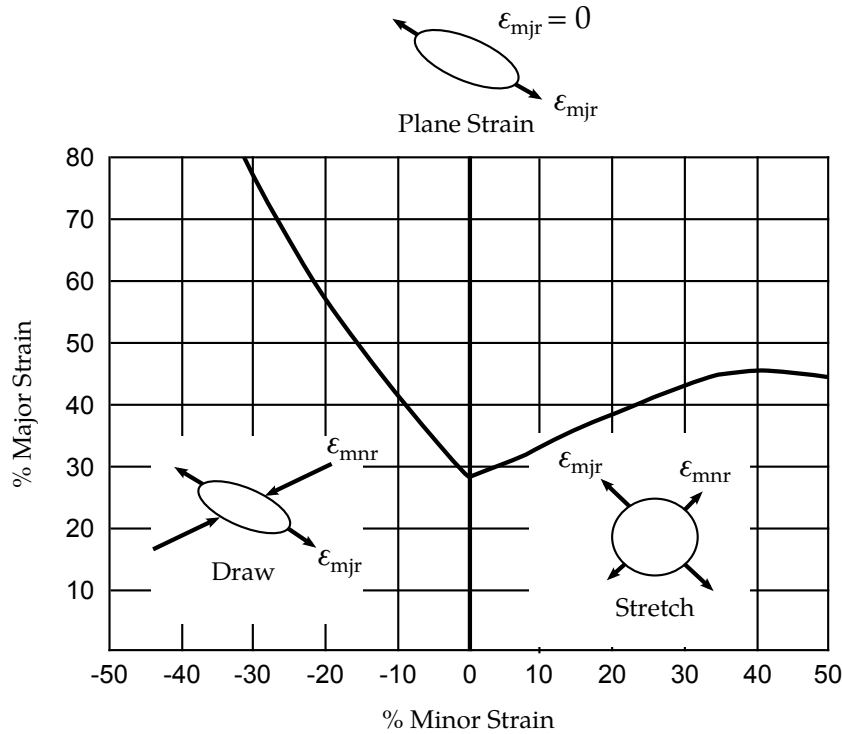


Figure 23.39.1. Flow limit diagram.

$$\epsilon_{ij}^{n+1} = \epsilon_{ij}^n + \epsilon_{ij}^{\nabla n+1/2} \Delta t^{n+1/2}, \quad (23.39.2)$$

and are stored as history variables. The resulting strain measure is logarithmic.

23.42 Material Model 42: Planar Anisotropic Plasticity Model

This model is built into LS-DYNA as a user material model for modeling plane stress anisotropic plasticity in shells. Please note that only three cards are input here. The orthotropic angles must be defined later as for all materials of this type. This model is currently not vectorized.

This is an implementation of an anisotropic plasticity model for plane stress where the flow rule, see Material Type 37, simplifies to:

$$F(\sigma_{22})^2 + G(\sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2N\sigma_{12}^2 - 1 = 0. \quad (23.42.1)$$

The anisotropic parameters $R00$, $R45$, and $R90$ are defined in terms of F , G , H , and N as [Hill, 1989]:

$$\begin{aligned} R00 &= \frac{H}{G}, \\ 2R45 &= \frac{2N}{(F + G)} - 1, \\ R90 &= \frac{H}{F}. \end{aligned} \quad (23.42.2)$$

The yield function for this model is given as:

$$\sigma_y = A\varepsilon^m \dot{\varepsilon}^n. \quad (23.42.3)$$

To avoid numerical problems the minimum strain rate, $\dot{\varepsilon}_{\min}$ must be defined and the initial yield stress σ_0 is calculated as

$$\sigma_0 = A\varepsilon_0^m \dot{\varepsilon}_{\min}^n = E\varepsilon_0, \quad (23.42.4)$$

$$\varepsilon_0 = \left(\frac{E}{A\dot{\varepsilon}_{\min}^n} \right)^{\frac{1}{m-1}}. \quad (23.42.5)$$

23.51 Material Model 51: Temperature and Rate Dependent Plasticity

The kinematics associated with the model are discussed in references [Hill 1948, Bammann and Aifantis 1987, Bammann 1989]. The description below is taken nearly verbatim from Bammann [Hill 1948].

With the assumption of linear elasticity we can write,

$$\overset{\circ}{\sigma} = \lambda \text{tr}(\mathbf{D}^e) \mathbf{1} + 2\mu \mathbf{D}^e, \quad (23.51.1)$$

where, the Cauchy stress σ is convected with the elastic spin \mathbf{W}^e as,

$$\overset{\circ}{\sigma} = \dot{\sigma} - \mathbf{W}^e \sigma + \sigma \mathbf{W}^e. \quad (23.51.2)$$

This is equivalent to writing the constitutive model with respect to a set of directors whose direction is defined by the plastic deformation [Bammann and Aifantis 1987, Bammann and Johnson 1987]. Decomposing both the skew symmetric and symmetric parts of the velocity gradient into elastic and plastic parts we write for the elastic stretching \mathbf{D}^e and the elastic spin \mathbf{W}^e ,

$$\mathbf{D}^e = \mathbf{D} - \mathbf{D}^p - \mathbf{D}^{\text{th}}, \mathbf{W}^e = \mathbf{W} = \mathbf{W}^p. \quad (23.51.3)$$

Within this structure it is now necessary to prescribe an equation for the plastic spin \mathbf{W}^p in addition to the normally prescribed flow rule for \mathbf{D}^p and the stretching due to the thermal expansion \mathbf{D}^{th} . As proposed, we assume a flow rule of the form,

$$\mathbf{D}^p = f(T) \sinh \left[\frac{|\tilde{\xi}| - \kappa - Y(T)}{V(T)} \right] \frac{\tilde{\xi}'}{|\tilde{\xi}'|}, \quad (23.51.4)$$

where T is the temperate, κ is the scalar hardening variable, $\tilde{\xi}'$ is the difference between the deviatoric Cauchy stress σ' and the tensor variable α' ,

$$\tilde{\xi}' = \sigma' - \alpha', \quad (23.51.5)$$

and $f(T)$, $Y(T)$, $V(T)$ are scalar functions whose specific dependence upon the temperature is given below. Assuming isotropic thermal expansion, and introducing the expansion coefficient \dot{A} , the thermal stretching can be written,

$$\mathbf{D}^{\text{th}} = \dot{A} \mathbf{1}. \quad (23.51.6)$$

The evolution of the internal variables α and κ are prescribed in a hardening minus recovery format as,

$$\dot{\alpha} = h(T) \mathbf{D}^p - [r_d(T) |\mathbf{D}^p| + r_s(T)] |\alpha| \alpha, \quad (23.51.7)$$

$$\dot{\kappa} = H(T)\mathbf{D}^P - [R_d(T)|\mathbf{D}^P| - R_s(T)]\kappa^2, \quad (23.51.8)$$

where h and H are the hardening moduli, $r_s(T)$ and $R_s(T)$ are scalar functions describing the diffusion controlled 'static' or 'thermal' recovery, and $r_d(T)$ and $R_d(T)$ are the functions describing dynamic recovery.

If we assume that $\mathbf{W}^P = 0$, we recover the Jaumann stress rate which results in the prediction of an oscillatory shear stress response in simple shear when coupled with a Prager kinematic hardening assumption [Johnson and Bammann 1984]. Alternatively we can choose,

$$\mathbf{W}^P = \mathbf{R}^T \dot{\mathbf{U}} \mathbf{U}^{-1} \mathbf{R}, \quad (23.51.9)$$

which recovers the Green-Naghdi rate of Cauchy stress and has been shown to be equivalent to Mandel's isoclinic state [Bammann and Aifantis 1987]. The model employing this rate allows a reasonable prediction of directional softening for some materials but in general under-predicts the softening and does not accurately predict the axial stresses which occur in the torsion of the thin walled tube.

The final equation necessary to complete our description of high strain rate deformation is one which allows us to compute the temperature change during the deformation. In the absence of a coupled thermomechanical finite element code we assume adiabatic temperature change and follow the empirical assumption that 90 - 95% of the plastic work is dissipated as heat. Hence,

$$\dot{T} = \frac{0.9}{\rho C_v} (\boldsymbol{\sigma} \cdot \mathbf{D}^P), \quad (23.51.10)$$

where ρ is the density of the material and C_v the specific heat.

In terms of the input parameters the functions defined above become:

$V(T) = C1 \bullet \exp(-C2/T)$	$h(T) = C9 \bullet \exp(C10/T)$
$Y(T) = C3 \bullet \exp(C4/T)$	$r_s(T) = C11 \bullet \exp(-C12/T)$
$f(T) = C5 \bullet \exp(-C6/T)$	$R_d(T) = C13 \bullet \exp(-C14/T)$
$r_d(T) = C7 \bullet \exp(-C8/T)$	$H(T) = C15 \bullet \exp(C16/T)$
	$R_s(T) = C17 \bullet \exp(-C18/T)$

and the heat generation coefficient is

$$HC = \frac{0.9}{\rho C_v}. \quad (23.51.11)$$

23.52 Material Model 52: Sandia's Damage Model

The evolution of the damage parameter, ϕ , is defined by [Bammann, et. al., 1990]

$$\phi = \beta \left[\frac{1}{(1 - \phi)^N} - (1 - \phi) \right]^{|D^P|}, \quad (23.52.1)$$

in which

$$\beta = \sin \left[\frac{2(2N - 1)p}{(2N - 1)\bar{\sigma}} \right], \quad (23.52.2)$$

where p is the pressure and $\bar{\sigma}$ is the effective stress.

23.53 Material Model 53: Low Density Closed Cell Polyurethane Foam

A rigid, low density, closed cell, polyurethane foam model developed at Sandia Laboratories [Neilsen et al., 1987] has been recently implemented for modeling impact limiters in automotive applications. A number of such foams were tested at Sandia and reasonable fits to the experimental data were obtained.

In some respects this model is similar to the crushable honeycomb model type 26 in that the components of the stress tensor are uncoupled until full volumetric compaction is achieved. However, unlike the honeycomb model this material possesses no directionality but includes the effects of confined air pressure in its overall response characteristics.

$$\sigma_{ij} = \sigma_{ij}^{\text{sk}} - \delta_{ij}\sigma^{\text{air}}, \quad (23.53.1)$$

where σ_{ij}^{sk} is the skeletal stress and σ^{air} is the air pressure computed from the equation:

$$\sigma^{\text{air}} = -\frac{p_0\gamma}{1 + \gamma - \phi'}, \quad (23.53.2)$$

where p_0 is the initial foam pressure usually taken as the atmospheric pressure and γ defines the volumetric strain

$$\gamma = V - 1 + \gamma_0, \quad (23.53.3)$$

where V is the relative volume and γ_0 is the initial volumetric strain which is typically zero. The yield condition is applied to the principal skeletal stresses which are updated independently of the air pressure. We first obtain the skeletal stresses:

$$\sigma_{ij}^{\text{sk}} = \sigma_{ij} + \sigma_{ij}\sigma^{\text{air}}, \quad (23.53.4)$$

and compute the trial stress, σ_i^{skt}

$$\sigma_{ij}^{\text{skt}} = \sigma_{ij}^{\text{sk}} + E\dot{\epsilon}_{ij}\Delta t, \quad (23.53.5)$$

where E is Young's modulus. Since Poisson's ratio is zero, the update of each stress component is uncoupled and $2G = E$ where G is the shear modulus. The yield condition is applied to the principal skeletal stresses such that if the magnitude of a principal trial stress component, σ_i^{skt} , exceeds the yield stress, σ_y , then

$$\sigma_i^{\text{sk}} = \min(\sigma_y, |\sigma_i^{\text{skt}}|) \frac{\sigma_i^{\text{skt}}}{|\sigma_i^{\text{skt}}|}. \quad (23.53.6)$$

The yield stress is defined by

$$\sigma_y = a + b(1 + c\gamma), \quad (23.53.7)$$

where a , b , and c are user defined input constants. After scaling the principal stresses they are transformed back into the global system and the final stress state is computed

$$\sigma_{ij} = \sigma_{ij}^{\text{sk}} - \delta_{ij}\sigma^{\text{air}}. \quad (23.53.8)$$

23.54 Material Models 54 and 55: Enhanced Composite Damage Model

These models are very close in their formulations. Material 54 uses the Chang matrix failure criterion (as Material 22), and material 55 uses the Tsay-Wu criterion for matrix failure.

Arbitrary orthotropic materials, e.g., unidirectional layers in composite shell structures can be defined. Optionally, various types of failure can be specified following either the suggestions of [Chang and Chang, 1984] or [Tsai and Wu, 1981]. In addition special measures are taken for failure under compression. See [Matzenmiller and Schweizerhof, 1990]. This model is only valid for thin shell elements.

The Chang/Chang criteria is given as follows: for the tensile fiber mode,

$$\sigma_{aa} > 0 \quad \text{then} \quad e_f^2 = \left(\frac{\sigma_{aa}}{X_t} \right)^2 + \beta \left(\frac{\sigma_{ab}}{S_c} \right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \quad (23.54.1)$$

$$E_a = E_b = G_{ab} = \nu_{ba} = \nu_{ab} = 0, \quad (23.54.2)$$

for the compressive fiber mode,

$$\sigma_{aa} < 0 \quad \text{then} \quad e_c^2 = \left(\frac{\sigma_{aa}}{X_c} \right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \quad (23.54.3)$$

$$E_a = \nu_{ba} = \nu_{ab} = 0, \quad (23.54.4)$$

for the tensile matrix mode,

$$\sigma_{bb} > 0 \quad \text{then} \quad e_m^2 = \left(\frac{\sigma_{bb}}{Y_t} \right)^2 + \left(\frac{\sigma_{ab}}{S_c} \right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \quad (23.54.5)$$

$$E_b = \nu_{ba} = 0 \quad \rightarrow \quad G_{ab} = 0, \quad (23.54.6)$$

and for the compressive matrix mode,

$$\sigma_{bb} < 0 \quad \text{then} \quad e_d^2 = \left(\frac{\sigma_{bb}}{2S_c} \right)^2 + \left[\left(\frac{Y_c}{2S_c} \right)^2 - 1 \right] \frac{\sigma_{bb}}{Y_c} + \left(\frac{\sigma_{ab}}{S_c} \right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \quad (23.54.7)$$

$$E_b = \nu_{ba} = \nu_{ab} = 0 \Rightarrow \begin{cases} G_{ab} = 0 \\ X_C = 2Y_C, \text{ for 50\% fiber volume} \end{cases} \quad (23.54.8)$$

In the Tsay/Wu criteria the tensile and compressive fiber modes are treated as in the Chang/Chang criteria. The failure criterion for the tensile and compressive matrix mode is given as:

$$e_{\text{md}}^2 = \frac{\sigma_{\text{bb}}^2}{Y_c Y_t} + \left(\frac{\sigma_{\text{ab}}}{S_c} \right)^2 + \frac{(Y_c - Y_t) \sigma_{\text{bb}}}{Y_c Y_t} - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \quad (23.54.9)$$

For $\beta = 1$ we get the original criterion of Hashin [1980] in the tensile fiber mode. For $\beta = 0$, we get the maximum stress criterion which is found to compare better to experiments.

Failure can occur in any of four different ways:

1. If DFAILT is zero, failure occurs if the Chang/Chang failure criterion is satisfied in the tensile fiber mode.
2. If DFAILT is greater than zero, failure occurs if the tensile fiber strain is greater than DFAILT or less than DFAILC.
3. If EFS is greater than zero, failure occurs if the effective strain is greater than EFS.
4. If TFAIL is greater than zero, failure occurs according to the element time step as described in the definition of TFAIL above.

When failure has occurred in all the composite layers (through-thickness integration points), the element is deleted. Elements which share nodes with the deleted element become “crashfront” elements and can have their strengths reduced by using the SOFT parameter with TFAIL greater than zero.

Information about the status in each layer (integration point) and element can be plotted using additional integration point variables. The number of additional integration point variables for shells written to the LS-DYNA database is input by the *DATABASE_BINARY definition as variable NEIPS. For Models 54 and 55 these additional variables are tabulated below (i = shell integration point):

History Variable	Description	Value	LS-PREPOST History Variable
1. $ef(i)$	tensile fiber mode	1 – elastic 0 – failed	1
2. $ec(i)$	compressive fiber mode		2
3. $em(i)$	tensile matrix mode		3
4. $ed(i)$	compressive matrix mode		4
5. $efail$	$\max[ef(ip)]$		5
6. dam	damage parameter	-1 - element intact 10-8 - element in crashfront +1 - element failed	6

The following components, defined by the sum of failure indicators over all through-thickness integration points, are stored as element component 7 instead of the effective plastic strain.:

Description	Integration point
$\frac{1}{nip} \sum_{i=1}^{nip} ef(i)$	1
$\frac{1}{nip} \sum_{i=1}^{nip} ec(i)$	2
$\frac{1}{nip} \sum_{i=1}^{nip} cm(i)$	2

23.57 Material Model 57: Low Density Urethane Foam

The urethane foam model is available to model highly compressible foams such as those used in seat cushions and as padding on the Side Impact Dummy (SID). The compressive behavior is illustrated in Figure 23.57.1 where hysteresis on unloading is shown. This behavior under uniaxial loading is assumed not to significantly couple in the transverse directions. In tension the material behaves in a linear fashion until tearing occurs. Although our implementation may be somewhat unusual, it was first motivated by Shkolnikov [1991] and a paper by Storakers [1986]. The recent additions necessary to model hysteretic unloading and rate effects are due to Chang, et al., [1994]. These latter additions have greatly expanded the usefulness of this model.

The model uses tabulated input data for the loading curve where the nominal stresses are defined as a function of the elongations, ε_i , which are defined in terms of the principal stretches, λ_i , as:

$$\varepsilon_i = \lambda_i - 1. \quad (23.57.1)$$

The stretch ratios are found by solving for the eigenvalues of the left stretch tensor, V_{ij} , which is obtained via a polar decomposition of the deformation gradient matrix, F_{ij} :

$$F_{ij} = R_{ik}U_{kj} = V_{ik}R_{kj}. \quad (23.57.2)$$

The update of V_{ij} follows the numerically stable approach of Taylor and Flanagan [1989]. After solving for the principal stretches, the elongations are computed and, if the elongations are compressive, the corresponding values of the nominal stresses, τ_i are interpolated. If the elongations are tensile, the nominal stresses are given by

$$\tau_i = E\varepsilon_i. \quad (23.57.3)$$

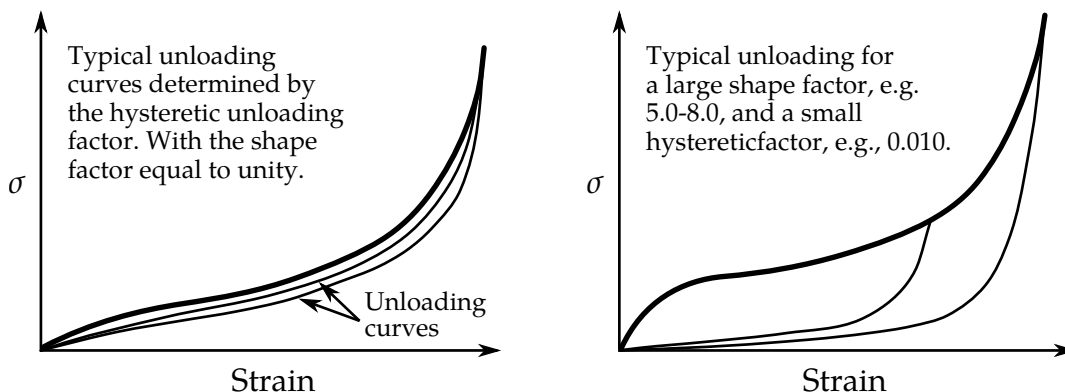


Figure 23.57.1. Behavior of the low-density urethane foam model.

The Cauchy stresses in the principal system become

$$\sigma_i = \frac{\tau_i}{\lambda_j \lambda_k}. \quad (23.57.4)$$

The stresses are then transformed back into the global system for the nodal force calculations.

When hysteretic unloading is used, the reloading will follow the unloading curve if the decay constant, β , is set to zero. If β is nonzero the decay to the original loading curve is governed by the expression:

$$1 - e^{-\beta t}. \quad (23.57.5)$$

The bulk viscosity, which generates a rate dependent pressure, may cause an unexpected volumetric response and, consequently, it is optional with this model.

Rate effects are accounted for through linear viscoelasticity by a convolution integral of the form

$$\sigma_{ij}^r = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.57.6)$$

where $g_{ijkl}(t - \tau)$ is the relaxation function. The stress tensor, σ_{ij}^r , augments the stresses determined from the foam, σ_{ij}^f ; consequently, the final stress, σ_{ij} , is taken as the summation of the two contributions:

$$\sigma_{ij} = \sigma_{ij}^f + \sigma_{ij}^r. \quad (23.57.7)$$

Since we wish to include only simple rate effects, the relaxation function is represented by one term from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^N \alpha_m e^{-\beta t}, \quad (23.57.8)$$

given by,

$$g(t) = E_d e^{-\beta_1 t}. \quad (23.57.9)$$

This model is effectively a Maxwell fluid which consists of a damper and spring in series. We characterize this in the input by a Young's modulus, E_d , and decay constant, β_1 . The formulation is performed in the local system of principal stretches where only the principal values of stress are computed and triaxial coupling is avoided. Consequently, the one-dimensional nature of this foam material is unaffected by this addition of rate effects. The addition of rate effects necessitates twelve additional history variables per integration point. The cost and memory overhead of this model comes primarily from the need to "remember" the local system of principal stretches.

Viscous damping is implemented by incrementation of the principal stress components. Firstly, we let

$$a = \frac{c\rho\mu L_e}{1 + \gamma}, \quad (23.57.10)$$

where c , ρ , μ , L_e and γ respectively denote the material sound speed, density, viscous coefficient, element characteristic length and element volumetric strain. The incremental stress components due to viscous damping are then given by

$$\Delta\sigma_i = a \left(\frac{\dot{\epsilon}_i - \dot{\epsilon}_m}{1 + \nu} + \frac{\dot{\epsilon}_m}{1 - 2\nu} \right) \quad i = 1,2,3 \quad (23.57.11)$$

and

$$\Delta\sigma_i = \frac{a\dot{\epsilon}_i}{2(1 + \nu)} \quad i = 4,5,6, \quad (23.57.12)$$

where $\dot{\epsilon}_i$ are the strain rates and $\dot{\epsilon}_m = \sum_{i=1}^3 \dot{\epsilon}_i / 3$.

23.58 Material Model 58: Laminated Composite Fabric

Parameters to control failure of an element layer are: ERODS, the maximum effective strain, i.e., maximum 1 = 100% straining. The layer in the element is completely removed after the maximum effective strain (compression/tension including shear) is reached.

The stress limits are factors used to limit the stress in the softening part to a given value,

$$\sigma_{\min} = \text{SLIMxx} \cdot \text{strength}, \quad (23.58.1)$$

thus, the damage value is slightly modified such that elastoplastic like behavior is achieved with the threshold stress. As a factor for SLIMxx a number between 0.0 and 1.0 is possible. With a factor of 1.0, the stress remains at a maximum value identical to the strength, which is similar to ideal elastoplastic behavior. For tensile failure a small value for SLIMTx is often reasonable; however, for compression SLIMCx = 1.0 is preferred. This is also valid for the corresponding shear value. If SLIMxx is smaller than 1.0 then localization can be observed depending on the total behavior of the lay-up. If the user is intentionally using SLIMxx < 1.0, it is generally recommended to avoid a drop to zero and set the value to something in between 0.05 and 0.10. Then elastoplastic behavior is achieved in the limit which often leads to less numerical problems. Defaults for SLIMXX = 1.0E-8.

The crashfront-algorithm is started if and only if a value for TSIZE (time step size, with element elimination after the actual time step becomes smaller than TSIZE) is input .

The damage parameters can be written to the postprocessing database for each integration point as the first three additional element variables and can be visualized.

Material models with FS = 1 or FS = -1 are favorable for complete laminates and fabrics, as all directions are treated in a similar fashion.

For material model FS = 1 an interaction between normal stresses and shear stresses is assumed for the evolution of damage in the a- and b- directions. For the shear damage is always the maximum value of the damage from the criterion in a- or b- direction is taken.

For material model FS = -1 it is assumed that the damage evolution is independent of any of the other stresses. A coupling is present only via the elastic material parameters and the complete structure.

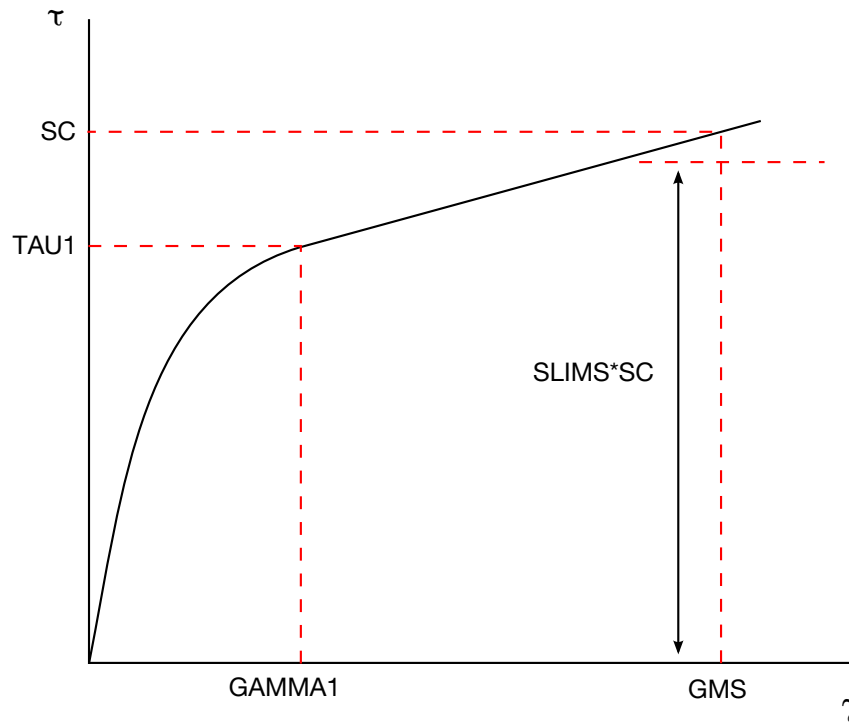


Figure 23.58.1. Stress-strain diagram for shear.

In tensile and compression directions and in a- as well as in b- direction, different failure surfaces can be assumed. The damage values, however, increase only when the loading direction changes.

Special control of shear behavior of fabrics

For fabric materials a nonlinear stress strain curve for the shear part of failure surface $FS = -1$ can be assumed as given below. This is not possible for other values of FS .

The curve, shown in [Figure 23.58.1](#), is defined by three points:

- the origin (0,0) is assumed,
- the limit of the first slightly nonlinear part (must be input), stress (TAU1) and strain (GAMMA1), see below.
- the shear strength at failure and shear strain at failure.

In addition a stress limiter can be used to keep the stress constant via the SLIMS parameter. This value must be less than or equal to 1.0 and positive, which leads to an elastoplastic behavior for the shear part. The default is 1.0E-08, assuming almost brittle failure once the strength limit SC is reached.

23.60 Material Model 60: Elastic With Viscosity

This material model was developed to simulate the forming of glass products (e.g., car windshields) at high temperatures. Deformation is by viscous flow but elastic deformations can also be large. The material model, in which the viscosity may vary with temperature, is suitable for treating a wide range of viscous flow problems and is implemented for brick and shell elements.

Volumetric behavior is treated as linear elastic. The deviatoric strain rate is considered to be the sum of elastic and viscous strain rates:

$$\dot{\epsilon}'_{\text{total}} = \dot{\epsilon}'_{\text{elastic}} + \dot{\epsilon}'_{\text{viscous}} = \frac{\dot{\sigma}'}{2G} + \frac{\dot{\sigma}'}{2\nu'} \quad (23.60.1)$$

where G is the elastic shear modulus, ν is the viscosity coefficient. The stress increment over one time step dt is

$$d\sigma' = 2G\dot{\epsilon}'_{\text{total}}dt - \frac{G}{\nu}dt\sigma'. \quad (23.60.2)$$

The stress before the update is used for σ' . For shell elements, the through-thickness strain rate is calculated as follows

$$d\sigma_{33} = 0 = K(\dot{\epsilon}_{11} + \dot{\epsilon}_{22} + \dot{\epsilon}_{33})dt + 2G\dot{\epsilon}'_{33}dt - \frac{G}{\nu}dt\sigma'_{33}, \quad (23.60.3)$$

where the subscript $ij = 33$ denotes the through-thickness direction and K is the elastic bulk modulus. This leads to:

$$\dot{\epsilon}_{33} = -a(\dot{\epsilon}_{11} + \dot{\epsilon}_{22}) + bp, \quad (23.60.4)$$

$$a = \frac{K - \frac{2}{3}G}{K + \frac{4}{3}G}, \quad (23.60.5)$$

$$b = \frac{Gdt}{\nu\left(K + \frac{4}{3}G\right)}, \quad (23.60.6)$$

in which p is the pressure defined as the negative of the hydrostatic stress.

23.61 Material Model 61: Maxwell/Kelvin Viscoelastic with Maximum Strain

The shear relaxation behavior is described for the Maxwell model by:

$$G(t) = G_{\infty} + (G_0 - G_{\infty})e^{-\beta t}. \quad (23.61.1)$$

A Jaumann rate formulation is used

$$s'_{ij}^{\nabla} = 2 \int_0^t G(t - \tau) \dot{\epsilon}'_{ij}(\tau) d\tau, \quad (23.61.2)$$

where the prime denotes the deviatoric part of the stress rate, s'_{ij}^{∇} , and $\dot{\epsilon}'_{ij}$ is the deviatoric strain rate.

For the Kelvin model the stress evolution equation is defined as:

$$\dot{s}_{ij} + \frac{1}{\tau} s_{ij} = (1 + \delta_{ij}) G_0 \dot{\epsilon}'_{ij} + (1 + \delta_{ij}) \frac{G_{\infty}}{\tau} \epsilon'_{ij}, \quad (23.61.3)$$

where δ_{ij} is the Kronecker delta, G_0 is the instantaneous shear modulus, G_{∞} is the long term shear modulus, and τ is the decay constant.

The pressure is determined from the bulk modulus and the volumetric strain:

$$p = -K \epsilon_v, \quad (23.61.4)$$

where

$$\epsilon_v = \ln \left(\frac{V}{V_0} \right), \quad (23.61.5)$$

defines the logarithmic volumetric strain from the relative volume.

Bandak's [1991] calculation of the total strain tensor, ϵ_{ij} , for output uses an incremental update based on Jaumann rate:

$$\epsilon_{ij}^{n+1} = \epsilon_{ij}^n + r_{ij}^n + \epsilon_{ij}^{\nabla n+1/2} \Delta t^{n+1/2}, \quad (23.61.6)$$

where

$$\Delta \epsilon_{ij}^{n+1/2} = \dot{\epsilon}_{ij}^{n+1/2} \Delta t^{n+1/2}, \quad (23.61.7)$$

and r_{ij}^n gives the rotation of the strain tensor at time t^n to the configuration at t^{n+1}

$$r_{ij}^n = \left(\epsilon_{ip}^n \omega_{pj}^{n+1/2} + \epsilon_{jp}^n \omega_{pi}^{n+1/2} \right) \Delta t^{n+1/2}. \quad (23.61.8)$$

23.62 Material Model 62: Viscous Foam

This model was written to represent the energy absorbing foam found on certain crash dummies, i.e., the ‘Confor Foam’ covering the ribs of the Eurosid dummy.

The model consists of a nonlinear elastic stiffness in parallel with a viscous damper. A schematic is shown in [Figure 23.62.1](#). The elastic stiffness is intended to limit total crush while the viscous damper absorbs energy. The stiffness E_2 prevents timestep problems.

Both E_1 and V_2 are nonlinear with crush as follows:

$$\begin{aligned} E_1^t &= E_1(V^{-n_1}), \\ V_2^t &= V_2(\text{abs}(1 - V))^{n_2}, \end{aligned} \quad (23.62.1)$$

where V is the relative volume defined by the ratio of the current to initial volume. Typical values are (units of N, mm, s)

$$\begin{aligned} E_1 &= 0.0036, \\ n_1 &= 4.0, \\ V_2 &= 0.0015, \\ E_2 &= 100.0, \\ n_2 &= 0.2, \\ \nu &= 0.05. \end{aligned} \quad (23.62.2)$$

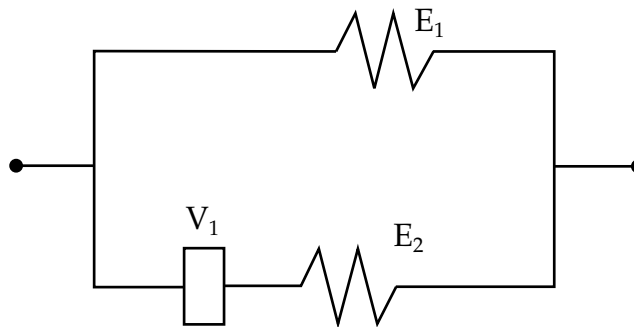


Figure 23.62.1. Schematic of Material Model 62.

23.63 Material Model 63: Crushable Foam

The intent of this model is to model crushable foams in side impact and other applications where cyclic behavior is unimportant.

This isotropic foam model crushes one-dimensionally with a Poisson's ratio that is essentially zero. The stress versus strain behavior is depicted in [Figure 23.63.1](#) where an example of unloading from point a to the tension cutoff stress at b then unloading to point c and finally reloading to point d is shown. At point d the reloading will continue along the loading curve. It is important to use nonzero values for the tension cutoff to prevent the disintegration of the material under small tensile loads. For high values of tension cutoff the behavior of the material will be similar in tension and compression. Viscous damping in the model follows an implementation identical to that of material type 57.

In the implementation we assume that Young's modulus is constant and update the stress assuming elastic behavior.

$$\sigma_{ij}^{\text{trial}} = \sigma_{ij}^n + E \dot{\epsilon}_{ij}^{n+1/2} \Delta t^{n+1/2}. \quad (23.63.1)$$

The magnitudes of the principal values, $\sigma_i^{\text{trial}}, i = 1, 3$ are then checked to see if the yield stress, σ_y , is exceeded and if so they are scaled back to the yield surface:

$$\text{if } \sigma_y < |\sigma_i^{\text{trial}}| \quad \text{then} \quad \sigma_i^{n+1} = \sigma_y \frac{\sigma_i^{\text{trial}}}{|\sigma_i^{\text{trial}}|}. \quad (23.63.2)$$

After the principal values are scaled, the stress tensor is transformed back into the global system. As seen in [Figure 23.63.1](#), the yield stress is a function of the natural logarithm of the relative volume, V , i.e., the volumetric strain.

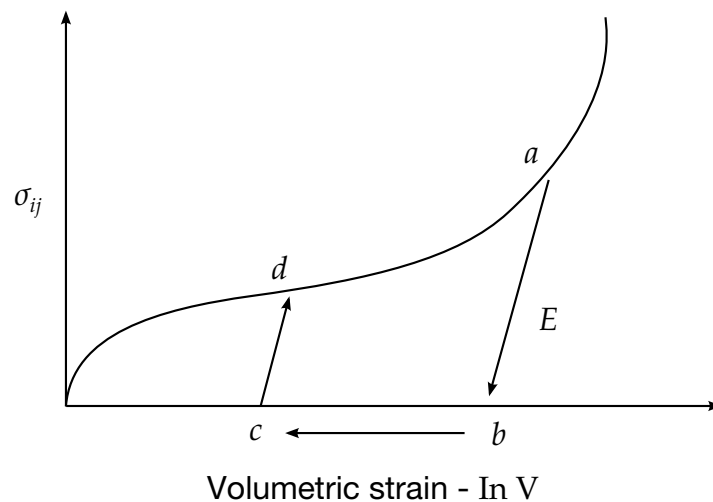


Figure 23.63.1. Yield stress versus volumetric strain curve for the crushable foam.

23.64 Material Model 64: Strain Rate Sensitive Power-Law Plasticity

This material model follows a constitutive relationship of the form:

$$\sigma = k\varepsilon^m \dot{\varepsilon}^n \quad (23.64.1)$$

where σ is the yield stress, ε is the effective plastic strain, $\dot{\varepsilon}$ is the effective plastic strain rate, and the constants k , m , and n can be expressed as functions of effective plastic strain or can be constant with respect to the plastic strain. The case of no strain hardening can be obtained by setting the exponent of the plastic strain equal to a very small positive value, i.e., 0.0001.

This model can be combined with the superplastic forming input to control the magnitude of the pressure in the pressure boundary conditions in order to limit the effective plastic strain rate so that it does not exceed a maximum value at any integration point within the model.

A fully viscoplastic formulation is optional. An additional cost is incurred but the improvement in results can be dramatic.

23.65 Material Model 65: Modified Zerilli/Armstrong

The Armstrong-Zerilli Material Model expresses the flow stress as follows. For fcc metals,

$$\sigma = C_1 + \{C_2(\varepsilon^P)^m [e^{(-C_3+C_4 \ln(\dot{\varepsilon}^*))T}] + C_5\} \left(\frac{\mu(T)}{\mu(293)} \right), \quad (23.65.1)$$

ε^P = effective plastic strain

$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}$ effective plastic strain rate

where $\dot{\varepsilon}_0 = 1, 1e-3, 1e-6$ for time units of seconds, milliseconds, and microseconds, respectively.

For bcc metals,

$$\sigma = C_1 + C_2 e^{(-C_3+C_4 \ln(\dot{\varepsilon}^*))T} + [C_5(\varepsilon^P)^n + C_6] \left(\frac{\mu(T)}{\mu(293)} \right), \quad (23.65.2)$$

where

$$\left(\frac{\mu(T)}{\mu(293)} \right) = B_1 + B_2 T + B_3 T^2. \quad (23.65.3)$$

The relationship between heat capacity (specific heat) and temperature may be characterized by a cubic polynomial equation as follows:

$$C_p = G_1 + G_2 T + G_3 T^2 + G_4 T^3. \quad (23.65.4)$$

A fully viscoplastic formulation is optional. An additional cost is incurred but the improvement in results can be dramatic.

23.66 Material Model 66: Linear Stiffness/Linear Viscous 3D Discrete Beam

The formulation of the discrete beam (Type 6) assumes that the beam is of zero length and requires no orientation node. A small distance between the nodes joined by the beam is permitted. The local coordinate system which determines (r, s, t) is given by the coordinate ID in the cross sectional input where the global system is the default. The local coordinate system axes rotate with the average of the rotations of the two nodes that define the beam.

For null stiffness coefficients, no forces corresponding to these null values will develop. The viscous damping coefficients are optional.

23.67 Material Model 67: Nonlinear Stiffness/Viscous 3D Discrete Beam

The formulation of the discrete beam (Type 6) assumes that the beam is of zero length and requires no orientation node. A small distance between the nodes joined by the beam is permitted. The local coordinate system which determines (r, s, t) is given by the coordinate ID in the cross sectional input where the global system is the default. The local coordinate system axes rotate with the average of the rotations of the two nodes that define the beam.

For null load curve ID's, no forces are computed. The force resultants are found from load curves (See [Figure 23.67.1](#)) that are defined in terms of the force resultant versus the relative displacement in the local coordinate system for the discrete beam.

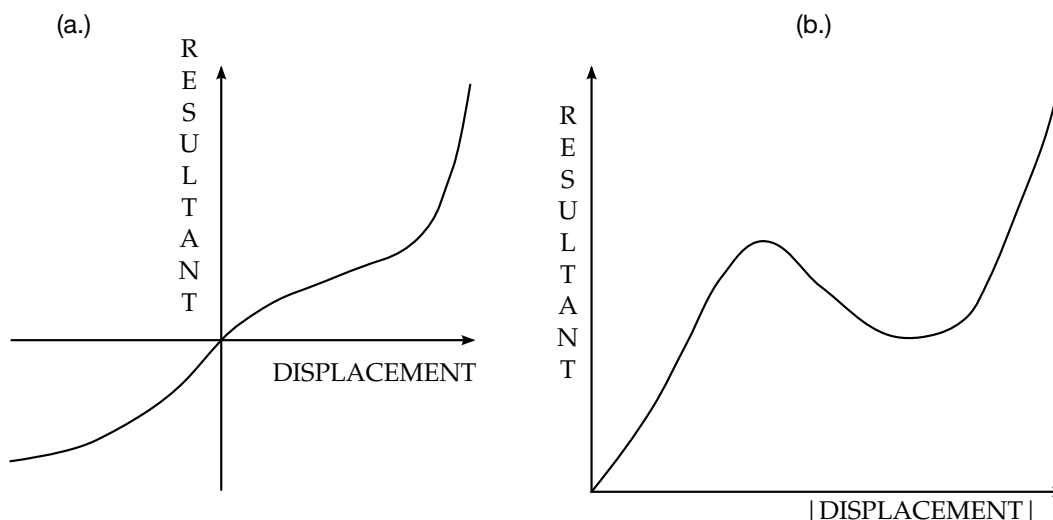


Figure 23.67.1. The resultant forces and moments are determined by a table lookup. If the origin of the load curve is at [0,0] as in (b.) and tension and compression responses are symmetric.

23.68 Material Model 68: Nonlinear Plastic/Linear Viscous 3D Discrete Beam

The formulation of the discrete beam (Type 6) assumes that the beam is of zero length and requires no orientation node. A small distance between the nodes joined by the beam is permitted. The local coordinate system which determines (r, s, t) is given by the coordinate ID in the cross sectional input where the global system is the default. The local coordinate system axes rotate with the average of the rotations of the two nodes that define the beam. Each force resultant in the local system can have a limiting value defined as a function of plastic displacement by using a load curve (See [Figure 23.68.1](#)). For the degrees of freedom where elastic behavior is desired, the load curve ID is simply set to zero.

Catastrophic failure, based on force resultants, occurs if the following inequality is satisfied:

$$\left(\frac{F_r}{F_r^{\text{fail}}}\right)^2 + \left(\frac{F_s}{F_s^{\text{fail}}}\right)^2 + \left(\frac{F_t}{F_t^{\text{fail}}}\right)^2 + \left(\frac{M_r}{M_r^{\text{fail}}}\right)^2 + \left(\frac{M_s}{M_s^{\text{fail}}}\right)^2 + \left(\frac{M_t}{M_t^{\text{fail}}}\right)^2 - 1. \geq 0. \quad (23.68.1)$$

Likewise, catastrophic failure based on displacement resultants occurs if:

$$\left(\frac{u_r}{u_r^{\text{fail}}}\right)^2 + \left(\frac{u_s}{u_s^{\text{fail}}}\right)^2 + \left(\frac{u_t}{u_t^{\text{fail}}}\right)^2 + \left(\frac{\theta_r}{\theta_r^{\text{fail}}}\right)^2 + \left(\frac{\theta_s}{\theta_s^{\text{fail}}}\right)^2 + \left(\frac{\theta_t}{\theta_t^{\text{fail}}}\right)^2 - 1. \geq 0. \quad (23.68.2)$$

After failure, the discrete element is deleted. If failure is included, either one or both of the criteria may be used.

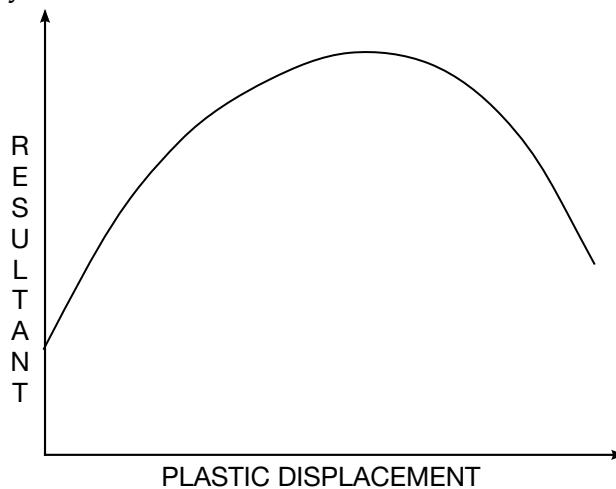


Figure 23.68.1. The resultant forces and moments are limited by the yield definition. The initial yield point corresponds to a plastic displacement of zero.

23.69 Material Model 69: Side Impact Dummy Damper (SID Damper)

The side impact dummy uses a damper that is not adequately treated by nonlinear force versus relative velocity curves, since the force characteristics are also dependent on the displacement of the piston. As the damper moves, the fluid flows through the open orifices to provide the necessary damping resistance. While moving as shown in [Figure 23.69.1](#), the piston gradually blocks off and effectively closes the orifices. The number of orifices and the size of their openings control the damper resistance and performance. The damping force is computed from the equation:

$$F = SF \left\{ KA_p V_p \left\{ \frac{C_1}{A_0^t} + C_2 |\mathbf{V}_p| \rho_{\text{fluid}} \left[\left(\frac{A_p}{CA_0^t} \right)^2 - 1 \right] \right\} - f(s + s_0) + V_p g(s + s_0) \right\}, \quad (23.69.1)$$

where K is a user defined constant or a tabulated function of the absolute value of the relative velocity, \mathbf{V}_p is the piston's relative velocity, C is the discharge coefficient, A_p is the piston area, A_0^t is the total open areas of orifices at time t , ρ_{fluid} is the fluid density, C_1 is the coefficient for the linear term, and C_2 is the coefficient for the quadratic term.

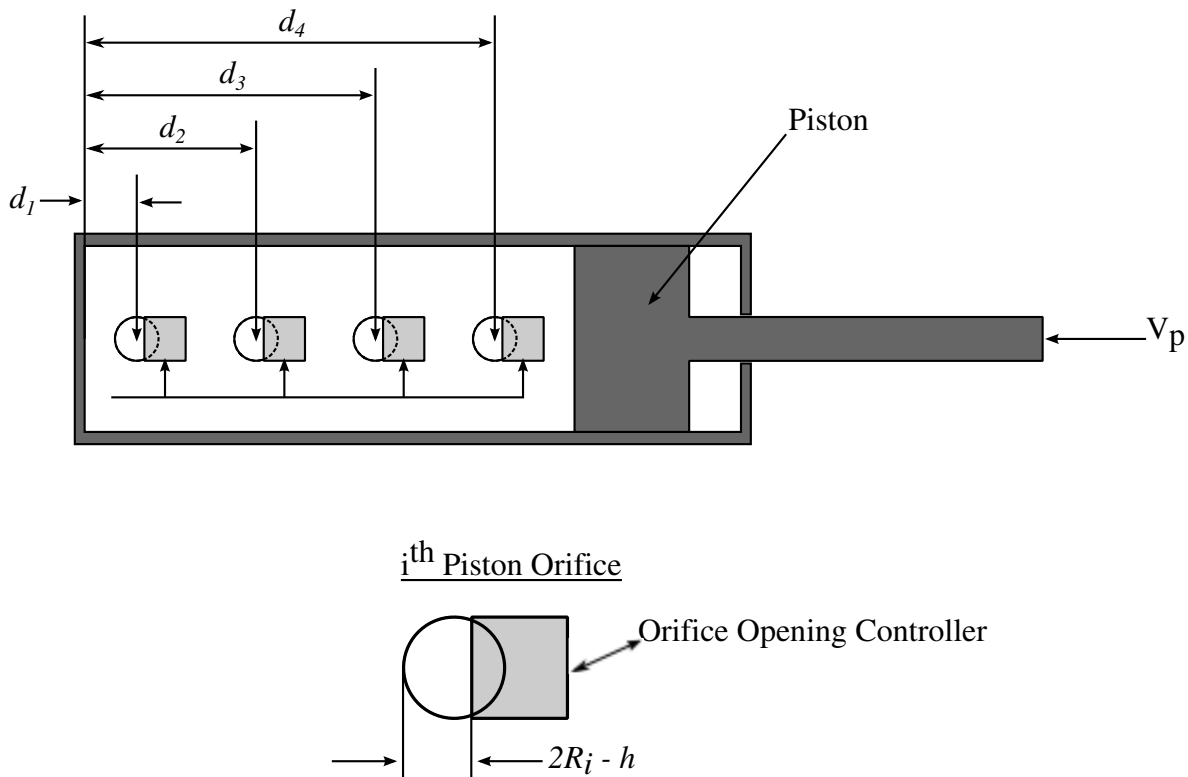


Figure 23.69.1. Mathematical model for the Side Impact Dummy damper.

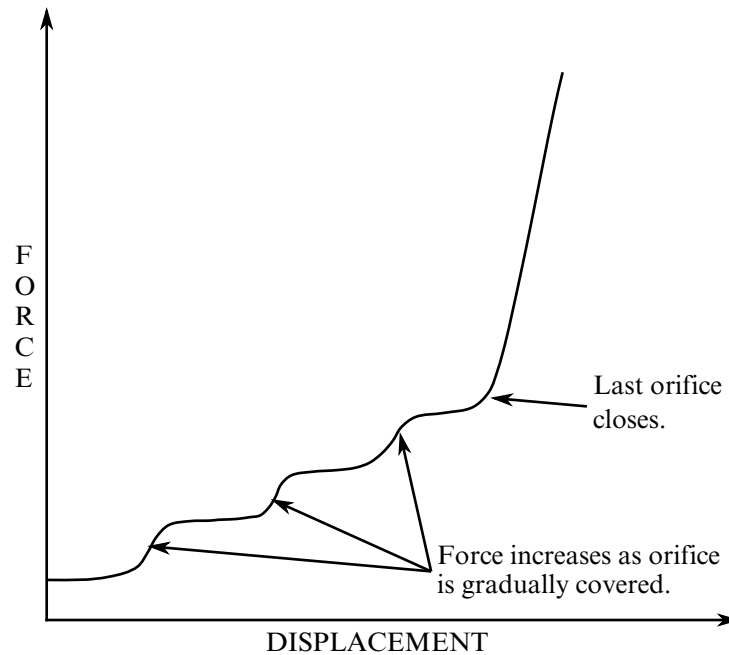


Figure 23.69.2. Force versus displacement as orifices are covered at a constant relative velocity. Only the linear velocity term is active.

In the implementation, the orifices are assumed to be circular with partial covering by the orifice controller. As the piston closes, the closure of the orifice is gradual. This gradual closure is taken into account to insure a smooth response. If the piston stroke is exceeded, the stiffness value, k , limits further movement, i.e., if the damper bottoms out in tension or compression, the damper forces are calculated by replacing the damper by a bottoming out spring and damper, k and c , respectively. The piston stroke must exceed the initial length of the beam element. The time step calculation is based in part on the stiffness value of the bottoming out spring. A typical force versus displacement curve at constant relative velocity is shown in [Figure 23.69.2](#). The factor, SF , which scales the force defaults to 1.0 and is analogous to the adjusting ring on the damper.

23.70 Material Model 70: Hydraulic/Gas Damper

This special purpose element represents a combined hydraulic and gas-filled damper which has a variable orifice coefficient. A schematic of the damper is shown in [Figure 23.70.1](#). Dampers of this type are sometimes used on buffers at the end of railroad tracks and as aircraft undercarriage shock absorbers. This material can be used only as a discrete beam element.

As the damper is compressed two actions contribute to the force that develops. First, the gas is adiabatically compressed into a smaller volume. Secondly, oil is forced through an orifice. A profiled pin may occupy some of the cross-sectional area of the orifice; thus, the orifice area available for the oil varies with the stroke. The force is assumed proportional to the square of the velocity and inversely proportional to the available area. The equation for this element is:

$$F = SCLF \cdot \left\{ K_h \left(\frac{V}{a_0} \right)^2 + \left[P_0 \left(\frac{C_0}{C_0 - S} \right)^n - P_a \right] \cdot A_p \right\}, \quad (23.70.1)$$

where S is the element deflection and V is the relative velocity across the element.

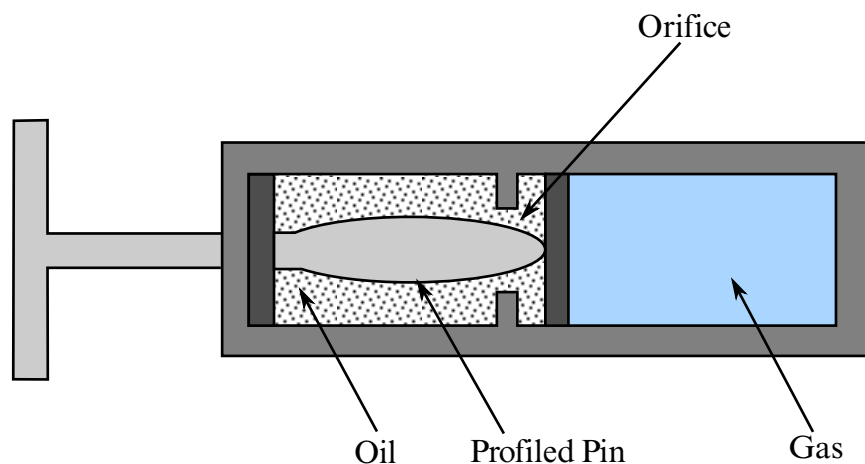


Figure 23.70.1. Schematic of Hydraulic/Gas damper.

23.71 Material Model 71: Cable

This material can be used only as a discrete beam element. The force, F , generated by the cable is nonzero only if the cable is in tension. The force is given by:

$$F = K \cdot \max(\Delta L, 0.), \quad (23.71.1)$$

where ΔL is the change in length

$$\Delta L = \text{current length} - (\text{initial length} - \text{offset}), \quad (23.71.2)$$

and the stiffness is defined as:

$$K = \frac{E \cdot \text{area}}{(\text{initial length} - \text{offset})}. \quad (23.71.3)$$

The area and offset are defined on either the cross section or element cards in the LS-DYNA input. For a slack cable the offset should be input as a negative length. For an initial tensile force the offset should be positive. If a load curve is specified, the Young's modulus will be ignored and the load curve will be used instead. The points on the load curve are defined as engineering stress versus engineering strain, i.e., the change in length over the initial length. The unloading behavior follows the loading.

23.73 Material Model 73: Low Density Viscoelastic Foam

This viscoelastic foam model is available to model highly compressible viscous foams. The hyperelastic formulation of this model follows that of material 57. Rate effects are accounted for through linear viscoelasticity by a convolution integral of the form

$$\sigma_{ij}^r = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.73.1)$$

where $g_{ijkl}(t - \tau)$ is the relaxation function. The stress tensor, σ_{ij}^r , augments the stresses determined from the foam, σ_{ij}^f ; consequently, the final stress, σ_{ij} , is taken as the summation of the two contributions:

$$\sigma_{ij} = \sigma_{ij}^f + \sigma_{ij}^r. \quad (23.73.2)$$

Since we wish to include only simple rate effects, the relaxation function is represented by up to six terms of the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^N \alpha_m e^{-\beta_m t}. \quad (23.73.3)$$

This model is effectively a Maxwell fluid which consists of a dampers and springs in series. The formulation is performed in the local system of principal stretches where only the principal values of stress are computed and triaxial coupling is avoided. Consequently, the one-dimensional nature of this foam material is unaffected by this addition of rate effects. The addition of rate effects necessitates 42 additional history variables per integration point. The cost and memory overhead of this model comes primarily from the need to “remember” the local system of principal stretches and the evaluation of the viscous stress components. Viscous damping in the model follows an implementation identical to that of material type 57.

23.74 Material Model 74: Elastic Spring for the Discrete Beam

This model permits elastic springs with damping to be combined and represented with a discrete beam element type 6. Linear stiffness and damping coefficients can be defined, and, for nonlinear behavior, a force versus deflection and force versus rate curves can be used. Displacement based failure and an initial force are optional.

If the linear spring stiffness is used, the force, F , is given by:

$$F = F_0 + K\Delta L + D\dot{\Delta L}, \quad (23.74.1)$$

where K is the stiffness constant, and D is the viscous damping coefficient.

If the load curve ID for $f(\Delta L)$ is specified, nonlinear behavior is activated. For this case the force is given by:

$$F = F_0 + K f(\Delta L) \left[1 + C1 \cdot \dot{\Delta L} + C2 \cdot \text{sgn}(\dot{\Delta L}) \ln \left(\max \left\{ 1, \frac{|\dot{\Delta L}|}{DLE} \right\} \right) \right] + D\dot{\Delta L} + g(\Delta L)h(\dot{\Delta L}), \quad (23.74.2)$$

where $C1$ and $C2$ are damping coefficients for nonlinear behavior, DLE is a factor to scale time units, and $g(\Delta L)$ is an optional load curve defining a scale factor versus deflection for load curve ID, $h(d\Delta L/dt)$.

In these equations, ΔL is the change in length

$$\Delta L = \text{current length} - \text{initial length}. \quad (23.74.3)$$

Failure can occur in either compression or tension based on displacement values of CDF and TDF, respectively. After failure no forces are carried. Compressive failure does not apply if the spring is initially zero length.

The cross sectional area is defined on the section card for the discrete beam elements, in *SECTION_BEAM. The square root of this area is used as the contact thickness offset if these elements are included in the contact treatment.

23.75 Material Model 75: Bilkhu/Dubois Foam Model

This model uses uniaxial and triaxial test data to provide a more realistic treatment of crushable foam. The Poisson's ratio is set to zero for the elastic response. The volumetric strain is defined in terms of the relative volume, V , as:

$$\gamma = -\ln(V). \quad (23.75.1)$$

In defining the curves, the stress and strain pairs should be positive values starting with a volumetric strain value of zero.

Viscous damping in the model follows an implementation identical to that of material type 57.

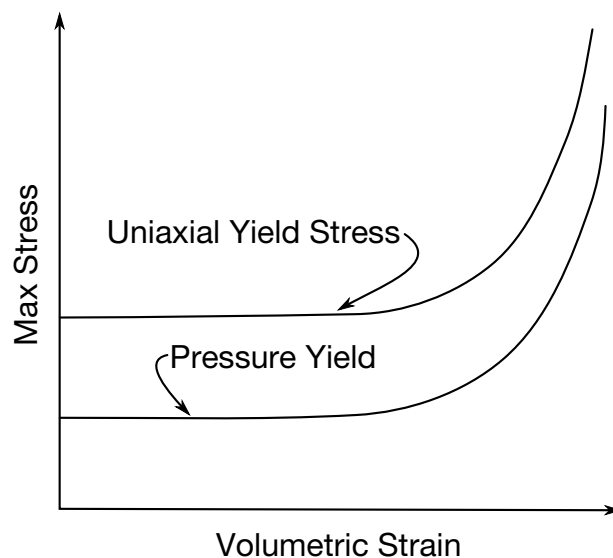


Figure 23.75.1. Behavior of crushable foam. Unloading is elastic.

23.76 Material Model 76: General Viscoelastic

23.76.1 Introduction

Material type 76 in LS-DYNA is a general viscoelastic Maxwell model having up to 18 terms in the prony series expansion and is useful for modeling dense continuum rubbers and solid explosives. It is characterized in the input by bulk and shear moduli, K_m and G_m , and associated decay constants, β_m^k and β_m^g . Either the coefficients of the prony series expansion can be used directly, or a relaxation curve may be specified to define the viscoelastic deviatoric and bulk behavior.

23.76.2 Constitutive Model

The model is a hypoelastic version of the model given by Christensen and can be stated as

$$\sigma^\nabla = \sum_m (K_m t_m^\nabla \mathbf{i} + 2G_m s_m^\nabla), \quad (23.76.1)$$

where t_m and s_m are (strain) quantities governed by the following evolution in time

$$s_m^\nabla = D_{\text{dev}} - \beta_m^g s_m \quad (23.76.2)$$

and

$$t_m^\nabla = D_{\text{vol}} - \beta_m^k t_m \quad (23.76.3)$$

Here K_m and G_m are bulk and shear moduli, respectively, β_m^k and β_m^g are the corresponding decay coefficients, D_{vol} and D_{dev} are the volumetric and deviatoric strain rates, \mathbf{i} is the 2nd order identity tensor and ∇ denotes the Jaumann objective rate. It should immediately be noted that for all decay coefficients equal to 0 (zero), the model is reduced to a time independent elastic model,

$$\sigma^\nabla = K D_{\text{vol}} \mathbf{i} + 2G D_{\text{dev}} \quad (23.76.4)$$

with bulk and shear modulus given by $K = \sum_m K_m$ and $G = \sum_m G_m$. For small displacement theory, the stress can be integrated to be given by

$$\sigma(t) = \sum_m \left(K_m \int_0^t e^{-\beta_m^k(t-\tau)} D_{\text{vol}}(\tau) d\tau \mathbf{i} + 2G_m \int_0^t e^{-\beta_m^g(t-\tau)} D_{\text{dev}}(\tau) d\tau \right) \quad (23.76.5)$$

For shell elements, the same theory applies except that the objective rate ∇ is the corotational time derivative instead of the Jaumann rate.

23.76.3 Tangent Modulus

For the implicit tangent modulus, we note that the internal force contribution depends on the displacement and time, which we denote $f_{\text{int}} = f_{\text{int}}(\mathbf{u}, t)$. The time derivative of this vector is the sum of a material and a geometric contribution. The material contribution is given by

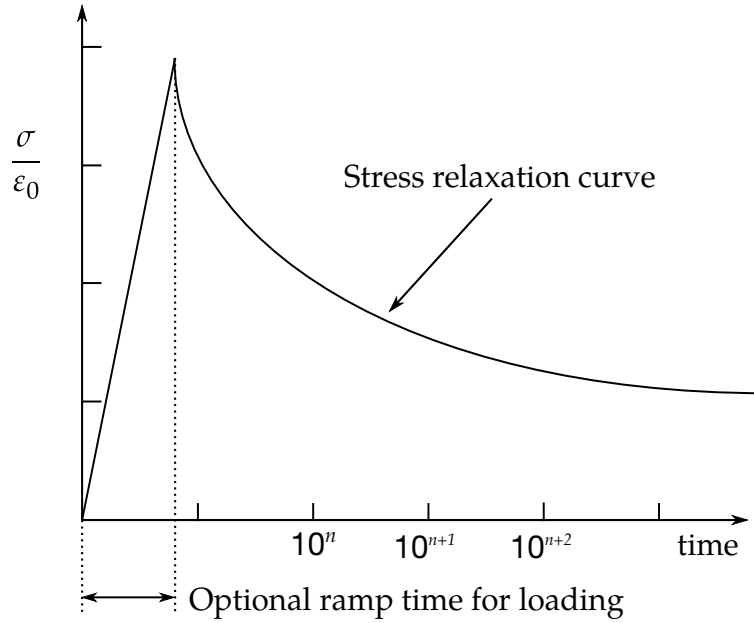


Figure 23.76.1. Relaxation curve. This curve defines stress versus time where time is defined on a logarithmic scale. For best results, the points defined in the load curve should be equally spaced on the logarithmic scale. Furthermore, the load curve should be smooth and defined in the positive quadrant. If nonphysical values are determined by least squares fit, LS-DYNA will terminate with an error message after the initialization phase is completed. If the ramp time for loading is included, then the relaxation which occurs during the loading phase is taken into account. This effect may or may not be important.

$$\dot{f}_{\text{int}}^{\text{mat}} = \int \mathbf{B}^T \sigma^{\nabla T} d\Omega \quad (23.76.6)$$

where \mathbf{B} is the strain displacement matrix, the integration is over the current configuration Ω and ∇T stands for Truesdell rate. This expression should later be identified with

$$\dot{f}_{\text{int}}^{\text{mat}} = \frac{\partial f_{\text{int}}^{\text{mat}}}{\partial \mathbf{u}} \dot{\mathbf{u}} + \frac{\partial f_{\text{int}}^{\text{mat}}}{\partial t} \quad (23.76.7)$$

in order to determine the tangent modulus. Neglecting discrepancies between the Jaumann and Truesdell rates, we can use (23.76.2) and (23.76.3) in (23.76.6) to get

$$\dot{f}_{\text{int}}^{\text{mat}} = \int \mathbf{B}^T \sum_m (K_m \mathbf{i} \otimes \mathbf{i} + 2G_m \mathbf{I}_{\text{dev}}) \mathbf{B} d\Omega \dot{\mathbf{u}} - \int \mathbf{B}^T \sum_m (K_m \beta_m^k t_m \mathbf{i} + 2G_m \beta_m^s s_m) d\Omega, \quad (23.76.8)$$

where \mathbf{I}_{dev} is the 4th order deviatoric identity tensor. Comparing this expression with (23.76.7) one can conclude that

$$\frac{\partial f_{\text{int}}^{\text{mat}}}{\partial \mathbf{u}} = \int \mathbf{B}^T \sum_m (K_m \mathbf{i} \otimes \mathbf{i} + 2G_m \mathbf{I}_{\text{dev}}) \mathbf{B} d\Omega, \quad (23.76.9)$$

and hence the tangent modulus is

$$\mathbf{C} = \sum_m (K_m \mathbf{i} \otimes \mathbf{i} + 2G_m \mathbf{I}_{\text{dev}}). \quad (23.76.10)$$

i.e., independent of deformation and time. In fact, this tangent modulus is equal to the classical elastic tangent modulus for a hypoelastic material, cf. (23.76.4).

23.76.4 Using the Relaxation Curve

Instead of inputting the stiffness and relaxation parameters, one can input a relaxation curve from test according to Figure 23.76.1. The time scale is determined by BSTART and LS-DYNA will determine all parameters as to best fit the curve.

23.77 Material Model 77: Hyperviscoelastic Rubber

Material type 77 in LS-DYNA consists of two hyperelastic rubber models, a general hyperelastic rubber model and an Ogden rubber model, that can be combined optionally with a viscoelastic stress contribution. As for the rate independent part, the constitutive law is determined by a strain energy function which in this case advantageously can be expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij} , as well as the constitutive tensor of interest, D_{ijkl}^{TC} , they are first calculated in the principal basis after which they are transformed back to the “base frame”, or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchhoff stress components are given by

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} \quad (\text{no sum}), \quad (23.77.1)$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau_{kl}^E. \quad (23.77.2)$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1} \tau_{ij}, \quad (23.77.3)$$

where $J = \lambda_1 \lambda_2 \lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchhoff stress can in the principal basis be expressed as

$$\begin{aligned} D_{ijj}^{TKE} &= \lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} - 2\tau_{ii}^E \delta_{ij} \\ D_{ijj}^{TKE} &= \frac{\lambda_j^2 \tau_{ii}^E - \lambda_i^2 \tau_{jj}^E}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \quad (\text{no sum}). \\ D_{ijj}^{TKE} &= \frac{\lambda_i}{2} \left(\frac{\partial \tau_{ii}^E}{\partial \lambda_i} - \frac{\partial \tau_{ii}^E}{\partial \lambda_j} \right), \quad i \neq j, \lambda_i = \lambda_j \end{aligned} \quad (23.77.4)$$

These components are transformed to the standard basis according to

$$D_{ijkl}^{TK} = q_{ip} q_{jq} q_{kr} q_{ls} D_{pqrs}^{TKE}, \quad (23.77.5)$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through

$$D_{ijkl}^{TC} = J^{-1} D_{ijkl}^{TK}. \quad (23.77.6)$$

When dealing with shell elements, the tangent moduli in the corotational coordinates is of interest. This matrix is given by

$$\widehat{D}_{ijkl}^{TC} = R_{pi} R_{qj} R_{rk} R_{sl} D_{pqrs}^{TC} = J^{-1} R_{pi} R_{qj} R_{rk} R_{sl} D_{pqrs}^{TK} = J^{-1} \hat{q}_{ip} \hat{q}_{jq} \hat{q}_{kr} \hat{q}_{ls} D_{pqrs}^{TKE}, \quad (23.77.7)$$

where R_{ij} is the matrix containing the unit basis vectors of the corotational system and $\hat{q}_{ij} = R_{ki} q_{kj}$. The latter matrix can be determined as the eigenvectors of the co-rotated left Cauchy-Green tensor (or the left stretch tensor). In LS-DYNA, the tangent stiffness matrix is after assembly transformed back to the standard basis according to standard transformation formulae.

23.77.1 General Hyperelastic Rubber Model

The strain energy function for the general hyperelastic rubber model is given by

$$W = \sum_{p,q=0}^n C_{pq} W_1^p W_2^q + \frac{1}{2} K (J - 1)^2, \quad (23.77.8)$$

where K is the bulk modulus,

$$\begin{aligned} W_1 &= I_1 I_3^{-\frac{1}{3}} - 3 \\ W_2 &= I_2 I_3^{-2/3} - 3, \end{aligned} \quad (23.77.9)$$

and

$$\begin{aligned} I_1 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_1^2 \lambda_3^2 \\ I_3 &= \lambda_1^2 \lambda_2^2 \lambda_3^2, \end{aligned} \quad (23.77.10)$$

are the invariants in terms of the principal stretches. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} = \sum_{p,q=0}^n C_{pq} (p W_1^{p-1} W'_{1i} W_2^q + q W_1^p W_2^{q-1} W'_{2i}) + KJ(J - 1), \quad (23.77.11)$$

where

$$\begin{aligned} W'_{1i} &:= \lambda_i \frac{\partial W_1}{\partial \lambda_i} = \left(2\lambda_i^2 - \frac{2}{3} I_1 \right) I_3^{-\frac{1}{3}} \\ W'_{2i} &:= \lambda_i \frac{\partial W_2}{\partial \lambda_i} = \left(2\lambda_i^2 (I_1 - \lambda_i^2) - \frac{4}{3} I_2 \right) I_3^{-\frac{2}{3}}. \end{aligned} \quad (23.77.12)$$

If C_{pq} is nonzero only for $pq = 01, 10, 11, 20, 02, 30$, then [Equation \(23.77.11\)](#) can be written as

$$\tau_{ii}^E = (C_{10} + C_{11}W_2 + 2C_{20}W_1 + 3C_{30}W_1^2)W'_{1i} + (C_{01} + C_{11}W_1 + 2C_{02}W_2)W'_{2i} + KJ(J - 1). \quad (23.77.13)$$

Proceeding with the constitutive tensor, we have

$$\begin{aligned} \lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = & \sum_{p,q=0}^n C_{pq} (p(p-1)W_1^{p-2}W'_{1i}W'_{1j}W_2^q + pW_1^{p-1}W''_{1ij}W_2^q + pqW_1^{p-1}W'_{1i}W_2^{q-1}W'_{2j} \\ & + qpW_1^{p-1}W'_{1j}W_2^{q-1}W'_{2i} + q(q-1)W_1^pW_2^{q-2}W'_{2i}W'_{2j} + qW_1^pW_2^{q-1}W''_{2ij}) \\ & + KJ(2J - 1), \end{aligned} \quad (23.77.14)$$

where

$$\begin{aligned} W''_{1ij} &:= \lambda_j \frac{\partial W'_{1i}}{\partial \lambda_j} = (4\lambda_i^2\delta_{ij} - \frac{4}{3}(\lambda_i^2 + \lambda_j^2) + \frac{4}{9}I_1)I_3^{-1/3} \\ W''_{2ij} &:= \lambda_j \frac{\partial W'_{2i}}{\partial \lambda_j} = ((4\lambda_i^2I_1 - 8\lambda_i^4)\delta_{ij} + 4\lambda_i^2\lambda_j^2 - \frac{8}{3}(\lambda_i^2(I_1 - \lambda_i^2) + \lambda_j^2(I_1 - \lambda_j^2)) + \frac{16}{9}I_2)I_3^{-2/3} \end{aligned} \quad (23.77.15)$$

Again, using only the nonzero coefficients mentioned above, Equation (23.77.14) is reduced to

$$\begin{aligned} \lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = & C_{11}(W'_{1i}W'_{2j} + W'_{1j}W'_{2i}) + 2(C_{20} + 3C_{30}W_1)W'_{1j}W'_{1i} + 2C_{02}W'_{2i}W'_{2j} + \\ & (C_{10} + C_{11}W_2 + 2C_{20}W_1 + 3C_{30}W_1^2)W''_{1ij} + (C_{01} + C_{11}W_1 + 2C_{02}W_2)W''_{2ij} + \\ & KJ(2J - 1). \end{aligned} \quad (23.77.16)$$

23.77.2 Ogden Rubber Model

The strain energy function for the Ogden rubber model is given by

$$W = \sum_{m=1}^n \frac{\mu_m}{\alpha_m} (\tilde{\lambda}_1^{\alpha_m} + \tilde{\lambda}_2^{\alpha_m} + \tilde{\lambda}_3^{\alpha_m} - 3) + K(J - 1 - \ln J), \quad (23.77.17)$$

where

$$\tilde{\lambda}_i = \frac{\lambda_i}{J^{1/3}}, \quad (23.77.18)$$

are the volumetric independent principal stretches, and μ_m and α_m are material parameters. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} = \sum_{m=1}^n \mu_m (\tilde{\lambda}_i^{\alpha_m} - \frac{1}{3}a_m) + K(J - 1), \quad (23.77.19)$$

where

$$a_m = \tilde{\lambda}_1^{\alpha_m} + \tilde{\lambda}_2^{\alpha_m} + \tilde{\lambda}_3^{\alpha_m}. \quad (23.77.20)$$

Proceeding with the constitutive tensor, we have

$$\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = \sum_{m=1}^n \frac{\mu_m \alpha_m}{3} \left(\frac{1}{3} a_m + 3 \tilde{\lambda}_i^{\alpha_m} \delta_{ij} - \tilde{\lambda}_i^{\alpha_m} - \tilde{\lambda}_j^{\alpha_m} \right) + KJ. \quad (23.77.21)$$

23.77.3 The Viscoelastic Contribution

As mentioned above, this material model is accompanied with a viscoelastic stress contribution. The rate form of this constitutive law can in co-rotational coordinates be written

$$\dot{\hat{\sigma}}_{ij}^{\text{ve}} = \sum_{m=1}^n 2G_m \hat{D}_{ij}^{\text{dev}} - \sum_{m=1}^n 2\beta_m G_m \int_0^t e^{-\beta_m(t-\tau)} \hat{D}_{ij}^{\text{dev}}(\tau) d\tau. \quad (23.77.22)$$

Here n is a number less than or equal to 6, $\hat{\sigma}_{ij}^{\text{ve}}$ is the co-rotated viscoelastic stress, $\hat{D}_{ij}^{\text{dev}}$ is the deviatoric co-rotated rate-of-deformation and G_m and β_m are material parameters. The parameters G_m can be thought of as shear moduli and β_m as decay coefficients determining the relaxation properties of the material. This rate form can be integrated in time to form the corotated viscoelastic stress

$$\hat{\sigma}_{ij}^{\text{ve}} = \sum_{m=1}^6 2G_m \int_0^t e^{-\beta_m(t-\tau)} \hat{D}_{ij}^{\text{dev}}(\tau) d\tau. \quad (23.77.23)$$

For the constitutive matrix, we refer to Borrvall [2002] and here simply state that it is equal to

$$\hat{D}_{ijkl}^{\text{TCve}} = \sum_{m=1}^n 2G_m \left(\frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) - \frac{1}{3} \delta_{ij} \delta_{kl} \right). \quad (23.77.24)$$

23.77.4 Stress Update for Shell Elements

In principal, the stress update for material 77 and shell elements follows closely the one that is implemented for material 27. The stress is evaluated in corotational coordinates after which it is transformed back to the standard basis according to

$$\sigma_{ij} = R_{ik} R_{jl} \hat{\sigma}_{kl}, \quad (23.77.25)$$

or equivalently the internal force is assembled in the corotational system and then transformed back to the standard basis according to standard transformation formulae. Here

R_{ij} is the rotation matrix containing the corotational basis vectors. The so-called corotated stress $\hat{\sigma}_{ij}$ is evaluated as the sum of the stresses given in Sections 19.77.1 and 19.77.4.

The viscoelastic stress contribution is incrementally updated with aid of the corotated rate of deformation. To be somewhat more precise, the values of the 12 integrals in Equation (23.77.23) are kept as history variables that are updated in each time step. Each integral is discretized in time and the mean value theorem is used in each time step to determine their values.

For the hyperelastic stress contribution, the principal stretches are needed and here taken as the square root of the eigenvalues of the co-rotated left Cauchy-Green tensor \hat{b}_{ij} . The corotated left Cauchy-Green tensor is incrementally updated with the aid of a time increment Δt , the corotated velocity gradient \hat{L}_{ij} , and the angular velocity $\hat{\Omega}_{ij}$ with which the embedded coordinate system is rotating,

$$\hat{b}_{ij} = \hat{b}_{ij} + \Delta t(\hat{L}_{ik} - \hat{\Omega}_{ik})\hat{b}_{kj} + \Delta t\hat{b}_{ik}(\hat{L}_{ik} - \hat{\Omega}_{ik}). \quad (23.77.26)$$

The primary reason for taking a corotational approach is to facilitate the maintenance of a vanishing normal stress through the thickness of the shell, something that is achieved by adjusting the corresponding component of the corotated velocity gradient \hat{L}_{33} accordingly. The problem can be stated as to determine \hat{L}_{33} such that when updating the left Cauchy-Green tensor through Equation (23.77.26) and subsequently the stress through formulae in Sections 19.77.1 and 19.77.4, $\hat{\sigma}_{33} = 0$. To this end, it is assumed that

$$\hat{L}_{33} = \alpha(\hat{L}_{11} + \hat{L}_{22}), \quad (23.77.27)$$

for some parameter α that is determined in the following three step procedure. In the first two steps, $\alpha = 0$ and $\alpha = -1$, respectively, resulting in two trial normal stresses $\hat{\sigma}_{33}^{(0)}$ and $\hat{\sigma}_{33}^{(-1)}$. Then it is assumed that the actual normal stress depends linearly on α , meaning that the latter can be determined from

$$0 = \sigma_{33}^{(\alpha)} = \sigma_{33}^{(0)} + \alpha(\sigma_{33}^{(0)} - \sigma_{33}^{(-1)}). \quad (23.77.28)$$

In the current implementation, α is given by

$$\alpha = \begin{cases} \frac{\hat{\sigma}_{33}^{(0)}}{\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}} & \text{if } |\hat{\sigma}_{33}^{(-1)} - \hat{\sigma}_{33}^{(0)}| \geq 10^{-4} \\ -1 & \text{otherwise} \end{cases} \quad (23.77.29)$$

and the stresses are determined from this value of α . Finally, to make sure that the normal stress through the thickness vanishes, it is set to 0 (zero) before exiting the stress update routine.

23.78 Material Model 78: Soil/Concrete

Concrete pressure is positive in compression. Volumetric strain is defined as the natural log of the relative volume and is positive in compression where the relative volume, V , is the ratio of the current volume to the initial volume. The tabulated data should be given in order of increasing compression. If the pressure drops below the cutoff value specified, it is reset to that value and the deviatoric stress state is eliminated.

If the load curve ID is provided as a positive number, the deviatoric perfectly plastic pressure dependent yield function ϕ , is described in terms of the second invariant, J_2 , the pressure, p , and the tabulated load curve, $F(p)$, as

$$\phi = \sqrt{3J_2} - F(p) = \sigma_y - F(p), \quad (23.78.1)$$

where J_2 is defined in terms of the deviatoric stress tensor as:

$$J_2 = \frac{1}{2} S_{ij} S_{ij}, \quad (23.78.2)$$

assuming that if the ID is given as negative, then the yield function becomes:

$$\phi = J_2 - F(p), \quad (23.78.3)$$

being the deviatoric stress tensor.

If cracking is invoked, the yield stress is multiplied by a factor f which reduces with plastic stain according to a trilinear law as shown in [Figure 23.78.1](#).

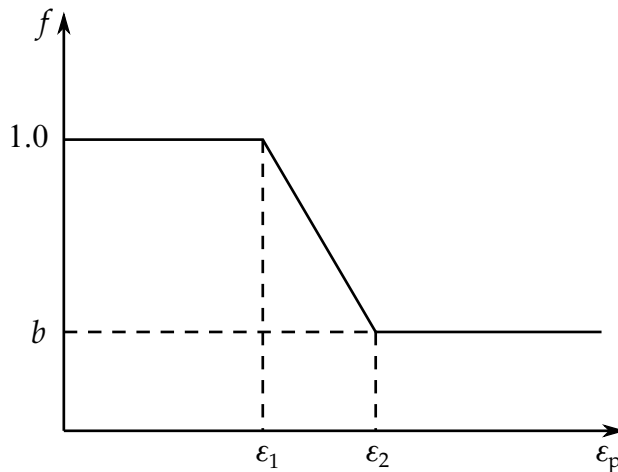


Figure 23.78.1. Strength reduction factor.

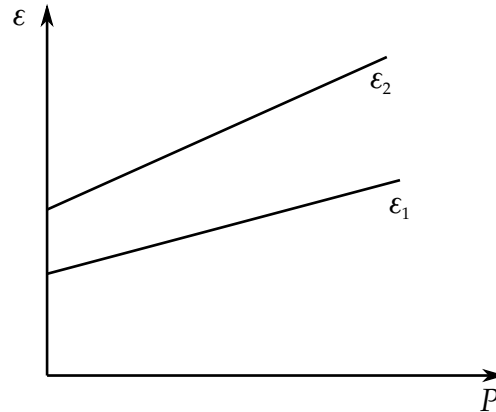


Figure 23.78.2. Cracking strain versus pressure.

b = residual strength factor

ε_1 = plastic stain at which cracking begins.

ε_2 = plastic stain at which residual strength is reached.

ε_1 and ε_2 are tabulated functions of pressure that are defined by load curves (see [Figure 23.78.2](#)). The values on the curves are pressure versus strain and should be entered in order of increasing pressure. The strain values should always increase monotonically with pressure.

By properly defining the load curves, it is possible to obtain the desired strength and ductility over a range of pressures. See [Figure 23.78.3](#).

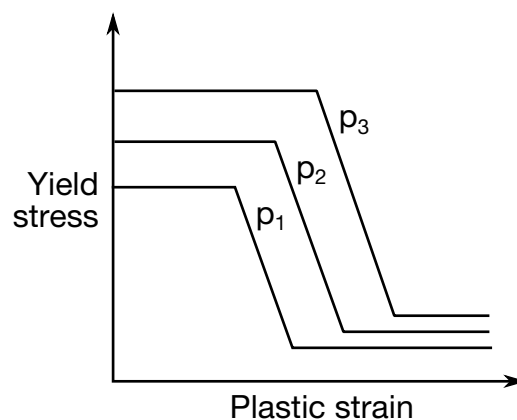


Figure 23.78.3. Example Caption

23.79 Material Model 79: Hysteretic Soil

This model is a nested surface model with up to ten superposed “layers” of elastic perfectly plastic material, each with its own elastic moduli and yield values. Nested surface models give hysteretic behavior, as the different “layers” yield at different stresses.

The constants a_0 , a_1 , and a_2 govern the pressure sensitivity of the yield stress. Only the ratios between these values are important - the absolute stress values are taken from the stress-strain curve.

The stress strain pairs $(\gamma_1, \tau_1), \dots (\gamma_N, \tau_N)$ define a shear stress versus shear strain curve. The first point on the curve is assumed by default to be (0,0) and does not need to be entered. The slope of the curve must decrease with increasing γ . Not all ten points need be to be defined. This curve applies at the reference pressure; at other pressures the curve varies according to a_0 , a_1 , and a_2 as in the soil and crushable foam model, Material 5.

The elastic moduli G and K are pressure sensitive.

$$\begin{aligned} G &= G_0(p - p_0)^b, \\ K &= K_0(p - p_0)^b, \end{aligned} \quad (23.79.1)$$

where K_0 is an input value, p is the current pressure, p_0 the cut-off or reference pressure (must be zero or negative), and G_0 is the initial shear modulus at small shear strain:

$$G_0 = \text{SFLC} \times \frac{\tau_1}{\gamma_1} \quad (23.79.2)$$

G_0 represents the total of the elastic slopes of all the layers.

If p attempts to fall below p_0 (i.e., more tensile) the shear stresses are set to zero and the pressure is set to p_0 . Thus, the material has no stiffness or strength in tension. The pressure in compression is calculated as follows:

$$p = [-K_0 \ln(V)]^{1/1-b}, \quad (23.79.3)$$

where V is the relative volume, i.e., the ratio between the original and current volume.

23.80 Material Model 80: Ramberg-Osgood Plasticity

The Ramberg-Osgood equation is an empirical constitutive relation to represent the one-dimensional elastic-plastic behavior of many materials, including soils. This model allows a simple rate independent representation of the hysteretic energy dissipation observed in soils subjected to cyclic shear deformation. For monotonic loading, the stress-strain relationship is given by:

$$\begin{aligned}\frac{\gamma}{\gamma_y} &= \frac{\tau}{\tau_y} + \alpha \left| \frac{\tau}{\tau_y} \right|^r & \text{if } \gamma \geq 0, \\ \frac{\gamma}{\gamma_y} &= \frac{\tau}{\tau_y} - \alpha \left| \frac{\tau}{\tau_y} \right|^r & \text{if } \gamma < 0,\end{aligned}\tag{23.80.1}$$

where γ is the shear and τ is the stress. The model approaches perfect plasticity as the stress exponent $r \rightarrow \infty$. These equations must be augmented to correctly model unloading and reloading material behavior. The first load reversal is detected by $\gamma\dot{\gamma} < 0$. After the first reversal, the stress-strain relationship is modified to

$$\begin{aligned}\frac{(\gamma - \gamma_0)}{2\gamma_y} &= \frac{(\tau - \tau_0)}{2\tau_y} + \alpha \left| \frac{(\tau - \tau_0)}{2\tau_y} \right|^r & \text{if } \gamma \geq 0, \\ \frac{(\gamma - \gamma_0)}{2\gamma_y} &= \frac{(\tau - \tau_0)}{2\tau_y} - \alpha \left| \frac{(\tau - \tau_0)}{2\tau_y} \right|^r & \text{if } \gamma < 0,\end{aligned}\tag{23.80.2}$$

where γ_0 and τ_0 represent the values of strain and stress at the point of load reversal. Subsequent load reversals are detected by $(\gamma - \gamma_0)\dot{\gamma} < 0$.

The Ramberg-Osgood equations are inherently one-dimensional and are assumed to apply to shear components. To generalize this theory to the multidimensional case, it is assumed that each component of the deviatoric stress and deviatoric tensorial strain is independently related by the one-dimensional stress-strain equations. A projection is used to map the result back into deviatoric stress space if required. The volumetric behavior is elastic, and, therefore, the pressure p is found by

$$p = -K\varepsilon_v,\tag{23.80.3}$$

where ε_v is the volumetric strain.

23.81 Material Models 81 and 82: Plasticity with Damage and Orthotropic Option

With this model an elasto-viscoplastic material with an arbitrary stress versus strain curve and arbitrary strain rate dependency can be defined. Damage is considered before rupture occurs. Also, failure based on a plastic strain or a minimum time step size can be defined.

An option in the keyword input, ORTHO, is available, which invokes an orthotropic damage model. This option is an extension to include orthotropic damage as a means of treating failure in aluminum panels. Directional damage begins after a defined failure strain is reached in tension and continues to evolve until a tensile rupture strain is reached in either one of the two orthogonal directions.

The stress versus strain behavior may be treated by a bilinear stress strain curve by defining the tangent modulus, ETAN. Alternately, a curve similar to that shown in [Figure 23.81.1](#) is expected to be defined by (EPS1,ES1) - (EPS8,ES8); however, an effective stress versus effective plastic strain curve (LCSS) may be input instead if eight points are insufficient. The cost is roughly the same for either approach. The most general approach

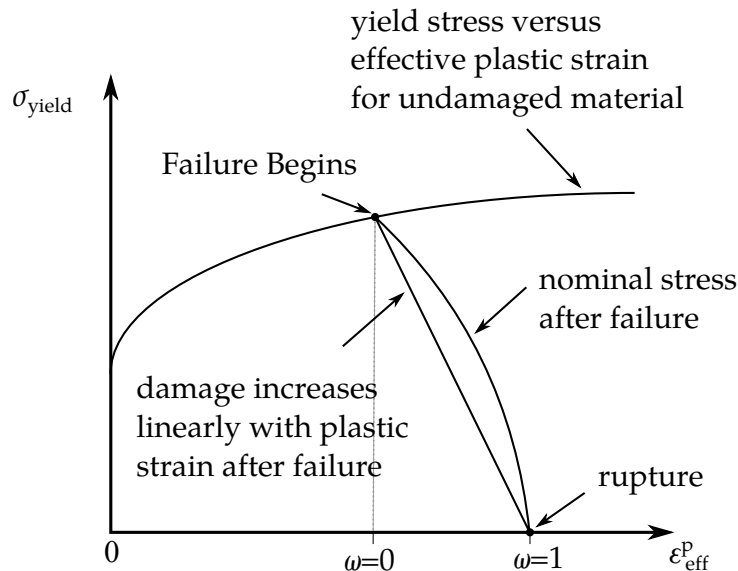


Figure .23.81.1. Stress strain behavior when damage is included.

is to use the table definition (LCSS) discussed below.

Two options to account for strain rate effects are possible. Strain rate may be accounted for using the Cowper-Symonds model which scales the yield stress with the factor,

$$1 + \left(\frac{\dot{\epsilon}}{C} \right)^{1/p}, \quad (23.81.1)$$

where $\dot{\epsilon}$ is the strain rate, $\dot{\epsilon} = \sqrt{\dot{\epsilon}_{ij}\dot{\epsilon}_{ij}}$. If the viscoplastic option is active, VP = 1.0, and if SIGY is > 0 then the dynamic yield stress is computed from the sum of the static stress, $\sigma_y^s(\epsilon_{\text{eff}}^P)$, which is typically given by a load curve ID, and the initial yield stress, SIGY, multiplied by the Cowper-Symonds rate term as follows:

$$\sigma_y(\epsilon_{\text{eff}}^P, \dot{\epsilon}_{\text{eff}}^P) = \sigma_y^s(\epsilon_{\text{eff}}^P) + \text{SIGY} \cdot \left(\frac{\dot{\epsilon}_{\text{eff}}^P}{C} \right)^{1/p}, \quad (23.81.2)$$

where the plastic strain rate is used. With this latter approach similar results can be obtained between this model and material model: *MAT_ANISOTROPIC_VISCOPLASTIC. If SIGY = 0, the following equation is used instead where the static stress, $\sigma_y^s(\epsilon_{\text{eff}}^P)$, must be defined by a load curve:

$$\sigma_y(\epsilon_{\text{eff}}^P, \dot{\epsilon}_{\text{eff}}^P) = \sigma_y^s(\epsilon_{\text{eff}}^P) \left[1 + \left(\frac{\dot{\epsilon}_{\text{eff}}^P}{C} \right)^{1/p} \right]. \quad (23.81.3)$$

This latter equation is always used if the viscoplastic option is off. For complete generality a load curve (LCSR) to scale the yield stress may be input instead. In this curve the scale factor versus strain rate is defined.

The constitutive properties for the damaged material are obtained from the undamaged material properties. The amount of damage evolved is represented by the constant, ω , which varies from zero if no damage has occurred to unity for complete rupture. For uniaxial loading, the nominal stress in the damaged material is given by

$$\sigma_{\text{nominal}} = \frac{P}{A}, \quad (23.81.4)$$

where P is the applied load and A is the surface area. The true stress is given by:

$$\sigma_{\text{true}} = \frac{P}{A - A_{\text{loss}}}, \quad (23.81.5)$$

where A_{loss} is the void area. The damage variable can then be defined:

$$\omega = \frac{A_{\text{loss}}}{A}, \quad 0 \leq \omega \leq 1. \quad (23.81.6)$$

In this model damage is defined in terms of plastic strain after the failure strain is exceeded:

$$\omega = \frac{\epsilon_{\text{eff}}^P - \epsilon_{\text{failure}}^P}{\epsilon_{\text{rupture}}^P - \epsilon_{\text{failure}}^P} \quad \text{if} \quad \epsilon_{\text{failure}}^P \leq \epsilon_{\text{eff}}^P \leq \epsilon_{\text{rupture}}^P. \quad (23.81.7)$$

After exceeding the failure strain softening begins and continues until the rupture strain is reached.

By default, deletion of a shell element occurs when all integration points in the shell have failed. A parameter is available, NUMINT, that defines the number of through thickness integration points for shell element deletion. The default of all integration points is not recommended since shells undergoing large strain are often not deleted due to nodal fiber rotations which limit strains at active integration points after most points have failed. Better results are obtained if NUMINT is set to 1 or a number less than one half of the number of through thickness points. For example, if four through thickness points are used, NUMINT should not exceed 2, even for fully integrated shells which have 16 integration points.

23.81.1 Material Model 82: Isotropic Elastic-Plastic with Anisotropic Damage

Material 82 is an isotropic elastic-plastic material model with anisotropic damage.. The stress update in the case of shell elements is performed as follows. For a given stress state σ_{ij}^t and damage parameters D_i^t , $i = 1, 2$ at time t , the local stress is obtained as

$$\sigma_{ij}^l = q_{ki}q_{lj}\sigma_{kl}^t, \quad (23.81.8)$$

where q_{ij} is an orthogonal matrix determining the direction of the damage. The directions are determined as follows. The first direction is taken as the one in which the plastic strain first reaches the plastic strain at impending failure, see below. The other direction is orthogonal to the first and in the plane of the shell.

For this local stress, the undamaged stress is computed as

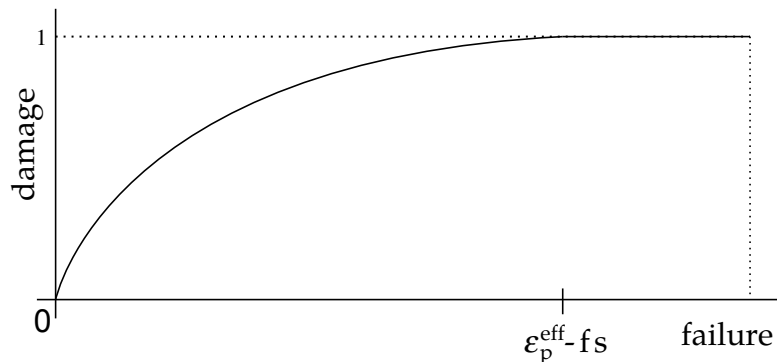


Figure 23.81.2. A nonlinear damage curve is optional. Note that the origin of the curve is at (0,0). It is permissible to input the failure strain, f_s , as zero for this option. The nonlinear damage curve is useful for controlling the softening behavior after the failure strain is reached.

(23.81.9)

$$\begin{aligned}\sigma_{22}^u &= \frac{\sigma_{22}^l}{1 - D_2^t}, \\ \sigma_{12}^u &= \frac{2\sigma_{12}^l}{2 - D_1^t - D_2^t}, \\ \sigma_{23}^u &= \frac{\sigma_{23}^l}{1 - D_2^t}, \\ \sigma_{13}^u &= \frac{\sigma_{13}^l}{1 - D_1^t}.\end{aligned}$$

A new undamaged stress σ_{ij}^{u+} is then computed following a standard elastic-plastic stress update. The damage at the next time step is computed according to

$$D_i^{t+} = \max\left(D_i^t, \frac{\varepsilon_{ii}^p - \varepsilon_f}{\varepsilon_r - \varepsilon_f}\right), \quad i = 1, 2, \quad (23.81.10)$$

where ε_f is the plastic strain at impending failure, ε_r is the plastic strain at rupture and ε_{ii}^p is the current plastic strain in the local i direction. There is also an option of defining a nonlinear damage curve, with this option the new damage is computed as

$$D_i^{t+} = \max\left(D_i^t, f(\varepsilon_{ii}^p - \varepsilon_f)\right), \quad i = 1, 2, \quad (23.81.11)$$

for a user-defined load curve f .

The new local (damaged) stress is given by

$$\begin{aligned}\sigma_{11}^{l+} &= \sigma_{11}^{u+}(1 - D_1^{t+}), \\ \sigma_{22}^{l+} &= \sigma_{22}^{u+}(1 - D_2^{t+}), \\ \sigma_{12}^{l+} &= \sigma_{12}^{u+} \frac{2 - D_1^{t+} - D_2^{t+}}{2}, \\ \sigma_{23}^{l+} &= \sigma_{23}^{u+}(1 - D_2^{t+}), \\ \sigma_{13}^{l+} &= \sigma_{13}^{u+}(1 - D_1^{t+}),\end{aligned} \quad (23.81.12)$$

which is transformed back to the local system to obtain the new global damaged stress as

$$\sigma_{ij}^{t+} = q_{ik}q_{jl}\sigma_{kl}^{l+}. \quad (23.81.13)$$

An integration point is completely failed, i.e., it is removed from the calculations, when $\max(D_1, D_2) > 0.999$. The element is removed from the model when a user specified number of integration points in that element have failed.

There are options of using visco-plasticity in the current model. The details of this part of the stress update is omitted here.

The Rc-Dc Damage Model

The Rc-Dc model is defined as the following, see the report on the Fundamental Study of Crack Initiation and Propagation [2003]. The damage D is updated as

$$D^{t+} = D^t + \omega_1 \omega_2 \Delta \varepsilon_p \quad (23.81.14)$$

where $\Delta \varepsilon_p$ is the plastic strain increment and

$$\begin{aligned} \omega_1 &= (1 + \gamma p)^{-\alpha}, \\ \omega_2 &= (2 - A_D)^\beta. \end{aligned} \quad (23.81.15)$$

Here p is the pressure, α , β and γ are material parameters and

$$A_D = \begin{cases} 1.9999 & \text{if } \max(S_1, S_2) \leq 0 \\ \min\left(\left|\frac{S_1}{S_2}\right|, \left|\frac{S_2}{S_1}\right|\right) & \text{otherwise} \end{cases}. \quad (23.81.16)$$

where S_1 and S_2 are the in-plane principal stress values. Fracture is initiated when the accumulation of damage is greater than a critical damage D_c given by

$$D_c = D_0(1 + b\|\nabla D\|^\lambda). \quad (23.81.17)$$

Here D_0 , b and λ are material parameters and ∇D is the spatial gradient of damage. We have added an option to use a non-local formulation with D as the non-local variable and a characteristic length l . More information on this can be found in the LS-DYNA Keyword User's Manual [Hallquist 2003]. With this option we compute D_c as,

$$D_c = D_0, \quad (23.81.18)$$

hence the parameters b and λ are not used. A fracture fraction given by

$$F = \frac{D - D_c}{D_s} \quad (23.81.19)$$

defines the degradations of the material by the Rc-Dc model. Here D_s is yet another parameter determined by the user. The stress update of material 82 is modified accordingly.

Upon entry the stress is divided by the factor $1 - F^t$ to account for the Rc-Dc damage. Before exiting the routine, the stress is multiplied by the new Rc-Fc (reversed) fracture fraction $1 - F^{t+}$. An integration point is considered failed when $\min(1 - D_1, 1 - D_2)(1 - F) < 0.001$.

23.83 Material Model 83: Fu-Chang's Foam With Rate Effects

This model allows rate effects to be modeled in low and medium density foams, see [Figure 23.83.1](#). Hysteretic unloading behavior in this model is a function of the rate sensitivity with the most rate sensitive foams providing the largest hysteresis and visa versa. The unified constitutive equations for foam materials by Fu-Chang [1995] provide the basis for this model. This implementation incorporates the coding in the reference in modified form to ensure reasonable computational efficiency. The mathematical description given below is excerpted from the reference.

The strain is divided into two parts: a linear part and a non-linear part of the strain

$$\mathbf{E}(t) = \mathbf{E}^L(t) + \mathbf{E}^N(t), \quad (23.83.1)$$

and the strain rate becomes

$$\dot{\mathbf{E}}(t) = \dot{\mathbf{E}}^L(t) + \dot{\mathbf{E}}^N(t). \quad (23.83.2)$$

$\dot{\mathbf{E}}^N$ is an expression for the past history of \mathbf{E}^N . A postulated constitutive equation may be written as:

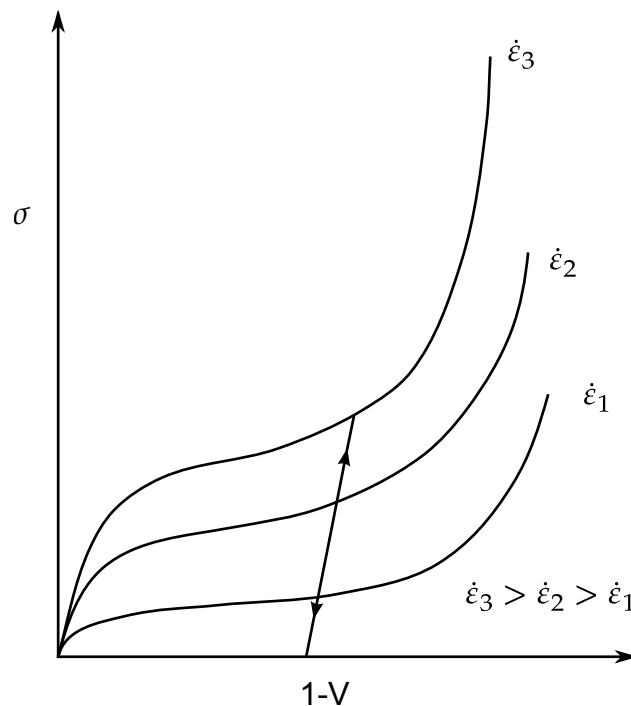


Figure 23.83.1. Rate effects in Fu Chang's foam model.

$$\sigma(t) = \int_{\tau=0}^{\infty} [\mathbf{E}_t^N(\tau), \mathbf{S}(t)] d\tau, \quad (23.83.3)$$

where $\mathbf{S}(t)$ is the state variable and $\int_{\tau=0}^{\infty}$ is a functional of all values of τ in $T_\tau: 0 \leq \tau \leq \infty$ and

$$\mathbf{E}_t^N(\tau) = \mathbf{E}^N(t - \tau), \quad (23.83.4)$$

where τ is the history parameter:

$$\mathbf{E}_t^N(\tau = \infty) \Leftrightarrow \text{the virgin material.} \quad (23.83.5)$$

It is assumed that the material remembers only its immediate past, i.e., a neighborhood about $\tau = 0$. Therefore, an expansion of $\mathbf{E}_t^N(\tau)$ in a Taylor series about $\tau = 0$ yields:

$$\mathbf{E}_t^N(\tau) = \mathbf{E}^N(0) + \frac{\partial \mathbf{E}_t^N}{\partial t}(0) d\tau. \quad (23.83.6)$$

Hence, the postulated constitutive equation becomes:

$$\sigma(t) = \sigma^*(\mathbf{E}^N(t), \dot{\mathbf{E}}^N(t), \mathbf{S}(t)), \quad (23.83.7)$$

where we have replaced $\frac{\partial \mathbf{E}_t^N}{\partial t}$ by $\dot{\mathbf{E}}^N$, and σ^* is a function of its arguments.

For a special case,

$$\sigma(t) = \sigma^*(\dot{\mathbf{E}}^N(t), \mathbf{S}(t)), \quad (23.83.8)$$

we may write

$$\dot{\mathbf{E}}_t^N = f(\mathbf{S}(t), \mathbf{s}(t)), \quad (23.83.9)$$

which states that the nonlinear strain rate is the function of stress and a state variable which represents the history of loading. Therefore, the proposed kinetic equation for foam materials is:

$$\dot{\mathbf{E}}^N = \frac{\sigma}{\|\sigma\|} D_0 \exp \left[-c_0 \left(\frac{\text{tr}(\sigma \mathbf{S})}{(\|\sigma\|)^2} \right)^{2n_0} \right], \quad (23.83.10)$$

where D_0 , c_0 , and n_0 are material constants, and \mathbf{S} is the overall state variable. If either $D_0 = 0$ or $c_0 \rightarrow \infty$ then the nonlinear strain rate vanishes.

$$\dot{S}_{ij} = [c_1(a_{ij}R - c_2S_{ij})P + c_3W^{n_1}(\|\dot{\mathbf{E}}^N\|)^{n_2}I_{ij}]R \quad (23.83.11)$$

$$R = 1 + c_4 \left(\frac{\|\dot{\mathbf{E}}^N\|}{c_5} - 1 \right)^{n_3} \quad (23.83.12)$$

$$P = \text{tr}(\sigma \dot{\mathbf{E}}^N) \quad (23.83.13)$$

$$W = \int \text{tr}(\sigma d\mathbf{E}), \quad (23.83.14)$$

where $c_1, c_2, c_3, c_4, c_5, n_1, n_2, n_3$, and a_{ij} are material constants and:

$$\begin{aligned} \|\sigma\| &= (\sigma_{ij}\sigma_{ij})^{\frac{1}{2}}, \\ \|\dot{\mathbf{E}}\| &= (\dot{E}_{ij}\dot{E}_{ij})^{\frac{1}{2}}, \\ \|\dot{\mathbf{E}}^N\| &= (\dot{E}_{ij}^N\dot{E}_{ij}^N)^{\frac{1}{2}}. \end{aligned} \quad (23.83.15)$$

In the implementation by Fu Chang the model was simplified such that the input constants a_{ij} and the state variables S_{ij} are scalars.

Viscous damping in the model follows an implementation identical to that of material type 57.

23.84 Material Model 84 and 85: Winfrith Concrete

Pressure is positive in compression; volumetric strain is given by the natural log of the relative volume and is negative in compression. The tabulated data are given in order of increasing compression, with no initial zero point.

If the volume compaction curve is omitted, the following scaled curve is automatically used where p_1 is the pressure at uniaxial compressive failure computed from:

$$p_1 = \frac{\sigma_c}{3}, \quad (23.84.1)$$

and K is the unloading bulk modulus computed from

$$K = \frac{E_s}{3(1 - 2\nu)}, \quad (23.84.2)$$

where E_s is the input tangent modulus for concrete and ν is Poisson's ratio.

Volumetric Strain	Pressure (MPa)
$-p_1/K$	$1.00 \times p_1$
-0.002	$1.50 \times p_1$
-0.004	$3.00 \times p_1$
-0.010	$4.80 \times p_1$
-0.020	$6.00 \times p_1$
-0.030	$7.50 \times p_1$
-0.041	$9.45 \times p_1$
-0.051	$11.55 \times p_1$
-0.062	$14.25 \times p_1$
-0.094	$25.05 \times p_1$

Table 23.3. Default pressure versus volumetric strain curve for concrete if the curve is not defined.

23.87 Material Model 87: Cellular Rubber

This material model provides a cellular rubber model combined with linear viscoelasticity as outlined by Christensen [1980].

Rubber is generally considered to be fully incompressible since the bulk modulus greatly exceeds the shear modulus in magnitude. To model the rubber as an unconstrained material a hydrostatic work term, $W_H(J)$, is included in the strain energy functional which is function of the relative volume, J , [Ogden, 1984]:

$$\begin{aligned}
 W(J_1, J_2, J) &= \sum_{p,q=0}^n C_{pq} (J_1 - 3)^p (J_2 - 3)^q + W_H(J) \\
 J_1 &= I_1 I_3^{-1/3} \\
 J_2 &= I_2 I_3^{-2/3}
 \end{aligned} \tag{23.87.1}$$

In order to prevent volumetric work from contributing to the hydrostatic work the first and second invariants are modified as shown. This procedure is described in more detail by Sussman and Bathe [1987].

The effects of confined air pressure in its overall response characteristics are included by augmenting the stress state within the element by the air pressure.

$$\sigma_{ij} = \sigma_{ij}^{\text{sk}} - \delta_{ij} \sigma^{\text{air}}, \tag{23.87.2}$$

where σ_{ij}^{sk} is the bulk skeletal stress and σ^{air} is the air pressure computed from the equation:

$$\sigma^{\text{air}} = -\frac{p_0 \gamma}{1 + \gamma - \phi'}, \tag{23.87.3}$$

where p_0 is the initial foam pressure usually taken as the atmospheric pressure and γ defines the volumetric strain

$$\gamma = V - 1 + \gamma_0, \tag{23.87.4}$$

where V is the relative volume of the voids and γ_0 is the initial volumetric strain which is typically zero. The rubber skeletal material is assumed to be incompressible.

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \tag{23.87.5}$$

or in terms of the second Piola-Kirchhoff stress, S_{ij} , and Green's strain tensor, E_{ij} ,

$$S_{ij} = \int_0^t G_{ijkl}(t - \tau) \frac{\partial E_{kl}}{\partial \tau} d\tau, \quad (23.87.6)$$

where $g_{ijkl}(t - \tau)$ and $G_{ijkl}(t - \tau)$ are the relaxation functions for the different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

Since we wish to include only simple rate effects, the relaxation function is represented by one term from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^N \alpha_m e^{-\beta_m t}, \quad (23.87.7)$$

given by,

$$g(t) = E_d e^{-\beta_1 t}. \quad (23.87.8)$$

This model is effectively a Maxwell fluid which consists of a damper and spring in series. We characterize this in the input by a shear modulus, G , and decay constant, β_1 .

The Mooney-Rivlin rubber model is obtained by specifying $n = 1$. In spite of the differences in formulations with Model 27, we find that the results obtained with this model are nearly identical with those of 27 as long as large values of Poisson's ratio are used.

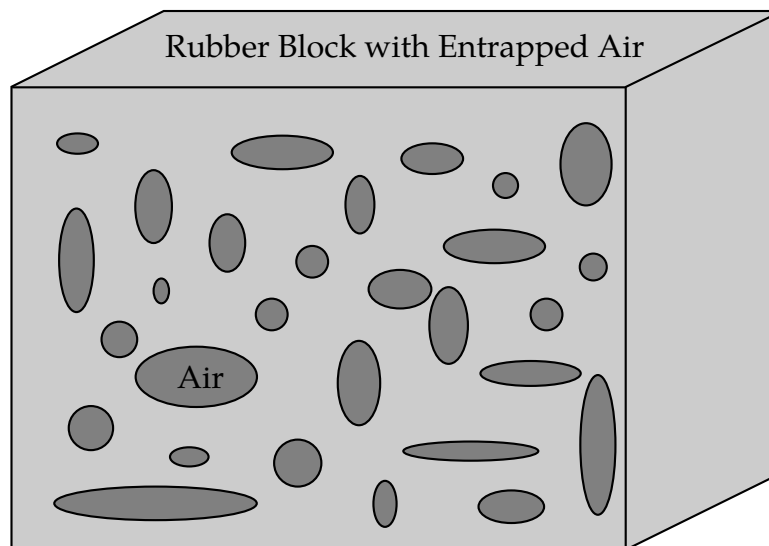


Figure 23.87.1. Cellular rubber with entrapped air. By setting the initial air pressure to zero, an open cell, cellular rubber can be simulated.

23.88 Material Model 88: MTS Model

The Mechanical Threshold Stress (MTS) model is due to Mauldin, Davidson, and Henninger [1990] and is available for applications involving large strains, high pressures and strain rates. As described in the foregoing reference, this model is based on dislocation mechanics and provides a better understanding of the plastic deformation process for ductile materials by using an internal state variable called the mechanical threshold stress. This kinematic quantity tracks the evolution of the material's microstructure along some arbitrary strain, strain rate, and temperature-dependent path using a differential form that balances dislocation generation and recovery processes. Given a value for the mechanical threshold stress, the flow stress is determined using either a thermal-activation-controlled or a drag-controlled kinetics relationship. An equation-of-state is required for solid elements and a bulk modulus must be defined below for shell elements.

The flow stress σ is given by:

$$\sigma = \hat{\sigma}_a + \frac{G}{G_0} [s_{th} \hat{\sigma} + s_{th,i} \hat{\sigma}_i + s_{th,s} \hat{\sigma}_s]. \quad (23.88.1)$$

The first product in the equation for σ contains a micro-structure evolution variable, i.e., $\hat{\sigma}$, called the Mechanical Threshold Stress (MTS), that is multiplied by a constant-structure deformation variable s_{th} ; s_{th} is a function of absolute temperature T and the plastic strain-rates $\dot{\epsilon}^P$. The evolution equation for $\hat{\sigma}$ is a differential hardening law representing dislocation-dislocation interactions:

$$\frac{\partial \hat{\sigma}}{\partial \epsilon^P} \equiv \Theta_o \left[1 - \frac{\tanh\left(\alpha \frac{\hat{\sigma}}{\hat{\sigma}_{\epsilon s}}\right)}{\tanh(\alpha)} \right]. \quad (23.88.2)$$

The term, $\frac{\partial \hat{\sigma}}{\partial \epsilon^P}$, represents the hardening due to dislocation generation and the stress ratio, $\frac{\hat{\sigma}}{\hat{\sigma}_{\epsilon s}}$, represents softening due to dislocation recovery. The threshold stress at zero strain-hardening $\hat{\sigma}_{\epsilon s}$ is called the saturation threshold stress. Relationships for Θ_o , $\hat{\sigma}_{\epsilon s}$ are:

$$\Theta_o = a_o + a_1 \ln\left(\frac{\dot{\epsilon}^P}{\epsilon_0}\right) + a_2 \sqrt{\frac{\dot{\epsilon}^P}{\epsilon_0}}, \quad (23.88.3)$$

which contains the material constants a_o , a_1 , and a_2 . The constant, $\hat{\sigma}_{\epsilon s}$, is given as:

$$\hat{\sigma}_{\epsilon s} = \hat{\sigma}_{\epsilon s o} \left(\frac{\dot{\epsilon}^P}{\dot{\epsilon}_{\epsilon s o}} \right)^{kT/Gb^3A}, \quad (23.88.4)$$

which contains the input constants: $\hat{\sigma}_{\epsilon_{SO}}$, $\dot{\epsilon}_{\epsilon_{SO}}$, b , A , and k . The shear modulus G appearing in these equations is assumed to be a function of temperature and is given by the correlation.

$$G = G_0 - \frac{b_1}{e^{\frac{b_2}{T}} - 1}, \quad (23.88.5)$$

which contains the constants: G_0 , b_1 , and b_2 . For thermal-activation controlled deformation s_{th} is evaluated via an Arrhenius rate equation of the form:

$$s_{th} = \left[1 - \left(\frac{kT \ln \left(\frac{\dot{\epsilon}_0}{\dot{\epsilon}P} \right)}{Gb^3 g_0} \right)^{\frac{1}{q}} \right]^{\frac{1}{p}}. \quad (23.88.6)$$

The absolute temperature is given as:

$$T = T_{ref} + \rho c_p E, \quad (23.88.7)$$

where E is the internal energy density per unit initial volume.

23.89 Material Model 89: Plasticity Polymer

Unlike other LS-DYNA material models, both the input stress-strain curve and the strain to failure are defined as total true strain, not plastic strain. The input can be defined from uniaxial tensile tests; nominal stress and nominal strain from the tests must be converted to true stress and true strain. The elastic component of strain must not be subtracted out.

The stress-strain curve is permitted to have sections steeper (i.e. stiffer) than the elastic modulus. When these are encountered the elastic modulus is increased to prevent spurious energy generation.

23.90 Material Model 90: Acoustic

This model is appropriate for tracking low-pressure stress waves in an acoustic media such as air or water and can be used only with the acoustic pressure element formulation. The acoustic pressure element requires only one unknown per node. This element is very cost effective.

23.91 Material Model 91: Soft Tissue

The overall strain energy W is "uncoupled" and includes two isotropic deviatoric matrix terms, a fiber term F , and a bulk term:

$$W = C_1(\tilde{I}_1 - 3) + C_2(\tilde{I}_2 - 3) + F(\lambda) + \frac{1}{2}K[\ln(J)]^2. \quad (23.91.1)$$

Here, \tilde{I}_1 and \tilde{I}_2 are the deviatoric invariants of the right Cauchy deformation tensor, λ is the deviatoric part of the stretch along the current fiber direction, and $J = \det \mathbf{F}$ is the volume ratio. The material coefficients C_1 and C_2 are the Mooney-Rivlin coefficients, while K is the effective bulk modulus of the material (input parameter XK).

The derivatives of the fiber term F are defined to capture the behavior of crimped collagen. The fibers are assumed to be unable to resist compressive loading - thus the model is isotropic when $\lambda < 1$. An exponential function describes the straightening of the fibers, while a linear function describes the behavior of the fibers once they are straightened past a critical fiber stretch level $\lambda \geq \lambda^*$ (input parameter XLAM):

$$\frac{\partial F}{\partial \lambda} = \begin{cases} 0 & \lambda < 1 \\ \frac{C_3}{\lambda} [\exp(C_4(\lambda - 1)) - 1] & \lambda < \lambda^* \\ \frac{1}{\lambda} (C_5\lambda + C_6) & \lambda \geq \lambda^* \end{cases}. \quad (23.91.2)$$

Coefficients C_3 , C_4 , and C_5 must be defined by the user. C_6 is determined by LS-DYNA to ensure stress continuity at $\lambda = \lambda^*$. Sample values for the material coefficients $C_1 - C_5$ and λ^* for ligament tissue can be found in Quapp and Weiss [1998]. The bulk modulus K should be at least 3 orders of magnitude larger than C_1 to ensure near-incompressible material behavior.

Viscoelasticity is included via a convolution integral representation for the time-dependent second Piola-Kirchoff stress $\mathbf{S}(\mathbf{C}, t)$:

$$\mathbf{S}(\mathbf{C}, t) = \mathbf{S}^e(\mathbf{C}) + \int_0^t 2G(t-s) \frac{\partial W}{\partial \mathbf{C}(s)} ds. \quad (23.91.3)$$

Here, \mathbf{S}^e is the elastic part of the second PK stress as derived from the strain energy, and $G(t-s)$ is the reduced relaxation function, represented by a Prony series:

$$G(t) = \sum_{i=1}^6 S_i \exp\left(-\frac{t}{T_i}\right). \quad (23.91.4)$$

Puso and Weiss [1998] describe a graphical method to fit the Prony series coefficients to relaxation data that approximates the behavior of the continuous relaxation function proposed by Y-C. Fung, as quasilinear viscoelasticity.

23.94 Material Model 94: Inelastic Spring Discrete Beam

The yield force is taken from the load curve:

$$F^Y = F_y(\Delta L^{\text{plastic}}), \quad (23.94.1)$$

where L^{plastic} is the plastic deflection. A trial force is computed as:

$$F^T = F^n + K \cdot \Delta \dot{L} \cdot \Delta t, \quad (23.94.2)$$

and is checked against the yield force to determine F:

$$F = \begin{cases} F^Y & \text{if } F^T > F^Y \\ F^T & \text{if } F^T \leq F^Y \end{cases} \quad (23.94.3)$$

The final force, which includes rate effects and damping, is given by:

$$F^{n+1} = F \cdot \left[1 + C1 \cdot \Delta \dot{L} + C2 \cdot \text{sgn}(\Delta \dot{L}) \ln \left(\max \left\{ 1, \frac{|\Delta \dot{L}|}{\text{DLE}} \right\} \right) \right] + D \Delta \dot{L} + g(\Delta L) h(\Delta \dot{L}), \quad (23.94.4)$$

where C1, C2 are damping coefficients, DLE is a factor to scale time units.

Unless the origin of the curve starts at (0,0), the negative part of the curve is used when the spring force is negative where the negative of the plastic displacement is used to interpolate, F_y . The positive part of the curve is used whenever the force is positive. In these equations, ΔL is the change in length

$$\Delta L = \text{current length} - \text{initial length}. \quad (23.94.5)$$

23.96 Material Model 96: Brittle Damage Model

A full description of the tensile and shear damage parts of this material model is given in Govindjee, Kay and Simo [1994,1995]. It is an anisotropic brittle damage model designed primarily for concrete, though it can be applied to a wide variety of brittle materials. It admits progressive degradation of tensile and shear strengths across smeared cracks that are initiated under tensile loadings. Compressive failure is governed by a simplistic J2 flow correction that can be disabled if not desired. Damage is handled by treating the rank 4 elastic stiffness tensor as an evolving internal variable for the material. Softening induced mesh dependencies are handled by a characteristic length method [Oliver 1989].

Description of properties:

1. E is the Young's modulus of the undamaged material also known as the virgin modulus.
2. ν is the Poisson's ratio of the undamaged material also known as the virgin Poisson's ratio.
3. f_n is the initial principal tensile strength (stress) of the material. Once this stress has been reached at a point in the body a smeared crack is initiated there with a normal that is co-linear with the 1st principal direction. Once initiated, the crack is fixed at that location, though it will convect with the motion of the body. As the loading progresses the allowed tensile traction normal to the crack plane is progressively degraded to a small machine dependent constant.

The degradation is implemented by reducing the material's modulus normal to the smeared crack plane according to a maximum dissipation law that incorporates exponential softening. The restriction on the normal tractions is given by

$$\phi_t = (\mathbf{n} \otimes \mathbf{n}) : \boldsymbol{\sigma} - f_n + (1 - \varepsilon)f_n(1 - \exp[-H\alpha]) \leq 0, \quad (23.96.1)$$

where \mathbf{n} is the smeared crack normal, ε is the small constant, H is the softening modulus, and α is an internal variable. H is set automatically by the program; see g_c below. α measures the crack field intensity and is output in the equivalent plastic strain field, $\bar{\varepsilon}^p$, in a normalized fashion.

The evolution of alpha is governed by a maximum dissipation argument. When the normalized value reaches unity it means that the material's strength has been reduced to 2% of its original value in the normal and parallel directions to the smeared crack. Note that for plotting purposes, it is never output greater than 5.

4. f_s is the initial shear traction that may be transmitted across a smeared crack plane. The shear traction is limited to be less than or equal to $f_s(1 - \beta)(1 - \exp[-H\alpha])$, through the use of two orthogonal shear damage surfaces. Note that the shear degradation is coupled to the tensile degradation through the internal variable α which measures the intensity of the crack field. β is the shear retention factor defined below. The shear degradation is taken care of by reducing the material's shear stiffness parallel to the smeared crack plane.
5. g_c is the fracture toughness of the material. It should be entered as fracture energy per unit area crack advance. Once entered the softening modulus is automatically calculated based on element and crack geometries.
6. β is the shear retention factor. As the damage progresses the shear tractions allowed across the smeared crack plane asymptote to the product βf_s .
7. η represents the viscosity of the material. Viscous behavior is implemented as a simple Perzyna regularization method. This allows for the inclusion of first order rate effects. The use of some viscosity is recommended as it serves as regularizing parameter that increases the stability of calculations.
8. σ_y is a uniaxial compressive yield stress. A check on compressive stresses is made using the J2 yield function $s:s - \sqrt{\frac{2}{3}}\sigma_y \leq 0$, where s is the stress deviator. If violated, a J2 return mapping correction is executed. This check is executed when (1) no damage has taken place at an integration point yet, (2) when damage has taken place at a point but the crack is currently closed, and (3) during active damage after the damage integration (ie. as an operator split). Note that if the crack is open, the plasticity correction is done in the plane-stress subspace of the crack plane.

Remark: A variety of experimental data has been replicated using this model from quasi-static to explosive situations. Reasonable properties for a standard grade concrete would be $E = 3.15 \times 10^6$ psi, $f_n = 450$ psi, $f_s = 2100$ psi, $\nu = 0.2$, $g_c = 0.8$ lbs/in, $\beta = 0.03$, $\eta = 0.0$ psi-sec, $\sigma_y = 4200$ psi. For stability, values of η between 104 to 106 psi/sec are recommended. Our limited experience thus far has shown that many problems require nonzero values of η to run to avoid error terminations. Various other internal variables such as crack orientations and degraded stiffness tensors are internally calculated but currently not available for output.

23.97 Material Model 97: General Joint Discrete Beam

For explicit calculations, the additional stiffness due to this joint may require additional mass and inertia for stability. Mass and rotary inertia for this beam element is based on the defined mass density, the volume, and the mass moment of inertia defined in the *SECTION_ BEAM input.

The penalty stiffness applies to explicit calculations. For implicit calculations, constraint equations are generated and imposed on the system equations; therefore, these constants, RPST and RPSR, are not used.

23.98 Material Model 98: Simplified Johnson Cook

Johnson and Cook express the flow stress as

$$\sigma_y = \left(A + B \bar{\epsilon}^{p^n} \right) (1 + C \ln \dot{\epsilon}^*), \quad (23.98.1)$$

where

A, B, C and n are input constants

$\bar{\epsilon}^p$ effective plastic strain

$\dot{\epsilon}^* = \frac{\dot{\epsilon}}{\dot{\epsilon}_0}$ effective strain rate for $\dot{\epsilon}_0 = 1 \text{ s}^{-1}$

The maximum stress is limited by SIGMAX and SIGSAT by:

$$\sigma_y = \min \left\{ \min \left[A + B \bar{\epsilon}^{p^n}, \text{SIGMAX} \right] (1 + C \ln \dot{\epsilon}^*), \text{SIGSAT} \right\}. \quad (23.98.2)$$

Failure occurs when the effective plastic strain exceeds PSFAIL.

If the viscoplastic option is active, $VP = 1.0$, the parameters SIGMAX and SIGSAT are ignored since these parameters make convergence of the viscoplastic strain iteration loop difficult to achieve. The viscoplastic option replaces the plastic strain in the forgoing equations by the viscoplastic strain and the strain rate by the viscoplastic strain rate. Numerical noise is substantially reduced by the viscoplastic formulation.

23.100 Material Model 100: Spot Weld

This material model applies to beam element type 9 for spot welds. These beam elements may be placed between any two deformable shell surfaces, see [Figure 23.100.1](#), and tied with type 7 constraint contact which eliminates the need to have adjacent nodes at spot weld locations. Beam spot welds may be placed between rigid bodies and rigid/deformable bodies by making the node on one end of the spot weld a rigid body node which can be an extra node for the rigid body. In the same way, rigid bodies may also be tied together with this spot weld option.

It is advisable to include all spot welds, which provide the tracked nodes, and spot welded materials, which define the reference segments, within a single type 7 tied interface. As a constraint method, multiple type 7 interfaces are treated independently which can lead to significant problems if such interfaces share common nodal points. The offset option, "o 7", should not be used with spot welds.

The DAMAGE-FAILURE option causes one additional line to be read with the damage parameter and a flag that determines how failure is computed from the resultants. On this line the parameter, DMG, if nonzero, invokes damage mechanics combined with the plasticity model to achieve a smooth drop off of the resultant forces prior to the removal of the spot weld. The parameter FOPT determines the method used in computing resultant based failure, which is unrelated to damage.

The weld material is modeled with isotropic hardening plasticity coupled to two failure models. The first model specifies a failure strain which fails each integration point

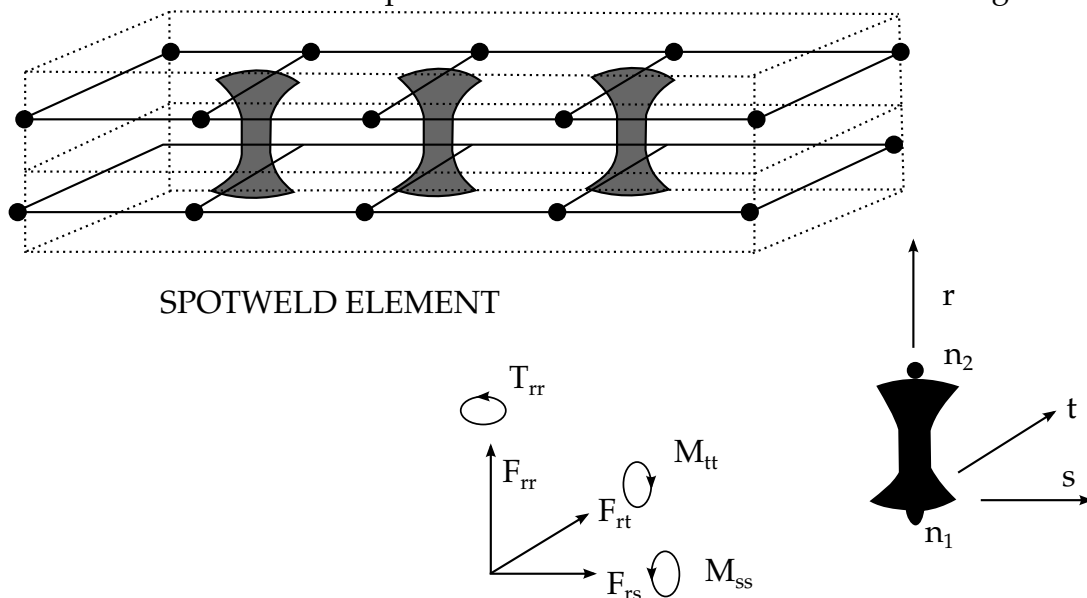


Figure 23.100.1. Deformable spotwelds can be arbitrarily placed within the structure.

in the spot weld independently. The second model fails the entire weld if the resultants are outside of the failure surface defined by:

$$\left(\frac{N_{rr}}{N_{rrF}}\right)^2 + \left(\frac{N_{rs}}{N_{rsF}}\right)^2 + \left(\frac{N_{rt}}{N_{rtF}}\right)^2 + \left(\frac{M_{rr}}{M_{rrF}}\right)^2 + \left(\frac{M_{ss}}{M_{ssF}}\right)^2 + \left(\frac{T_{rr}}{T_{rrF}}\right)^2 - 1 = 0, \quad (23.100.1)$$

where the numerators in the equation are the resultants calculated in the local coordinates of the cross section, and the denominators are the values specified in the input. If the user defined parameter, NF, which the number of force vectors stored for filtering, is nonzero the resultants are filtered before failure is checked. The default value is set to zero which is generally recommended unless oscillatory resultant forces are observed in the time history databases. Even though these welds should not oscillate significantly, this option was added for consistency with the other spot weld options. NF affects the storage since it is necessary to store the resultant forces as history variables.

If the failure strain is set to zero, the failure strain model is not used. In a similar manner, when the value of a resultant at failure is set to zero, the corresponding term in the failure surface is ignored. For example, if only N_{rrF} is nonzero, the failure surface is reduced to $|N_{rr}| = N_{rrF}$. None, either, or both of the failure models may be active depending on the specified input values.

The inertias of the spot welds are scaled during the first time step so that their stable time step size is Δt . A strong compressive load on the spot weld at a later time may reduce the length of the spot weld so that stable time step size drops below Δt . If the value of Δt is zero, mass scaling is not performed, and the spot welds will probably limit the time step size. Under most circumstances, the inertias of the spot welds are small enough that scaling them will have a negligible effect on the structural response and the use of this option is encouraged.

Spotweld force history data is written into the SWFORC ASCII file. In this database the resultant moments are not available, but they are in the binary time history database.

The constitutive properties for the damaged material are obtained from the undamaged material properties. The amount of damage evolved is represented by the constant, ω , which varies from zero if no damage has occurred to unity for complete rupture. For uniaxial loading, the nominal stress in the damaged material is given by

$$\sigma_{\text{nominal}} = \frac{P}{A}, \quad (23.100.2)$$

where P is the applied load and A is the surface area. The true stress is given by:

$$\sigma_{\text{true}} = \frac{P}{A - A_{\text{loss}}}, \quad (23.100.3)$$

where A_{loss} is the void area. The damage variable can then be defined:

$$\omega = \frac{A_{\text{loss}}}{A}, 0 \leq \omega \leq 1. \quad (23.100.4)$$

In this model damage is defined in terms of plastic strain after the failure strain is exceeded:

$$\omega = \frac{\varepsilon_{\text{eff}}^{\text{P}} - \varepsilon_{\text{failure}}^{\text{P}}}{\varepsilon_{\text{rupture}}^{\text{P}} - \varepsilon_{\text{failure}}^{\text{P}}} \quad \text{if} \quad \varepsilon_{\text{failure}}^{\text{P}} \leq \varepsilon_{\text{eff}}^{\text{P}} \leq \varepsilon_{\text{rupture}}^{\text{P}}. \quad (23.100.5)$$

After exceeding the failure strain softening begins and continues until the rupture strain is reached.

23.101 Material Model 101: GE Thermoplastics

The constitutive model for this approach is:

$$\dot{\epsilon}_p = \dot{\epsilon}_0 \exp(A\{\sigma - S(\epsilon_p)\}) \times \exp(-p\alpha A), \quad (23.101.1)$$

where $\dot{\epsilon}_0$ and A are rate dependent yield stress parameters, $S(\epsilon_p)$ internal resistance (strain hardening) and α is a pressure dependence parameter.

In this material the yield stress may vary throughout the finite element model as a function of strain rate and hydrostatic stress. Post yield stress behavior is captured in material softening and hardening values. Finally, ductile or brittle failure measured by plastic strain or maximum principal stress respectively is accounted for by automatic element deletion.

Although this may be applied to a variety of engineering thermoplastics, GE Plastics have constants available for use in a wide range of commercially available grades of their engineering thermoplastics.

23.102 Material Model 102: Hyperbolic Sine

Resistance to deformation is both temperature and strain rate dependent. The flow stress equation is:

$$\sigma = \frac{1}{\alpha} \sinh^{-1} \left(\left[\frac{Z}{A} \right]^{\frac{1}{N}} \right), \quad (23.102.1)$$

where Z , the Zener-Holloman temperature compensated strain rate, is:

$$Z = \dot{\epsilon} \exp \left(\frac{Q}{GT} \right). \quad (23.102.2)$$

The units of the material constitutive constants are as follows: A (1/sec), N (dimensionless), α (1/MPa), the activation energy for flow, Q (J/mol), and the universal gas constant, G [J/(mol · K)]. The value of G will only vary with the unit system chosen. Typically it will be either 8.3145 J/(mol · K), or 40.8825 lb · in/mol · R.

The final equation necessary to complete the description of high strain rate deformation herein is one that allows computation of the temperature change during the deformation. In the absence of a coupled thermo-mechanical finite element code we assume adiabatic temperature change and follow the empirical assumption that 90-95% of the plastic work is dissipated as heat. Thus the heat generation coefficient is

$$HC \approx \frac{0.9}{\rho C_v}, \quad (23.102.3)$$

where ρ is the material density and C_v is the specific heat.

23.103 Material Model 103: Anisotropic Viscoplastic

The uniaxial stress-strain curve is given on the following form

$$\sigma(\epsilon_{\text{eff}}^p, \dot{\epsilon}_{\text{eff}}^p) = \sigma_0 + Q_{r1}[(1 - \exp(-C_{r1}\epsilon_{\text{eff}}^p))] + Q_{r2}[1 - \exp(-C_{r2}\epsilon_{\text{eff}}^p)] \\ + Q_{\chi1}[(1 - \exp(-C_{\chi1}\epsilon_{\text{eff}}^p))] + Q_{\chi2}[(1 - \exp(-C_{\chi2}\epsilon_{\text{eff}}^p))] + V_k \dot{\epsilon}_{\text{eff}}^p V_m, \quad (23.103.1)$$

For bricks the following yield criteria is used

$$F(\sigma_{22} - \sigma_{33})^2 + G(\sigma_{33} - \sigma_{11})^2 + H(\sigma_{11} - \sigma_{22})^2 + 2L\sigma_{23}^2 + 2M\sigma_{31}^2 + 2N\sigma_{12}^2 \\ = [\sigma(\epsilon_{\text{eff}}^p, \dot{\epsilon}_{\text{eff}}^p)]^2, \quad (23.103.2)$$

where ϵ_{eff}^p is the effective plastic strain and $\dot{\epsilon}_{\text{eff}}^p$ is the effective plastic strain rate. For shells the anisotropic behavior is given by R_{00} , R_{45} and R_{90} . When $V_k = 0$ the material will behave elasto-plastically. Default values are given by:

$$F = G = H = \frac{1}{2}, \quad (23.103.3)$$

$$L = M = N = \frac{3}{2}, \quad (23.103.4)$$

$$R_{00} = R_{45} = R_{90} = 1. \quad (23.103.5)$$

Strain rate is accounted for using the Cowper-Symonds model which, e.g., model 3, scales the yield stress with the factor:

$$1 + \left(\frac{\dot{\epsilon}}{C}\right)^{1/p}. \quad (23.103.6)$$

To convert these constants set the viscoelastic constants, V_k and V_m , to the following values:

$$V_k = \sigma \left(\frac{1}{C}\right)^{\frac{1}{p}}, \quad (23.103.7) \\ V_m = \frac{1}{p}.$$

This model properly treats rate effects and should provide superior results to models 3 and 24.

23.104 Material Model 104: Continuum Damage Mechanics Model

Anisotropic Damage model (FLAG = -1). At each thickness integration points, an anisotropic damage law acts on the plane stress tensor in the directions of the principal total shell strains, ε_1 and ε_2 , as follows:

$$\begin{aligned}\sigma_{11} &= (1 - D_1(\varepsilon_1))\sigma_{110}, \\ \sigma_{22} &= (1 - D_2(\varepsilon_2))\sigma_{220}, \\ \sigma_{12} &= \left(1 - \frac{D_1 + D_2}{2}\right)\sigma_{120}.\end{aligned}\tag{23.104.1}$$

The transverse plate shear stresses in the principal strain directions are assumed to be damaged as follows:

$$\begin{aligned}\sigma_{13} &= \left(1 - \frac{D_1}{2}\right)\sigma_{130}, \\ \sigma_{23} &= \left(1 - \frac{D_2}{2}\right)\sigma_{230}.\end{aligned}\tag{23.104.2}$$

In the anisotropic damage formulation, $D_1(\varepsilon_1)$ and $D_2(\varepsilon_2)$ are anisotropic damage functions for the loading directions 1 and 2, respectively. Stresses σ_{110} , σ_{220} , σ_{120} , σ_{130} and σ_{230} are stresses in the principal shell strain directions as calculated from the undamaged elastic-plastic material behavior. The strains ε_1 and ε_2 are the magnitude of the principal strains calculated upon reaching the damage thresholds. Damage can only develop for tensile stresses, and the damage functions $D_1(\varepsilon_1)$ and $D_2(\varepsilon_2)$ are identical to zero for negative strains ε_1 and ε_2 . The principal strain directions are fixed within an integration point as soon as either principal strain exceeds the initial threshold strain in tension. A more detailed description of the damage evolution for this material model is given in the description of material 82.

The Continuum Damage Mechanics (CDM) model (FLAG ≥ 0) is based on a CDM model proposed by Lemaitre [1992]. The effective stress $\tilde{\sigma}$, which is the stress calculated over the section that effectively resist the forces and reads.

$$\tilde{\sigma} = \frac{\sigma}{1 - D},\tag{23.104.3}$$

where D is the damage variable. The evolution equation for the damage variable is defined as

$$\dot{D} = \begin{cases} \frac{Y}{S(1-D)} \dot{r} & \text{for } r > r_D \text{ and } \sigma_1 > 0 \\ 0 & \text{otherwise} \end{cases} \quad (23.104.4)$$

where r_D is the damage threshold, Y is a positive material constant, S is the strain energy release rate, and σ_1 is the maximal principal stress. The strain energy density release rate is

$$Y = \frac{1}{2} \mathbf{e}_e : \mathbf{C} : \mathbf{e}_e = \frac{\sigma_{vm}^2 R_v}{2E(1-D)^2}, \quad (23.104.5)$$

where σ_{vm} is the equivalent von Mises stress. The triaxiality function R_v is defined as

$$R_v = \frac{2}{3} (1 + \nu) + 3(1 - 2\nu) \left(\frac{\sigma_H}{\sigma_{vm}} \right)^2. \quad (23.104.6)$$

The uniaxial stress-strain curve is given in the following form

$$\sigma(r, \dot{\epsilon}_{eff}^P) = \sigma_0 + Q_1(1 - \exp(-C_1 r)) + Q_2(1 - \exp(-C_2 r)) + V_k \dot{\epsilon}_{eff}^P V_m, \quad (23.104.7)$$

where r is the damage accumulated plastic strain, which can be calculated by

$$\dot{r} = \dot{\epsilon}_{eff}^P (1 - D). \quad (23.104.8)$$

For bricks the following yield criteria is used

$$F(\tilde{\sigma}_{22} - \tilde{\sigma}_{33})^2 + G(\tilde{\sigma}_{33} - \tilde{\sigma}_{11})^2 + H(\tilde{\sigma}_{11} - \tilde{\sigma}_{22})^2 + 2L\tilde{\sigma}_{23}^2 + 2M\tilde{\sigma}_{31}^2 + 2N\tilde{\sigma}_{12}^2 = \sigma(r, \dot{\epsilon}_{eff}^P), \quad (23.104.9)$$

where r is the damage effective viscoplastic strain and $\dot{\epsilon}_{eff}^P$ is the effective viscoplastic strain rate. For shells the anisotropic behavior is given by the R-values: R_{00} , R_{45} , and R_{90} . When $V_k = 0$ the material will behave as an elastoplastic material without rate effects. Default values for the anisotropic constants are given by:

$$F = G = H = \frac{1}{2}, \quad (23.104.10)$$

$$L = M = N = \frac{3}{2}, \quad (23.104.11)$$

$$R_{00} = R_{45} = R_{90} = 1, \quad (23.104.12)$$

so that isotropic behavior is obtained.

Strain rate is accounted for using the Cowper-Symonds model which scales the yield stress with the factor:

$$1 + \left(\frac{\dot{\epsilon}}{C} \right)^{1/p}. \quad (23.104.13)$$

To convert these constants, set the viscoelastic constants, V_k and V_m , to the following values:

$$\begin{aligned} V_k &= \sigma \left(\frac{1}{C} \right)^{\frac{1}{p}}, \\ V_m &= \frac{1}{p}. \end{aligned} \quad (23.104.14)$$

23.106 Material Model 106: Elastic Viscoplastic Thermal

If LCSS is not given any value the uniaxial stress-strain curve has the form

$$\begin{aligned} \sigma(\epsilon_{\text{eff}}^p) = & \sigma_0 + Q_{r1}(1 - \exp(-C_{r1}\epsilon_{\text{eff}}^p)) + Q_{r2}(1 - \exp(-C_{r2}\epsilon_{\text{eff}}^p)) \\ & + Q_{\chi1}(1 - \exp(-C_{\chi1}\epsilon_{\text{eff}}^p)) + Q_{\chi2}(1 - \exp(-C_{\chi2}\epsilon_{\text{eff}}^p)). \end{aligned} \quad (23.106.1)$$

Viscous effects are accounted for using the Cowper-Symonds model, which scales the yield stress with the factor:

$$1 + \left(\frac{\dot{\epsilon}_{\text{eff}}^p}{C} \right)^{1/p}. \quad (23.106.2)$$

23.110 Material Model 110: Johnson-Holmquist Ceramic Model

The Johnson-Holmquist plasticity damage model is useful for modeling ceramics, glass and other brittle materials. A more detailed description can be found in a paper by Johnson and Holmquist [1993].

The equivalent stress for a ceramic-type material is given in terms of the damage parameter D by

$$\sigma^* = \sigma_i^* - D(\sigma_i^* - \sigma_f^*). \quad (23.110.1)$$

Here,

$$\sigma_i^* = a(p^* + t^*)^n (1 + c \ln \dot{\epsilon}^*), \quad (23.110.2)$$

represents the intact, undamaged behavior. The superscript, '*', indicates a normalized quantity. The stresses are normalized by the equivalent stress at the Hugoniot elastic limit (see below), the pressures are normalized by the pressure at the Hugoniot elastic limit, and the strain rate by the reference strain rate defined in the input. In this equation a is the intact normalized strength parameter, c is the strength parameter for strain rate dependence, $\dot{\epsilon}^*$ is the normalized plastic strain rate, and,

$$\begin{aligned} t^* &= \frac{T}{\text{PHEL}}, \\ p^* &= \frac{p}{\text{PHEL}}, \end{aligned} \quad (23.110.3)$$

where T is the maximum tensile pressure strength, PHEL is the pressure component at the Hugoniot elastic limit, and p is the pressure.

$$D = \sum \Delta \epsilon^P / \epsilon_f^P, \quad (23.110.4)$$

represents the accumulated damage based upon the increase in plastic strain per computational cycle and the plastic strain to fracture

$$\epsilon_f^P = d_1 (p^* + t^*)^{d_2}, \quad (23.110.5)$$

where d_1 and d_2 are user defined input parameters. The equation:

$$\sigma_f^* = b(p^*)^m (1 + c \ln \dot{\epsilon}^*) \leq \text{SFMAX}, \quad (23.110.6)$$

represents the damaged behavior where b is an input parameter and SFMAX is the maximum normalized fracture strength. The parameter, d_1 , controls the rate at which damage accumulates. If it approaches 0, full damage can occur in one time step, i.e., instantaneously. This rate parameter is also the best parameter to vary if one attempts to reproduce results generated by another finite element program.

In undamaged material, the hydrostatic pressure is given by

$$p = k_1\mu + k_2\mu^2 + k_3\mu^3, \quad (23.110.7)$$

where $\mu = \rho/\rho_0 - 1$. When damage starts to occur, there is an increase in pressure. A fraction defined in the input, between 0 and 1, of the elastic energy loss, β , is converted into hydrostatic potential energy, which results in an increase in pressure. The details of this pressure increase are given in the reference.

Given HEL and the shear modulus, g , μ_{hel} can be found iteratively from

$$\text{HEL} = k_1\mu_{\text{hel}} + k_2\mu_{\text{hel}}^2 + k_3\mu_{\text{hel}}^3 + \frac{4}{3}g \left(\frac{\mu_{\text{hel}}}{1 + \mu_{\text{hel}}} \right), \quad (23.110.8)$$

and, subsequently, for normalization purposes,

$$\text{PHEL} = k_1\mu_{\text{hel}} + k_2\mu_{\text{hel}}^2 + k_3\mu_{\text{hel}}^3, \quad (23.110.9)$$

and

$$\sigma_{\text{hel}} = 1.5(\text{HEL} - \text{PHEL}). \quad (23.110.10)$$

These are calculated automatically by LS-DYNA if PHEL is zero on input.

23.111 Material Model 111: Johnson-Holmquist Concrete Model

This model can be used for concrete subjected to large strains, high strain rates, and high pressures. The equivalent strength is expressed as a function of the pressure, strain rate, and damage. The pressure is expressed as a function of the volumetric strain and includes the effect of permanent crushing. The damage is accumulated as a function of the plastic volumetric strain, equivalent plastic strain and pressure. A more detailed description of this model can be found in the paper by Holmquist, Johnson, and Cook [1993]

The normalized equivalent stress is defined as

$$\sigma^* = \frac{\sigma}{f'_c} \quad (23.111.1)$$

where σ is the actual equivalent stress, and f'_c is the quasi-static uniaxial compressive strength. The yield stress is defined in terms of the input parameters a , b , c , and n as:

$$\sigma^* = [a(1 - D) + bp^{*n}][1 - c \ln(\dot{\epsilon}^*)], \quad (23.111.2)$$

where D is the damage parameter, $p^* = p/f'_c$ is the normalized pressure and $\dot{\epsilon}^* = \dot{\epsilon}/\dot{\epsilon}_0$ is the dimensionless strain rate. The model accumulates damage both from equivalent plastic strain and plastic volumetric strain, and is expressed as

$$D = \sum \frac{\Delta \epsilon_p + \Delta \mu_p}{D_1(p^* + T^*)^{D_2}}, \quad (23.111.3)$$

where $\Delta \epsilon_p$ and $\Delta \mu_p$ are the equivalent plastic strain and plastic volumetric strain, D_1 and D_2 are material constants and $T^* = T/f'_c$ is the normalized maximum tensile hydrostatic pressure where T is the maximum tensile hydrostatic pressure.

The pressure for fully dense material is expressed as:

$$P = K_1 \bar{\mu} + K_2 \bar{\mu}^2 + K_3 \bar{\mu}^3, \quad (23.111.4)$$

where K_1 , K_2 and K_3 are material constants and the modified volumetric strain is defined as

$$\bar{\mu} = \frac{\mu - \mu_{\text{lock}}}{1 + \mu_{\text{lock}}}, \quad (23.111.5)$$

where μ_{lock} is the locking volumetric strain.

23.115 Material Model 115: Elastic Creep Model

The effective creep strain, $\bar{\epsilon}^c$, given as:

$$\bar{\epsilon}^c = A \bar{\sigma}^n \bar{t}^m, \quad (23.115.1)$$

where A , n , and m are constants and \bar{t} is the effective time. The effective stress, $\bar{\sigma}$, is defined as:

$$\bar{\sigma} = \sqrt{\frac{3}{2} \sigma_{ij} \sigma_{ij}}. \quad (23.115.2)$$

The creep strain, therefore, is only a function of the deviatoric stresses. The volumetric behavior for this material is assumed to be elastic. By varying the time constant m primary creep ($m < 1$), secondary creep ($m = 1$), and tertiary creep ($m > 1$) can be modeled. This model is described by Whirley and Henshall (1992).

23.116 Material Model 116: Composite Layup

This material is for modeling the elastic responses of composite lay-ups that have an arbitrary number of layers through the shell thickness. A pre-integration is used to compute the extensional, bending, and coupling stiffness for use with the Belytschko-Tsay resultant shell formulation. The angles of the local material axes are specified from layer to layer in the *SECTION_SHELL input. This material model must be used with the user defined integration rule for shells, which allows the elastic constants to change from integration point to integration point. Since the stresses are not computed in the resultant formulation, the stress output to the binary databases for the resultant elements are zero.

This material law is based on standard composite lay-up theory. The implementation, [Jones 1975], allows the calculation of the force, \mathbf{N} , and moment, \mathbf{M} , stress resultants from:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{21} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_z^0 \end{Bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{21} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix}, \quad (23.116.1)$$

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{16} \\ B_{21} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x^0 \\ \varepsilon_y^0 \\ \varepsilon_z^0 \end{Bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{21} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix}, \quad (23.116.2)$$

where A_{ij} is the extensional stiffness, D_{ij} is the bending stiffness, and B_{ij} is the coupling stiffness, which is a null matrix for symmetric lay-ups. The mid-surface strains and curvatures are denoted by ε_{ij}^0 and κ_{ij} , respectively. Since these stiffness matrices are symmetric, 18 terms are needed per shell element in addition to the shell resultants, which are integrated in time. This is considerably less storage than would typically be required with through thickness integration which requires a minimum of eight history variables per integration point, e.g., if 100 layers are used 800 history variables would be stored. Not only is memory much less for this model, but the CPU time required is also considerably reduced.

23.117 Material Model 117-118: Composite Matrix

This material is used for modeling the elastic responses of composites where pre-integration, which is done outside of LS-DYNA unlike the lay-up option above, is used to compute the extensional, bending, and coupling stiffness coefficients for use with the Belytschko-Tsay and the assumed strain resultant shell formulations. Since the stresses are not computed in the resultant formulation, the stresses output to the binary databases for the resultant elements are zero.

The calculation of the force, N_{ij} , and moment, M_{ij} , stress resultants is given in terms of the membrane strains, ϵ_i^0 , and shell curvatures, κ_i , as:

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \epsilon_z^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix}, \quad (23.117.1)$$

where $C_{ij} = C_{ji}$. In this model this symmetric matrix is transformed into the element local system and the coefficients are stored as element history variables.

In a variation of this model, *MAT_COMPOSITE_DIRECT, the resultants are already assumed to be given in the element local system which reduces the storage since the 21 coefficients are not stored as history variables as part of the element data. The shell thickness is built into the coefficient matrix and, consequently, within the part ID, which references this material ID, the thickness must be uniform.

23.119 Material Model 119: General Nonlinear 6DOF Discrete Beam

Catastrophic failure, which is based on displacement resultants, occurs if either of the following inequalities are satisfied:

$$\left(\frac{u_r}{u_r^{tfail}}\right)^2 + \left(\frac{u_s}{u_s^{tfail}}\right)^2 + \left(\frac{u_t}{u_t^{tfail}}\right)^2 + \left(\frac{\theta_r}{\theta_r^{tfail}}\right)^2 + \left(\frac{\theta_s}{\theta_s^{tfail}}\right)^2 + \left(\frac{\theta_t}{\theta_t^{tfail}}\right)^2 - 1 \geq 0, \quad (23.119.1)$$

$$\left(\frac{u_r}{u_r^{cfail}}\right)^2 + \left(\frac{u_s}{u_s^{cfail}}\right)^2 + \left(\frac{u_t}{u_t^{cfail}}\right)^2 + \left(\frac{\theta_r}{\theta_r^{cfail}}\right)^2 + \left(\frac{\theta_s}{\theta_s^{cfail}}\right)^2 + \left(\frac{\theta_t}{\theta_t^{cfail}}\right)^2 - 1 \geq 0. \quad (23.119.2)$$

After failure the discrete element is deleted. If failure is included either the tension failure or the compression failure or both may be used.

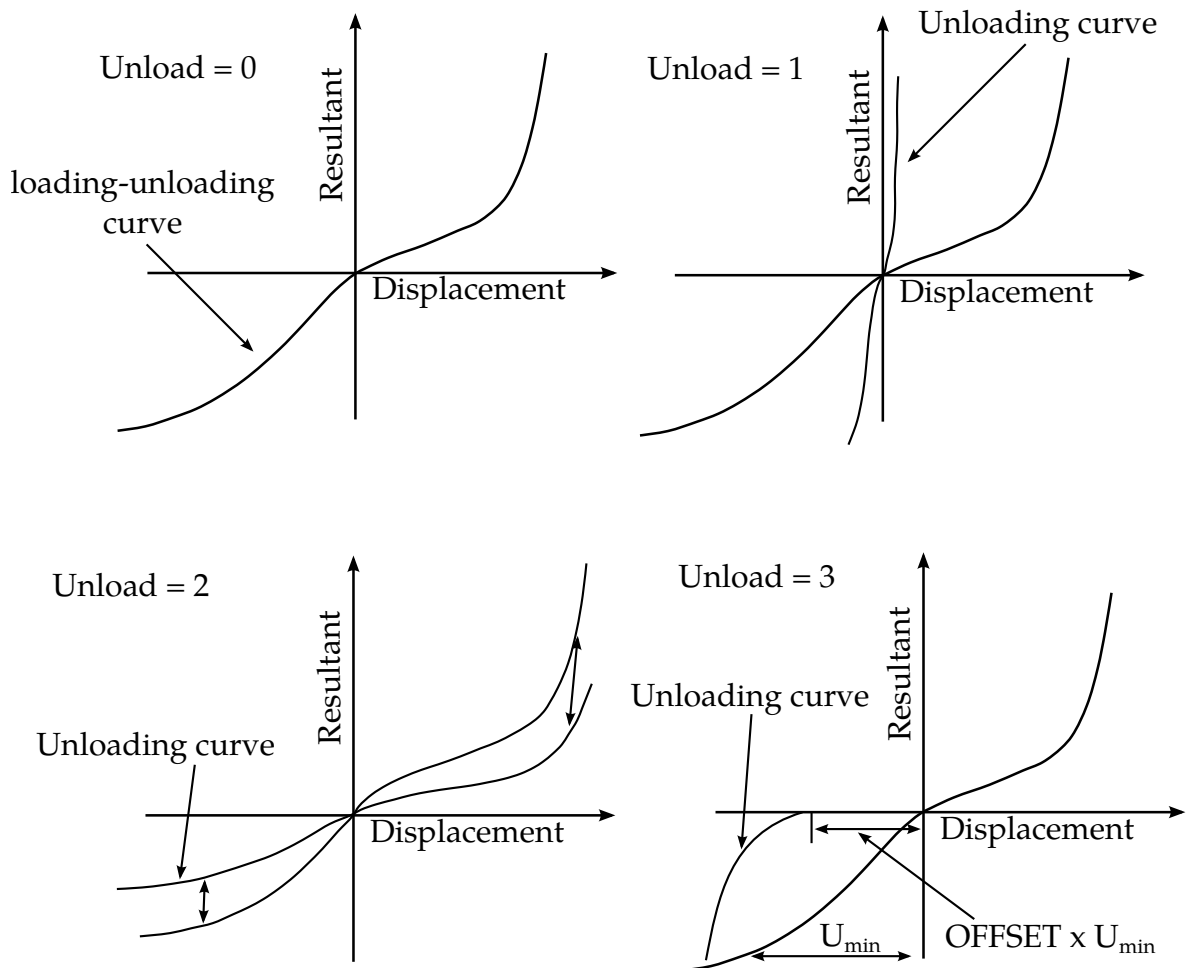


Figure 23.119.1. Load and unloading behavior.

23.120 Material Model 120: Gurson

The Gurson flow function is defined as:

$$\Phi = \frac{\sigma_M^2}{\sigma_Y^2} + 2q_1 f^* \cosh\left(\frac{3q_2 \sigma_H}{2\sigma_Y}\right) - 1 - (q_1 f^*)^2 = 0, \quad (23.120.1)$$

where σ_M is the equivalent von Mises stress, σ_Y is the Yield stress, σ_H is the mean hydrostatic stress. The effective void volume fraction is defined as

$$f^*(f) = \begin{cases} f & f \leq f_c \\ f_c + \frac{1/q_1 - f_c}{f_F - f_c} (f - f_c) & f > f_c \end{cases}. \quad (23.120.2)$$

The growth of void volume fraction is defined as

$$\dot{f} = \dot{f}_G + \dot{f}_N, \quad (23.120.3)$$

where the growth of existing voids is defined as

$$\dot{f}_G = (1 - f) \dot{\epsilon}_{kk}^p, \quad (23.120.4)$$

and the nucleation of new voids is defined as

$$\dot{f}_N = A \dot{\epsilon}_p, \quad (23.120.5)$$

where

$$A = \frac{f_N}{S_N \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{\epsilon_p - \epsilon_N}{S_N}\right)^2\right). \quad (23.120.6)$$

23.120 Material Model 120: Gurson RCDC

The Rc-Dc model is defined as follows. The damage D is given by

$$D = \int \omega_1 \omega_2 d\varepsilon^P, \quad (23.120.1)$$

where ε^P is the equivalent plastic strain,

$$\omega_1 = \left(\frac{1}{1 - \gamma \sigma_m} \right)^\alpha, \quad (23.120.2)$$

is the triaxial stress weighting term and

$$\omega_2 = (2 - A_D)^\beta, \quad (23.120.3)$$

is the asymmetric strain weighting term. In the above σ_m is the mean stress and

$$A_D = \min \left(\left| \frac{S_2}{S_3} \right|, \left| \frac{S_3}{S_2} \right| \right). \quad (23.120.4)$$

Fracture is initiated when the accumulation of damage satisfies

$$\frac{D}{D_c} > 1, \quad (23.120.5)$$

where D_c is the critical damage given by

$$D_c = D_0 (1 + b |\nabla D|^\lambda). \quad (23.120.6)$$

23.124 Material Model 124: Tension-Compression Plasticity

This is an isotropic elastic-plastic material where a unique yield stress versus plastic strain curve can be defined for compression and tension. Failure can occur based on plastic strain or a minimum time step size. Rate effects are modeled by using the Cowper-Symonds strain rate model.

The stress-strain behavior follows one curve in compression and another in tension. The sign of the mean stress determines the state where a positive mean stress (i.e., a negative pressure) is indicative of tension. Two load curves, $f_t(p)$ and $f_c(p)$, are defined, which give the yield stress, σ_y , versus effective plastic strain for both the tension and compression regimes. The two pressure values, p_t and p_c , when exceeded, determine if the tension curve or the compressive curve is followed, respectively. If the pressure, p , falls between these two values, a weighted average of the two curves are used:

$$\text{if } -p_t \leq p \leq p_c \quad \begin{cases} \text{scale} = \frac{p_c - p}{p_c + p_t} \\ \sigma_y = \text{scale} \cdot f_t(p) + (1 - \text{scale}) \cdot f_c(p) \end{cases} \quad (23.124.1)$$

Strain rate is accounted for using the Cowper and Symonds model, which scales the yield stress with the factor

$$1 + \left(\frac{\dot{\epsilon}}{C} \right)^{1/p}, \quad (23.124.2)$$

where $\dot{\epsilon}$ is the strain rate $\dot{\epsilon} = \sqrt{\dot{\epsilon}_{ij}\dot{\epsilon}_{ij}}$.

23.126 Material Model 126: Metallic Honeycomb

For efficiency it is strongly recommended that the load curve ID's: LCA, LCB, LCC, LCS, LCAB, LCBC, and LCCA, contain exactly the same number of points with corresponding strain values on the abscissa. If this recommendation is followed the cost of the table lookup is insignificant. Conversely, the cost increases significantly if the abscissa strain values are not consistent between load curves.

The behavior before compaction is orthotropic where the components of the stress tensor are uncoupled, i.e., a component of strain will generate resistance in the local a -direction with no coupling to the local b and c directions. The elastic moduli vary from their initial values to the fully compacted values linearly with the relative volume:

$$\begin{aligned} E_{aa} &= E_{aau} + \beta_{aa}(E - E_{aau}), & G_{ab} &= G_{abu} + \beta(G - G_{abu}), \\ E_{bb} &= E_{bbu} + \beta_{bb}(E - E_{bbu}), & G_{bc} &= G_{bcu} + \beta(G - G_{bcu}), \\ E_{cc} &= E_{ccu} + \beta_{cc}(E - E_{ccu}), & G_{ca} &= G_{cau} + \beta(G - G_{cau}), \end{aligned} \quad (23.126.1)$$

where

$$\beta = \max \left[\min \left(\frac{1 - V}{1 - V_f}, 1 \right), 0 \right], \quad (23.126.2)$$

and G is the elastic shear modulus for the fully compacted honeycomb material

$$G = \frac{E}{2(1 + \nu)}. \quad (23.126.3)$$

The relative volume, ν , is defined as the ratio of the current volume over the initial volume, and typically, $\nu = 1$ at the beginning of a calculation.

The load curves define the magnitude of the stress as the material undergoes deformation. The first value in the curve should be less than or equal to zero corresponding to tension and increase to full compaction. **Care should be taken when defining the curves so the extrapolated values do not lead to negative yield stresses.**

At the beginning of the stress update we transform each element's stresses and strain rates into the local element coordinate system. For the uncompacted material, the trial stress components are updated using the elastic interpolated moduli according to:

$$\begin{aligned} \sigma_{aa}^{n+1\text{trial}} &= \sigma_{aa}^n + E_{aa}\Delta\epsilon_{aa}, & \sigma_{ab}^{n+1\text{trial}} &= \sigma_{ab}^n + 2G_{ab}\Delta\epsilon_{ab}, \\ \sigma_{bb}^{n+1\text{trial}} &= \sigma_{bb}^n + E_{bb}\Delta\epsilon_{bb}, & \sigma_{bc}^{n+1\text{trial}} &= \sigma_{bc}^n + 2G_{bc}\Delta\epsilon_{bc}, \\ \sigma_{cc}^{n+1\text{trial}} &= \sigma_{cc}^n + E_{cc}\Delta\epsilon_{cc}, & \sigma_{ca}^{n+1\text{trial}} &= \sigma_{ca}^n + 2G_{ca}\Delta\epsilon_{ca}, \end{aligned} \quad (23.126.4)$$

We then independently check each component of the updated stresses to ensure that they do not exceed the permissible values determined from the load curves, e.g., if

$$|\sigma_{ij}^{n+1\text{trial}}| > \lambda \sigma_{ij}(\varepsilon_{ij}), \quad (23.126.5)$$

then

$$\sigma_{ij}^{n+1} = \sigma_{ij}(\varepsilon_{ij}) \frac{\lambda \sigma_{ij}^{n+1\text{trial}}}{|\sigma_{ij}^{n+1\text{trial}}|}. \quad (23.126.6)$$

The components of $\sigma_{ij}(\varepsilon_{ij})$ are defined by load curves. The parameter λ is either unity or a value taken from the load curve number, LCSR, that defines λ as a function of strain-rate. Strain-rate is defined here as the Euclidean norm of the deviatoric strain-rate tensor.

For fully compacted material we assume that the material behavior is elastic-perfectly plastic and updated the stress components according to:

$$s_{ij}^{\text{trial}} = s_{ij}^n + 2G\Delta\varepsilon_{ij}^{\text{dev}^{n+1/2}}, \quad (23.126.7)$$

where the deviatoric strain increment is defined as

$$\Delta\varepsilon_{ij}^{\text{dev}} = \Delta\varepsilon_{ij} - \frac{1}{3}\Delta\varepsilon_{kk}\delta_{ij}. \quad (23.126.8)$$

We now check to see if the yield stress for the fully compacted material is exceeded by comparing

$$s_{\text{eff}}^{\text{trial}} = \left(\frac{3}{2} s_{ij}^{\text{trial}} s_{ij}^{\text{trial}} \right)^{1/2}, \quad (23.126.9)$$

the effective trial stress to the yield stress, σ_y . If the effective trial stress exceeds the yield stress, we simply scale back the stress components to the yield surface

$$s_{ij}^{n+1} = \frac{\sigma_y}{s_{\text{eff}}^{\text{trial}}} s_{ij}^{\text{trial}}. \quad (23.126.10)$$

We can now update the pressure using the elastic bulk modulus, K

$$p^{n+1} = p^n - K\Delta\varepsilon_{kk}^{n+1/2}, \quad (23.126.11)$$

$$K = \frac{E}{3(1-2\nu)},$$

and obtain the final value for the Cauchy stress

$$\sigma_{ij}^{n+1} = s_{ij}^{n+1} - p^{n+1}\delta_{ij}. \quad (23.126.12)$$

After completing the stress update we transform the stresses back to the global configuration.

23.126.1 Stress Update

If $LCA < 0$, a transversely anisotropic yield surface is obtained where the uniaxial limit stress, $\sigma^y(\varphi, \varepsilon^{vol})$, can be defined as a function of angle φ with the strong axis and volumetric strain, ε^{vol} . Elastically, the new material model is assumed to behave exactly as material 126 (with the restriction $E_{22u} = E_{33u}$ and $G_{12u} = G_{13u}$), see the LS-DYNA Keyword User's Manual [Hallquist 2003]. As for the plastic behavior, a natural question that arises is how to define the limit stress for a general multiaxial stress state that reduces to the uniaxial limit stress requirement when the stress is uniaxial. Having given it some thought, we feel that it is most convenient to work with the principal stresses and the corresponding directions to achieve this goal.

Assume that the elastic update results in a trial stress σ^{trial} in the material coordinate system. This stress tensor is diagonalized to obtain the principal stresses σ_1^{trial} , σ_2^{trial} and σ_3^{trial} and the corresponding principal directions \mathbf{q}_1 , \mathbf{q}_2 and \mathbf{q}_3 relative to the material coordinate system. The angle that each direction makes with the strong axis of anisotropy \mathbf{e}_1 is given by

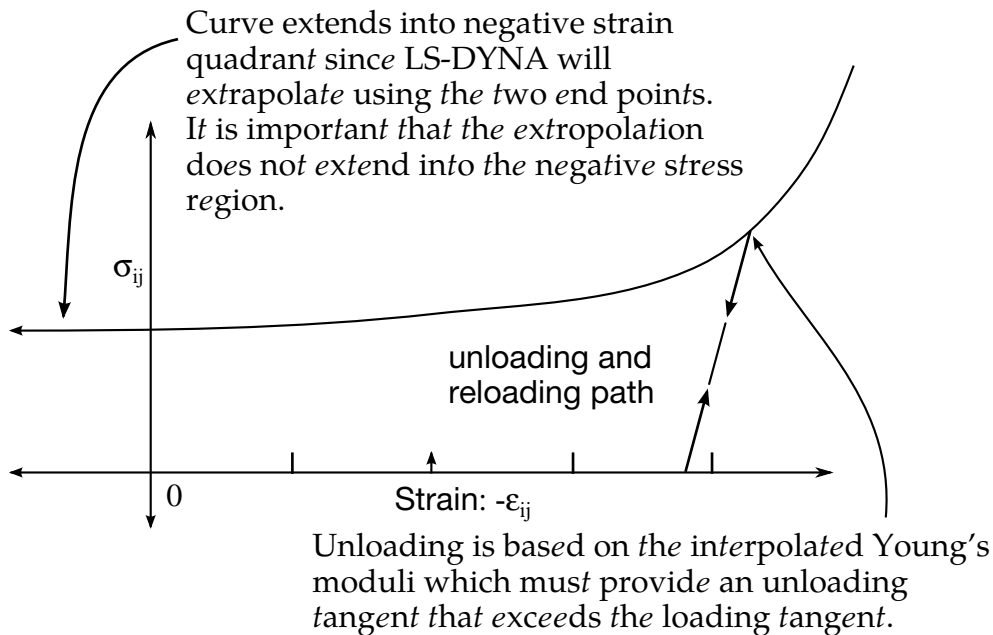


Figure 23.126.1. Stress quantity versus strain. Note that the “yield stress” at a strain of zero is nonzero. In the load curve definition the “time” value is the directional strain and the “function” value is the yield stress.

$$\varphi_i = \arccos|\mathbf{q}_i \cdot \mathbf{e}_1|, \quad i = 1, 2, 3. \quad (23.126.13)$$

Now a limit stress in the direction of a general multiaxial stress is determined as a convex combination of the uniaxial limit stress in each principal direction

$$\sigma^Y(\sigma^{\text{trial}}) = \frac{\sum_{j=1}^3 \sigma^Y(\phi_j, \varepsilon^{\text{vol}}) \sigma_j^{\text{trial}} \sigma_j^{\text{trial}}}{\sum_{k=1}^3 \sigma_k^{\text{trial}} \sigma_k^{\text{trial}}}, \quad (23.126.14)$$

Each of the principal stresses is updated as

$$\sigma_i = \sigma_i^{\text{trial}} \min \left(1, \frac{\sigma^Y(\sigma^{\text{trial}})}{\sqrt{\sum_{k=1}^3 \sigma_k^{\text{trial}} \sigma_k^{\text{trial}}}} \right), \quad (23.126.15)$$

and the new stress is transformed back to the material coordinate system⁴.

This stress update is not uniquely defined when the stress tensor possesses multiple eigenvalues, thus the following simple set of rules is applied. If all principal stresses are equal, one of the principal directions is chosen to coincide with the strong axis of anisotropy. If two principal stresses are equal, then one of the directions corresponding to this stress value is chosen perpendicular to the strong axis of anisotropy.

23.126.2 Support for Independent Shear and Hydrostatic Yield Stress Limit

The model just described turned out to be weak in shear [Okuda 2003] and there were no means of adding shear resistance without changing the behavior in pure uniaxial compression. We propose the following modification of the model where the user can prescribe the shear and hydrostatic resistance in the material without changing the uniaxial behavior.

Assume that the elastic update results in a trial stress σ^{trial} in the material coordinate system. This stress tensor is diagonalized to obtain the principal stresses σ_1^{trial} , σ_2^{trial} and σ_3^{trial} and the corresponding principal directions \mathbf{q}_1 , \mathbf{q}_2 and \mathbf{q}_3 relative to the material coordinate system. For this discussion we assume that the principal stress values are ordered so that $\sigma_1^{\text{trial}} \leq \sigma_2^{\text{trial}} \leq \sigma_3^{\text{trial}}$. Two cases need to be treated.

If $\sigma_1^{\text{trial}} \leq -\sigma_3^{\text{trial}}$ then the principal stress value with largest magnitude is σ_1^{trial} , and consequently $\sigma_1^{\text{trial}} \leq 0$. Let

⁴ Since each component of the stress tensor is scaled by the same factor in Equation 19.126.14, the stress is in practice not transformed back but the scaling is performed on the stress in the material coordinate system.

$$\begin{aligned}\sigma_u &= \sigma_1^{\text{trial}} + \max(|\sigma_2^{\text{trial}}|, |\sigma_3^{\text{trial}}|), \\ \sigma_p &= \frac{1}{3} \{-\max(|\sigma_2^{\text{trial}}|, |\sigma_3^{\text{trial}}|) + \sigma_2^{\text{trial}} + \sigma_3^{\text{trial}}\},\end{aligned}\quad (23.126.16)$$

and finally

$$\begin{aligned}\sigma_d^1 &= -\max(|\sigma_2^{\text{trial}}|, |\sigma_3^{\text{trial}}|) - \sigma_p \\ \sigma_d^2 &= \sigma_2^{\text{trial}} - \sigma_p, \\ \sigma_d^3 &= \sigma_3^{\text{trial}} - \sigma_p.\end{aligned}\quad (23.126.17)$$

The total stress is the sum of a uniaxial stress represented by σ_u , a hydrostatic stress represented by σ_p and a deviatoric stress represented by σ_d^1 , σ_d^2 and σ_d^3 . The angle that the direction of σ_u makes with the strong axis of anisotropy \mathbf{e}_1 is given by $\phi = \arccos|\mathbf{q}_1 \cdot \mathbf{e}_1|$.

Now a limit stress for the general multiaxial stress is determined as a convex combination of the three stress contributions as follows

$$\begin{aligned}\sigma^Y(\sigma^{\text{trial}}) \\ = \frac{\sigma_u^Y(\phi, \varepsilon^{\text{vol}})\sigma_u^2 + 3\sqrt{3}\sigma_p^Y(\varepsilon^{\text{vol}})\sigma_p^2 + \sqrt{2}\sigma_d^Y(\varepsilon^{\text{vol}})\{(\sigma_d^1)^2 + (\sigma_d^2)^2 + (\sigma_d^3)^2\}}{\sigma_u^2 + 3\sigma_p^2 + (\sigma_d^1)^2 + (\sigma_d^2)^2 + (\sigma_d^3)^2}.\end{aligned}\quad (23.126.18)$$

Here $\sigma_u^Y(\phi, \varepsilon^{\text{vol}})$ is the prescribed uniaxial stress limit, $\sigma_p^Y(\varepsilon^{\text{vol}})$ is the hydrostatic stress limit and $\sigma_d^Y(\varepsilon^{\text{vol}})$ is the stress limit in simple shear. The input for the first of these is exactly as for the old model. The other two functions are for now written

$$\begin{aligned}\sigma_p^Y(\varepsilon^{\text{vol}}) &= \sigma_p^Y + \sigma^S(\varepsilon^{\text{vol}}), \\ \sigma_d^Y(\varepsilon^{\text{vol}}) &= \sigma_d^Y + \sigma^S(\varepsilon^{\text{vol}}),\end{aligned}\quad (23.126.19)$$

where σ_p^Y and σ_d^Y are user prescribed constant stress limits and σ^S is the function describing the densification of the material when loaded in the direction of the strong material axis. We use the keyword parameters ECCU and GCAU for the new input as follows.

- ECCU σ_d^Y , average stress limit (yield) in simple shear.
- GCAU σ_p^Y , average stress limit (yield) in hydrostatic compression (pressure).

Both of these parameters should be positive.

Each of the principal stresses is updated as

$$\sigma_i = \sigma_i^{\text{trial}} \min \left(1, \frac{\sigma^Y(\sigma^{\text{trial}})}{\sqrt{\sum_{k=1}^3 \sigma_k^{\text{trial}} \sigma_k^{\text{trial}}}} \right), \quad (23.126.20)$$

and the new stress is transformed back to the material coordinate system.

If $\sigma_3^{\text{trial}} \geq -\sigma_1^{\text{trial}}$ then the principal stress value with largest magnitude is σ_3^{trial} and consequently $\sigma_3^{\text{trial}} \geq 0$. Let

$$\begin{aligned} \sigma_u &= \sigma_3^{\text{trial}} - \max(|\sigma_2^{\text{trial}}|, |\sigma_1^{\text{trial}}|), \\ \sigma_p &= \frac{1}{3} \{ \max(|\sigma_2^{\text{trial}}|, |\sigma_1^{\text{trial}}|) + \sigma_2^{\text{trial}} + \sigma_1^{\text{trial}} \}, \end{aligned} \quad (23.126.21)$$

and finally

$$\begin{aligned} \sigma_d^1 &= \sigma_1^{\text{trial}} - \sigma_p, \\ \sigma_d^2 &= \sigma_2^{\text{trial}} - \sigma_p, \\ \sigma_d^3 &= \max(|\sigma_2^{\text{trial}}|, |\sigma_1^{\text{trial}}|) - \sigma_p. \end{aligned} \quad (23.126.22)$$

The angle that the direction of σ_u makes with the strong axis of anisotropy \mathbf{e}_1 is given by $\phi = \arccos|\mathbf{q}_3 \cdot \mathbf{e}_1|$. The rest of the treatment is the same as for the case when $\sigma_1^{\text{trial}} \leq -\sigma_3^{\text{trial}}$. To motivate the model, let us consider three states of stress.

1. For a uniaxial stress σ , we have $\sigma_u = \sigma$ and $\sigma_p = \sigma_d^1 = \sigma_d^2 = \sigma_d^3 = 0$. This leads us to $\sigma^Y(\sigma^{\text{trial}}) = \sigma_u^Y(\phi, \varepsilon^{\text{vol}})$ and hence the stress level will be limited by the user prescribed uniaxial stress limit.
2. For a simple shear σ , we have $\sigma_d^1 = -\sigma_d^3 = -\sigma$ and $\sigma_p = \sigma_d^2 = \sigma_u = 0$. Hence $\sigma^Y(\sigma^{\text{trial}}) = \sqrt{2}\sigma_d^Y(\varepsilon^{\text{vol}})$ and the stress level will be limited by the user prescribed shear stress limit.
3. For a pressure σ , we have $\sigma_p = \sigma$ and $\sigma_u = \sigma_d^1 = \sigma_d^2 = \sigma_d^3 = 0$. Hence $\sigma^Y(\sigma^{\text{trial}}) = \sqrt{3}\sigma_p^Y(\varepsilon^{\text{vol}})$ and the stress level will be limited by the user prescribed hydrostatic stress limit.

23.127 Material Model 127: Arruda-Boyce Hyperviscoelastic Rubber

This material model, described in the paper by Arruda and Boyce [1993], provides a rubber model that is optionally combined with linear viscoelasticity. Rubber is generally considered to be fully incompressible since the bulk modulus greatly exceeds the shear modulus in magnitude; therefore, to model the rubber as an unconstrained material, a hydrostatic work term, $W_H(J)$, is included in the strain energy functional which is a function of the relative volume, J , [Ogden, 1984]:

$$W(J_1, J) = G \left[\frac{1}{2} (J_1 - 3) + \frac{1}{20N} (J_1^2 - 9) + \frac{11}{1050N^2} (J_1^3 - 27) \right] + G \left[\frac{19}{7000N^3} (J_1^4 - 81) + \frac{519}{673750N^4} (J_1^5 - 243) \right] + W_H(J), \quad (23.127.1)$$

G is the shear modulus, and N is the number of statistical links. The hydrostatic work term is expressed in terms of the bulk modulus, K , and J as:

$$W_H(J) = \frac{K}{2} (J - 1)^2. \quad (23.127.2)$$

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.127.3)$$

or in terms of the second Piola-Kirchhoff stress, S_{ij} , and Green's strain tensor, E_{ij} ,

$$S_{ij} = \int_0^t G_{ijkl}(t - \tau) \frac{\partial E_{kl}}{\partial \tau} d\tau, \quad (23.127.4)$$

where $g_{ijkl}(t - \tau)$ and $G_{ijkl}(t - \tau)$ are the relaxation functions for the different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

If we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^N \alpha_m e^{-\beta_m t}, \quad (23.127.5)$$

given by,

$$g(t) = \sum_{i=1}^n G_i e^{-\beta_i t}. \quad (23.127.6)$$

This model is effectively a Maxwell fluid which consists of a dampers and springs in series. We characterize this in the input by shear moduli, G_i , and decay constants, β_i . The viscoelastic behavior is optional and an arbitrary number of terms may be used.

23.128 Material Model 128: Heart Tissue

This material model provides a tissue model described in the paper by Guccione, McCulloch, and Waldman [1991]

The tissue model is described in terms of the energy functional in terms of the Green strain components, E_{ij} ,

$$\begin{aligned} W(E, J) &= \frac{C}{2} (e^Q - 1) + W_H(J), \\ Q &= b_1 E_{11}^2 + b_2 (E_{22}^2 + E_{33}^2 + E_{23}^2 + E_{32}^2) + b_3 (E_{12}^2 + E_{21}^2 + E_{13}^2 + E_{31}^2), \end{aligned} \quad (23.128.1)$$

where the hydrostatic work term is in terms of the bulk modulus, K , and the third invariant J , as:

$$W_H(J) = \frac{K}{2} (J - 1)^2. \quad (23.128.2)$$

The Green components are modified to eliminate any effects of volumetric work following the procedures of Ogden.

23.129 Material Model 129: Isotropic Lung Tissue

This material model provides a lung tissue model described in the paper by Vawter [1980].

The material is described by a strain energy functional expressed in terms of the invariants of the Green Strain:

$$W(I_1, I_2) = \frac{C}{2\Delta} e^{(\alpha I_1^2 + \beta I_2)} + \frac{12C_1}{\Delta(1 + C_2)} [A^{(1+C_2)} - 1], \quad (23.129.1)$$

$$A^2 = \frac{4}{3}(I_1 + I_2) - 1,$$

where the hydrostatic work term is in terms of the bulk modulus, K , and the third invariant J , as:

$$W_H(J) = \frac{K}{2}(J - 1)^2, \quad (23.129.2)$$

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.129.3)$$

or in terms of the second Piola-Kirchhoff stress, S_{ij} , and Green's strain tensor, E_{ij} ,

$$S_{ij} = \int_0^t G_{ijkl}(t - \tau) \frac{\partial E_{kl}}{\partial \tau} d\tau, \quad (23.129.4)$$

where $g_{ijkl}(t - \tau)$ and $G_{ijkl}(t - \tau)$ are the relaxation functions for the different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

If we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \alpha_0 + \sum_{m=1}^N \alpha_m e^{-\beta_m t}, \quad (23.129.5)$$

given by,

$$g(t) = \sum_{i=1}^n G_i e^{-\beta_i t}. \quad (23.129.6)$$

This model is effectively a Maxwell fluid which consists of a dampers and springs in series. We characterize this in the input by shear moduli, G_i , and decay constants, β_i . The viscoelastic behavior is optional and an arbitrary number of terms may be used.

23.130 Material Model 130: Special Orthotropic

The in-plane elastic matrix for in-plane plane stress behavior is given by:

$$\mathbf{C}_{\text{in plane}} = \begin{bmatrix} Q_{11p} & Q_{12p} & 0 & 0 & 0 \\ Q_{12p} & Q_{22p} & 0 & 0 & 0 \\ 0 & 0 & Q_{44p} & 0 & 0 \\ 0 & 0 & 0 & Q_{55p} & 0 \\ 0 & 0 & 0 & 0 & Q_{66p} \end{bmatrix}, \quad (23.130.1)$$

where the terms Q_{ijp} are defined as:

$$\begin{aligned} Q_{11p} &= \frac{E_{11p}}{1 - \nu_{12p}\nu_{21p}}, \\ Q_{22p} &= \frac{E_{22p}}{1 - \nu_{12p}\nu_{21p}}, \\ Q_{12p} &= \frac{\nu_{12p}E_{11p}}{1 - \nu_{12p}\nu_{21p}}, \\ Q_{44p} &= G_{12p}, \\ Q_{55p} &= G_{23p}, \\ Q_{66p} &= G_{31p}. \end{aligned} \quad (23.130.2)$$

The elastic matrix for bending behavior is given by:

$$\mathbf{C}_{\text{bending}} = \begin{bmatrix} Q_{11b} & Q_{12b} & 0 \\ Q_{12b} & Q_{22b} & 0 \\ 0 & 0 & Q_{44b} \end{bmatrix}, \quad (23.130.3)$$

where the terms Q_{ijb} are similarly defined.

23.131 Material Model 131: Isotropic Smeared Crack

The following documentation is taken nearly verbatim from the documentation of that by Lemmen and Meijer [2001].

Three methods are offered to model progressive failure. The maximum principal stress criterion detects failure if the maximum (most tensile) principal stress exceeds σ_{\max} . Upon failure, the material can no longer carry stress.

The second failure model is the smeared crack model with linear softening stress-strain curve using equivalent uniaxial strains. Failure is assumed to be perpendicular to the principal strain directions. A rotational crack concept is employed in which the crack directions are related to the current directions of principal strain. Therefore crack directions may rotate in time. Principal stresses are expressed as

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} = \begin{bmatrix} \bar{E}_1 & 0 & 0 \\ 0 & \bar{E}_2 & 0 \\ 0 & 0 & \bar{E}_3 \end{bmatrix} \begin{pmatrix} \tilde{\epsilon}_1 \\ \tilde{\epsilon}_2 \\ \tilde{\epsilon}_3 \end{pmatrix} = \begin{pmatrix} \bar{E}_1 \tilde{\epsilon}_1 \\ \bar{E}_2 \tilde{\epsilon}_2 \\ \bar{E}_3 \tilde{\epsilon}_3 \end{pmatrix}, \quad (23.131.1)$$

with \bar{E}_1 , \bar{E}_2 and \bar{E}_3 being secant stiffness in the terms that depend on internal variables.

In the model developed for DYCOSS it has been assumed that there is no interaction between the three directions in which case stresses simply follow

$$\sigma_j(\tilde{\epsilon}_j) = \begin{cases} E\tilde{\epsilon}_j & \text{if } 0 \leq \tilde{\epsilon}_j \leq \tilde{\epsilon}_{j,\text{ini}} \\ \left(1 - \frac{\tilde{\epsilon}_j - \tilde{\epsilon}_{j,\text{ini}}}{\tilde{\epsilon}_{j,\text{ult}} - \tilde{\epsilon}_{j,\text{ini}}}\right) \bar{\sigma} & \text{if } \tilde{\epsilon}_{j,\text{ini}} < \tilde{\epsilon}_j \leq \tilde{\epsilon}_{j,\text{ult}}, \\ 0 & \text{if } \tilde{\epsilon}_j > \tilde{\epsilon}_{j,\text{ult}} \end{cases} \quad (23.131.2)$$

with $\bar{\sigma}$ the ultimate stress, $\tilde{\epsilon}_{j,\text{ini}}$ the damage threshold, and $\tilde{\epsilon}_{j,\text{ult}}$ the ultimate strain in j -direction. The damage threshold is defined as

$$\tilde{\epsilon}_{j,\text{ini}} = \frac{\bar{\sigma}}{E}. \quad (23.131.3)$$

The ultimate strain is obtained by relating the crack growth energy and the dissipated energy

$$\int \int \bar{\sigma} d\tilde{\epsilon}_{j,\text{ult}} dV = GA, \quad (23.131.4)$$

with G the energy release rate, V the element volume and A the area perpendicular to the principal strain direction. The one-point elements in LS-DYNA have a single integration

point and the integral over the volume may be replaced by the volume. For linear softening it follows

$$\tilde{\varepsilon}_{j,\text{ult}} = \frac{2GA}{V\bar{\sigma}}, \quad (23.131.5)$$

The above formulation may be regarded as a damage equivalent to the maximum principle stress criterion.

The third model is a damage model presented by Brekelmans et. al. [1991]. Here the Cauchy stress tensor σ is expressed as

$$\sigma = (1 - D)E\varepsilon, \quad (23.131.6)$$

where D represents the current damage and the factor $(1 - D)$ is the reduction factor caused by damage. The scalar damage variable is expressed as a function of a so-called damage equivalent strain ε_d

$$D = D(\varepsilon_d) = 1 - \frac{\varepsilon_{\text{ini}}(\varepsilon_{\text{ult}} - \varepsilon_d)}{\varepsilon_d(\varepsilon_{\text{ult}} - \varepsilon_{\text{ini}})}, \quad (23.131.7)$$

with ultimate and initial strains as defined by

$$\varepsilon_d = \frac{k-1}{2k(1-2\nu)}J_1 + \frac{1}{2k} \sqrt{\left(\frac{k-1}{1-2\nu}J_1\right)^2 + \frac{6k}{(1+\nu)^2}J_2}, \quad (23.131.8)$$

where the constant k represents the ratio of the strength in tension over the strength in compression

$$k = \frac{\sigma_{\text{ult,tension}}}{\sigma_{\text{ult,compression}}}, \quad (23.131.9)$$

J_1 and J_2 are the first and the second invariants of the strain tensor representing the volumetric and the deviatoric straining, respectively

$$\begin{aligned} J_1 &= \varepsilon_{ij}, \\ J_2 &= \varepsilon'_{ij}\varepsilon'_{ij}, \end{aligned} \quad (23.131.10)$$

where ε'_{ij} denotes the deviatoric part of the strain tensor.

If the compression and tension strength are equal the dependency on the volumetric strain vanishes in Equation (23.131.8) and failure is shear dominated. If the compressive strength is much larger than the strength in tension, k becomes large and the J_1 terms in Equation (23.131.8) dominate the behavior.

23.133 Material Model 133: Barlat_YLD2000

The yield condition for this material can be written

$$f(\sigma, \alpha, \varepsilon_p) = \sigma_{\text{eff}}(\sigma_{xx} - 2\alpha_{xx} - \alpha_{yy}, \sigma_{yy} - 2\alpha_{yy} - \alpha_{xx}, \sigma_{xy} - \alpha_{xy}) - \sigma_Y^t(\varepsilon_p, \dot{\varepsilon}_p, \beta) \leq 0, \quad (23.133.1)$$

where

$$\begin{aligned} \sigma_{\text{eff}}(s_{xx}, s_{yy}, s_{xy}) &= \left(\frac{1}{2} (\varphi' + \varphi'') \right)^{\frac{1}{a}}, \\ \varphi' &= |X'_1 - X'_2|^a, \\ \varphi'' &= |2X''_1 + X''_2|^a + |X''_1 + 2X''_2|^a. \end{aligned} \quad (23.133.2)$$

The X'_i and X''_i are the eigenvalues of X'_{ij} and X''_{ij} and are given by

$$\begin{aligned} X'_1 &= \frac{1}{2} \left(X'_{11} + X'_{22} + \sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}} \right), \\ X'_2 &= \frac{1}{2} \left(X'_{11} + X'_{22} - \sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}} \right), \end{aligned} \quad (23.133.3)$$

and

$$\begin{aligned} X''_1 &= \frac{1}{2} \left(X''_{11} + X''_{22} + \sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}} \right), \\ X''_2 &= \frac{1}{2} \left(X''_{11} + X''_{22} - \sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}} \right), \end{aligned} \quad (23.133.4)$$

respectively. The X'_{ij} and X''_{ij} are given by

$$\begin{aligned} \begin{pmatrix} X'_{11} \\ X'_{22} \\ X'_{12} \end{pmatrix} &= \begin{pmatrix} L'_{11} & L'_{12} & 0 \\ L'_{21} & L'_{22} & 0 \\ 0 & 0 & L'_{33} \end{pmatrix} \begin{pmatrix} s_{xx} \\ s_{yy} \\ s_{xy} \end{pmatrix}, \\ \begin{pmatrix} X''_{11} \\ X''_{22} \\ X''_{12} \end{pmatrix} &= \begin{pmatrix} L''_{11} & L''_{12} & 0 \\ L''_{21} & L''_{22} & 0 \\ 0 & 0 & L''_{33} \end{pmatrix} \begin{pmatrix} s_{xx} \\ s_{yy} \\ s_{xy} \end{pmatrix}, \end{aligned} \quad (23.133.5)$$

where

$$(23.133.6)$$

$$\begin{pmatrix} L''_{11} \\ L''_{12} \\ L''_{21} \\ L''_{22} \\ L''_{33} \end{pmatrix} = \frac{1}{9} \begin{pmatrix} -2 & 2 & 8 & -2 & 0 \\ 1 & -4 & -4 & 4 & 0 \\ 4 & -4 & -4 & 1 & 0 \\ -2 & 8 & 2 & -2 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{pmatrix} \begin{pmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_8 \end{pmatrix}.$$

where α_1 to α_8 are the parameters that determine the shape of the yield surface.

The yield stress is expressed as

$$\sigma_Y^t(\varepsilon_p, \dot{\varepsilon}_p, \beta) = \sigma_Y^v(\varepsilon_p, \dot{\varepsilon}_p) + \beta(\sigma_0 - \sigma_Y^v(\varepsilon_p, \dot{\varepsilon}_p)), \quad (23.133.7)$$

where β determines the fraction kinematic hardening and σ_0 is the initial yield stress. The yield stress for purely isotropic hardening is given by

$$\sigma_Y^v(\varepsilon_p, \dot{\varepsilon}_p) = \sigma_Y(\varepsilon_p) \left(1 + \left\{ \frac{\dot{\varepsilon}_p}{C} \right\}^{\frac{1}{p}} \right), \quad (23.133.8)$$

where C and p are the Cowper-Symonds material parameters for strain rate effects.

The evolution of back stress is given by

$$\dot{\alpha} = \dot{\lambda} \beta \frac{\left(\frac{\partial \sigma_Y}{\partial \varepsilon_p} \left(1 + \left\{ \frac{\dot{\varepsilon}_p}{C} \right\}^{1/p} \right) + \sigma_Y \frac{1}{p C \Delta t} \left\{ \frac{\dot{\varepsilon}_p}{C} \right\}^{1/p-1} \right) \frac{\partial \sigma_{\text{eff}}}{\partial s}}{\frac{\partial \sigma_{\text{eff}}}{\partial s} \cdot \frac{\partial \sigma_{\text{eff}}}{\partial s}}, \quad (23.133.9)$$

where Δt is the current time step size and $\dot{\lambda}$ is the rate of plastic strain multiplier.

For the plastic return algorithms, the gradient of effective stress is computed as

$$\begin{pmatrix} \frac{\partial \sigma_{\text{eff}}}{\partial s_{xx}} \\ \frac{\partial \sigma_{\text{eff}}}{\partial s_{yy}} \\ \frac{\partial \sigma_{\text{eff}}}{\partial s_{xy}} \end{pmatrix} = \frac{a \sigma_{\text{eff}}^{a-1}}{2} \begin{pmatrix} L'_{11} & L'_{21} & 0 \\ L'_{12} & L'_{22} & 0 \\ 0 & 0 & L'_{33} \end{pmatrix} \begin{pmatrix} \frac{\partial \phi'}{\partial X'_{11}} \\ \frac{\partial \phi'}{\partial X'_{22}} \\ \frac{\partial \phi'}{\partial X'_{12}} \end{pmatrix} + \frac{a \sigma_{\text{eff}}^{a-1}}{2} \begin{pmatrix} L''_{11} & L''_{21} & 0 \\ L''_{12} & L''_{22} & 0 \\ 0 & 0 & L''_{33} \end{pmatrix} \begin{pmatrix} \frac{\partial \phi''}{\partial X''_{11}} \\ \frac{\partial \phi''}{\partial X''_{22}} \\ \frac{\partial \phi''}{\partial X''_{12}} \end{pmatrix}, \quad (23.133.10)$$

with the aid of

$$\frac{\partial \varphi'}{\partial X'_{ij}} = a(X'_1 - X'_2)^{a-1} \frac{\partial (X'_1 - X'_2)}{\partial X'_{ij}}, \quad (23.133.11)$$

$$\begin{aligned} \frac{\partial \varphi''}{\partial X''_{ij}} = & a|2X''_1 + X''_2|^{a-1} \operatorname{sgn}(2X''_1 + X''_2) \frac{\partial (2X''_1 + X''_2)}{\partial X''_{ij}} + \\ & a|2X''_2 + X''_1|^{a-1} \operatorname{sgn}(2X''_2 + X''_1) \frac{\partial (2X''_2 + X''_1)}{\partial X''_{ij}}, \end{aligned} \quad (23.133.12)$$

and

$$\frac{\partial (X'_1 - X'_2)}{\partial X'_{11}} = \frac{X'_{11} - X'_{22}}{\sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}}, \quad (23.133.13)$$

$$\frac{\partial (X'_1 - X'_2)}{\partial X'_{22}} = \frac{X'_{22} - X'_{11}}{\sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}}, \quad (23.133.14)$$

$$\frac{\partial (X'_1 - X'_2)}{\partial X'_{12}} = \frac{4X'_{12}}{\sqrt{(X'_{11} - X'_{22})^2 + 4X'^2_{12}}}, \quad (23.133.15)$$

$$\frac{\partial (2X''_1 + X''_2)}{\partial X''_{11}} = \frac{3}{2} + \frac{1}{2} \frac{X''_{11} - X''_{22}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}, \quad (23.133.16)$$

$$\frac{\partial (2X''_1 + X''_2)}{\partial X''_{22}} = \frac{3}{2} + \frac{1}{2} \frac{X''_{22} - X''_{11}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}, \quad (23.133.17)$$

$$\frac{\partial (2X''_1 + X''_2)}{\partial X''_{12}} = \frac{2X''_{12}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}, \quad (23.133.18)$$

$$\frac{\partial (2X''_2 + X''_1)}{\partial X''_{11}} = \frac{3}{2} - \frac{1}{2} \frac{X''_{11} - X''_{22}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}, \quad (23.133.19)$$

$$\frac{\partial (2X''_2 + X''_1)}{\partial X''_{22}} = \frac{3}{2} - \frac{1}{2} \frac{X''_{22} - X''_{11}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}, \quad (23.133.20)$$

$$\frac{\partial (2X''_2 + X''_1)}{\partial X''_{12}} = -\frac{2X''_{12}}{\sqrt{(X''_{11} - X''_{22})^2 + 4X''^2_{12}}}. \quad (23.133.21)$$

The algorithm for the plane stress update as well as the formula for the tangent modulus is given in detail in Section 19.36.1 and is not repeated here.

23.133.1 Closest point projection algorithm

This section describes shortly the closest point projection algorithm that was implemented to improve accuracy, hence the implicit performance, of the model. The closest point projection comes down to solving the following system of equations

$$f_1 = t + \mathbf{A}\alpha + (\sigma_Y^t(\Delta\lambda) - \sigma_Y^t(0))h - \sigma^{\text{trial}}(\Delta\epsilon_{33}) + 2G\Delta\lambda D\nabla\sigma_{\text{eff}}(t) = 0, \quad (23.133.22)$$

$$f_2 = -\sigma_{\text{eff}}(t) + \sigma_Y^t(\Delta\lambda) = 0, \quad (23.133.23)$$

$$f_3 = \sigma_{33}^{\text{trial}}(\Delta\epsilon_{33}) + 2G\Delta\lambda(\nabla\sigma_{\text{eff}}^1(t) + \nabla\sigma_{\text{eff}}^2(t)) = 0, \quad (23.133.24)$$

where

$$h = \frac{\beta}{\nabla\sigma_{\text{eff}}(t)^T \mathbf{B} \nabla\sigma_{\text{eff}}(t)} \mathbf{B} \nabla\sigma_{\text{eff}}(t), \quad (23.133.25)$$

in terms of the unknown variables \mathbf{t} (stress), $\Delta\epsilon_{33}$ (thickness strain increment) and $\Delta\lambda$ (plastic strain increment). In the above

$$\mathbf{D} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0.5 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 2 & 1 & \\ 1 & 2 & \\ & & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 & 1 & \\ 1 & 2 & \\ & & 0.5 \end{bmatrix}, \quad H = \frac{\partial\sigma_Y^t}{\partial\epsilon_p}. \quad (23.133.26)$$

This system of equations is solved using a Newton method with an additional line search for robustness. Using the notation

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{t} \\ \Delta\lambda \\ \Delta\epsilon_{33} \end{bmatrix}, \quad (23.133.27)$$

a Newton step is completed as

$$x^+ = x^- - s \left(\frac{\partial f}{\partial x} \right)^{-1} \mathbf{f} \quad (23.133.28)$$

for a step size $s \leq 1$ chosen such that the norm of the objective function is decreasing. The gradient of the objective function is given by

$$\nabla f = \nabla f_{1-\beta} + \nabla f_{\beta} \quad (23.133.29)$$

where

$$\nabla f_{1-\beta} = \begin{bmatrix} \mathbf{I} + 2G\Delta\lambda D\nabla^2\sigma_{\text{eff}} & 2GD\nabla\sigma_{\text{eff}} & -C_3 \\ -(\nabla\sigma_{\text{eff}})^T & H & 0 \\ 2G\Delta\lambda \mathbf{e}^T \nabla^2\sigma_{\text{eff}} & 2G\mathbf{e}^T \nabla\sigma_{\text{eff}} & C_{33} \end{bmatrix} \quad (23.133.30)$$

$$\nabla f_{\beta} = \begin{bmatrix} \Delta \lambda H \frac{\partial h}{\partial t} & Hh & 0 \\ 0^T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (23.133.31)$$

and

$$\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad C_3 = \left(K - \frac{2G}{3} \right) \mathbf{e}, \quad C_{33} = \left(K + \frac{4G}{3} \right), \quad (23.133.32)$$

G and K stands for the shear and bulk modulus, respectively. This algorithm requires computation of the effective stress hessian. The derivation of this is quite straightforward but the expression for it is rather long and is hence omitted in this report.

23.134 Material Model 134: Viscoelastic Fabric

The viscoelastic fabric model is a variation on the general viscoelastic Material Model 76. This model is valid for 3 and 4 node membrane elements only and is strongly recommended for modeling isotropic viscoelastic fabrics where wrinkling may be a problem. For thin fabrics, buckling can result in an inability to support compressive stresses; thus, a flag is included for this option. If bending stresses are important, use a shell formulation with Model 76.

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.134.1)$$

where $g_{ijkl}(t - \tau)$ is the relaxation function.

If we wish to include only simple rate effects for the deviatoric stresses, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \sum_{m=1}^N G_m e^{-\beta_m t}. \quad (23.134.2)$$

We characterize this in the input by shear moduli, G_i , and decay constants, β_i . An arbitrary number of terms, up to 6, may be used when applying the viscoelastic model.

For volumetric relaxation, the relaxation function is also represented by the Prony series in terms of bulk moduli:

$$k(t) = \sum_{m=1}^N K_m e^{-\beta_{k_m} t}. \quad (23.134.3)$$

23.139 Material Model 139: Modified Force Limited

This material model is available for the Belytschko resultant beam element only. Plastic hinges form at the ends of the beam when the moment reaches the plastic moment. The plastic moment versus rotation relationship is specified by the user in the form of a load curve and scale factor. The points of the load curve are (plastic rotation in radians, plastic moment). Both quantities should be positive for all points, with the first point being (zero, initial plastic moment). Within this constraint any form of characteristic may be used, including flat or falling curves. Different load curves and scale factors may be specified at each node and about each of the local s and t axes.

Axial collapse occurs when the compressive axial load reaches the collapse load. Collapse load versus collapse deflection is specified in the form of a load curve. The points of the load curve are either (true strain, collapse force) or (change in length, collapse force). Both quantities should be entered as positive for all points, and will be interpreted as compressive. The first point should be (zero, initial collapse load).

The collapse load may vary with end moment as well as with deflections. In this case several load-deflection curves are defined, each corresponding to a different end moment. Each load curve should have the same number of points and the same deflection values. The end moment is defined as the average of the absolute moments at each end of the beam and is always positive.

Stiffness-proportional damping may be added using the damping factor λ . This is defined as follows:

$$\lambda = \frac{2 * \zeta}{\omega}, \quad (23.139.1)$$

where ζ is the damping factor at the reference frequency ω (in radians per second). For example if 1% damping at 2Hz is required

$$\lambda = \frac{2 * 0.01}{2\pi * 2} = 0.001592. \quad (23.139.2)$$

If damping is used, a small time step may be required. LS-DYNA does not check this so to avoid instability it may be necessary to control the timestep via a load curve. As a guide, the timestep required for any given element is multiplied by $0.3L/c\lambda$ when damping is present (L = element length, c = sound speed).

Moment Interaction

Plastic hinges can form due to the combined action of moments about the three axes. This facility is activated only when yield moments are defined in the material input. A hinge forms when the following condition is first satisfied.

$$\left(\frac{M_r}{M_{r\text{yield}}}\right)^2 + \left(\frac{M_s}{M_{s\text{yield}}}\right)^2 + \left(\frac{M_t}{M_{t\text{yield}}}\right)^2 \geq 1, \quad (23.139.3)$$

where,

M_r, M_s, M_t = current moments

$M_{r\text{yield}}, M_{s\text{yield}}, M_{t\text{yield}}$ = yield moments

Note that scale factors for hinge behavior defined in the input will also be applied to the yield moments: for example, $M_{s\text{yield}}$ in the above formula is given by the input yield moment about the local axis times the input scale factor for the local s-axis. For strain-softening characteristics, the yield moment should generally be set equal to the initial peak of the moment-rotation load curve.

On forming a hinge, upper limit moments are set. These are given by

$$M_{r\text{upper}} = \text{MAX} \left(M_r, \frac{M_{r\text{yield}}}{2} \right), \quad (23.139.4)$$

and similarly for M_s and M_t . Thereafter the plastic moments will be given by:

$$M_{rp} = \min(M_{r\text{curve}}, M_{r\text{upper}})$$

and similarly for s and t , where

M_{rp} = current plastic moment

$M_{r\text{curve}}$ = moment taken from load curve at the current rotation scaled according to the scale factor.

The effect of this is to provide an upper limit to the moment that can be generated; it represents the softening effect of local buckling at a hinge site. Thus if a member is bent about its local s -axis it will then be weaker in torsion and about its local t -axis. For moment-softening curves, the effect is to trim off the initial peak (although if the curves subsequently harden, the final hardening will also be trimmed off).

It is not possible to make the plastic moment vary with the current axial load, but it is possible to make hinge formation a function of axial load and subsequent plastic moment a function of the moment at the time the hinge formed. This is discussed in the next section.

Independent plastic hinge formation

In addition to the moment interaction equation, Cards 7 through 18 allow plastic hinges to form independently for the s -axis and t -axis at each end of the beam and also for the torsional axis. A plastic hinge is assumed to form if any component of the current moment exceeds the yield moment as defined by the yield moment vs. axial force curves input on cards 7 and 8. If any of the 5 curves is omitted, a hinge will not form for that component. The curves can be defined for both compressive and tensile axial forces. If the axial force falls outside the range of the curve, the first or last point in the curve will be used. A hinge forming for one component of moment does not affect the other components.

Upon forming a hinge, the magnitude of that component of moment will not be permitted to exceed the current plastic moment. The current plastic moment is obtained by interpolating between the plastic moment vs. plastic rotation curves input on cards 10, 12, 14, 16, or 18. Curves may be input for up to 8 hinge moments, where the hinge moment is defined as the yield moment at the time that the hinge formed. Curves must be input in order of increasing hinge moment and each curve should have the same plastic rotation values. The first or last curve will be used if the hinge moment falls outside the range of the curves. If no curves are defined, the plastic moment is obtained from the curves on cards 4 through 6. The plastic moment is scaled by the scale factors on lines 4 to 6.

A hinge will form if either the independent yield moment is exceeded or if the moment interaction equation is satisfied. If both are true, the plastic moment will be set to the minimum of the interpolated value and M_{rp} .

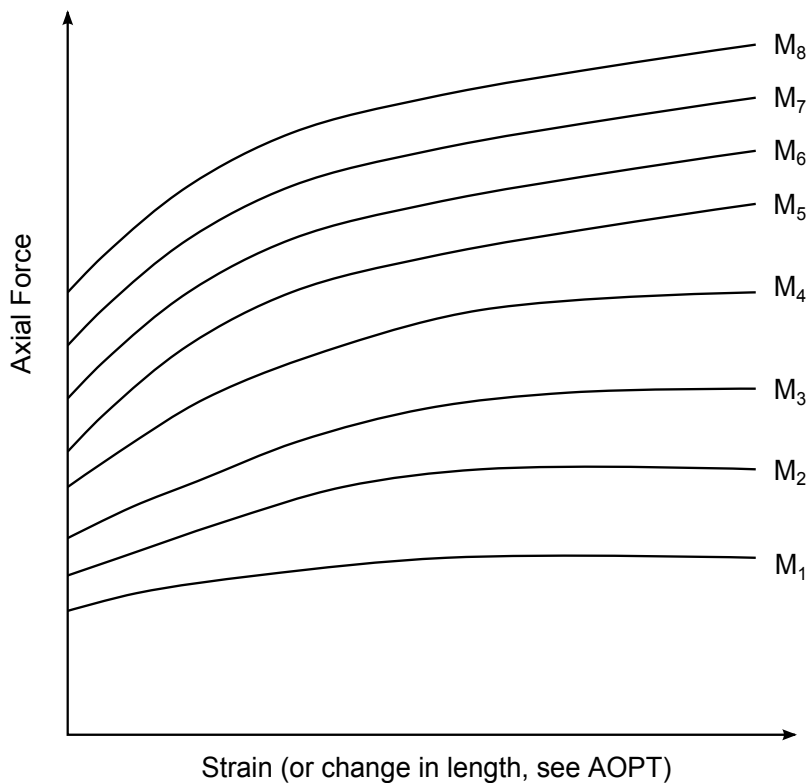


Figure 23.139.1. The force magnitude is limited by the applied end moment. For an intermediate value of the end moment LS-DYNA interpolates between the curves to determine the allowable force value.

23.141 Material Model 141: Rate Sensitive Polymer

$$\varepsilon_{ij} = D_o \exp \left[-\frac{1}{2} \left(\frac{Z_o^2}{3K_2} \right) \right] \left(\frac{S_{ij} - \Omega_{ij}}{\sqrt{K_2}} \right), \quad (23.141.1)$$

where D_o is the maximum inelastic strain rate, Z_o is the isotropic initial hardness of material, Ω_{ij} is the internal stress, S_{ij} is the deviatoric stress component, and K_2 is defined as follows:

$$K_2 = \frac{1}{2} (S_{ij} - \Omega_{ij})(S_{ij} - \Omega_{ij}), \quad (23.141.2)$$

and represent the second invariant of the overstress tensor. The elastic components of the strain are added to the inelastic strain to obtain the total strain. The following relationship defines the internal stress variable rate:

$$\dot{\Omega}_{ij} = \frac{2}{3} q \Omega_m \dot{\varepsilon}_{ij}^I - q \Omega_{ij} \dot{\varepsilon}_e^I, \quad (23.141.3)$$

where q is a material constant, Ω_m is a material constant that represents the maximum value of the internal stress, and $\dot{\varepsilon}_e^I$ is the effective inelastic strain.

23.142 Material Model 142: Transversely Isotropic Crushable Foam

A new material model for low density, transversely isotropic crushable foams, has been developed at DaimlerChrysler by Hirth, Du Bois, and Weimar. Hirth, Du Bois, and Weimar determined that material model 26, MAT_HONEYCOMB, which is commonly used to model foams, can systematically over estimate the stress when it is loaded off-axis. Their new material model overcomes this problem without requiring any additional input. Their new model can possibly replace the MAT_HONEYCOMB material, which is currently used in the frontal offset and side impact barriers.

Many polymers used for energy absorption are low density, crushable foams with no noticeable Poisson effect. Frequently manufactured by extrusion, they are transversely isotropic. This class of material is used to enhance automotive safety in low velocity (bumper impact) and medium velocity (interior head impact) applications. These materials require a transversely isotropic, elastoplastic material with a flow rule allowing for large permanent volumetric deformations.

The MAT_HONEYCOMB model uses a local coordinate system defined by the user. One of the axes of the local system coincides with the extrusion direction of the honeycomb in the undeformed configuration. As an element deforms, its local coordinate system rotates with its mean rigid body motion. Each of the six stress components is treated independently, and each has its own law relating its flow stress to its plastic strain.

The effect of off-axis loading on the MAT_HONEYCOMB model can be estimated by restricting our considerations to plane strain in two dimensions. Our discussion is restricted to the response of the foam before it becomes fully compacted. After compaction, its response is modeled with conventional J_2 plasticity. The model reduces to

$$\begin{aligned} |\sigma_{11}| &\leq \sigma_{11}^y(\varepsilon_V), \\ |\sigma_{22}| &\leq \sigma_{22}^y(\varepsilon_V), \\ |\sigma_{12}| &\leq \sigma_{12}^y(\varepsilon_V), \end{aligned} \quad (23.142.1)$$

where ε_V is the volumetric strain. For a fixed value of volumetric strain, the individual stress components respond in an elastic-perfectly plastic manner, i.e., the foam doesn't have any strain hardening.

In two dimensions, the stress tensor transforms according to

$$\begin{aligned} [\sigma] &= [R(\theta)]^T [\sigma_\theta] [R(\theta)], \\ [R(\theta)] &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \end{aligned} \quad (23.142.2)$$

where θ is the angle of the local coordinate system relative to the global system. For uniaxial loading along the global 1-axis, the stress will be (accounting for the sign of the volume strain),

$$\sigma_{11} = \{[\cos(\theta)]^2\sigma_{11}^y + [\sin(\theta)]^2\sigma_{22}^y + 2 \cdot \sin(\theta)\cos(\theta)\sigma_{12}^y\}\text{sgn}(\varepsilon_v), \quad (23.142.3)$$

assuming the strain is large enough to cause yielding in both directions.

If the shear strength is neglected, σ_{11} will vary smoothly between σ_{11}^y and σ_{22}^y and never exceed the maximum of the two yield stresses. This behavior is intuitively what we would like to see. However, if the value of shear yield stress isn't zero, σ_{11} will be greater than either σ_{11}^y or σ_{22}^y . To illustrate, if σ_{11}^y and σ_{22}^y are equal (a nominally isotropic response) the magnitude of the stress is

$$|\sigma_{11}| = \sigma_{11}^y + 2 \cdot \sin(\theta)\cos(\theta)\sigma_{12}^y, \quad (23.142.4)$$

and achieves a maximum value at 45 degrees of

$$|\sigma_{11}| = \sigma_{11}^y + \sigma_{12}^y. \quad (23.142.5)$$

For cases where there is anisotropy, the maximum occurs at a different angle and will have a different magnitude, but it will exceed the maximum uniaxial yield stress. In fact, a simple calculation using Mohr's circle shows that the maximum value will be

$$\sigma_{\max}^y = \frac{1}{2}(\sigma_{11}^y + \sigma_{22}^y) + \frac{1}{2}\sqrt{(\sigma_{11}^y - \sigma_{22}^y)^2 + 4\sigma_{12}^y}, \quad (23.142.6)$$

To correct for the systematic overestimation of the off-axis strength by MAT_HONEYCOMB, MAT_TRANSVERSELY_ISOTROPIC_CRUSHABLE_FOAM has been implemented in LS-DYNA. It uses a single yield surface, calculated dynamically from the six yield stresses specified by the user. The yield surface hardens and softens as a function of the volumetric strain through the yield stress functions. While the cost of the model is higher than for MAT_HONEYCOMB, its superior response off-axis makes it the model of choice for critical applications involving many types of low-density foams.

23.143 Material Model 143: Wood Model

The wood model is a transversely isotropic material and is available for solid elements. The development of this model was done by Murray [2002], who provided the documentation that follows, under a contract from the FHWA.

The general constitutive relation for an orthotropic material, written in terms of the principal material directions [Bodig & Jayne, 1993] is:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix}. \quad (23.143.1)$$

The subscripts 1, 2, and 3 refer to the longitudinal, tangential, and radial, stresses and strains ($\sigma_1 = \sigma_{11}$, $\sigma_2 = \sigma_{22}$, $\sigma_3 = \sigma_{33}$, $\varepsilon_1 = \varepsilon_{11}$, $\varepsilon_2 = \varepsilon_{22}$, $\varepsilon_3 = \varepsilon_{33}$), respectively. The subscripts 4, 5, and 6 are in a shorthand notation that refers to the shearing stresses and strains ($\sigma_4 = \sigma_{12}$, $\sigma_5 = \sigma_{13}$, $\sigma_6 = \sigma_{23}$, $\varepsilon_4 = \varepsilon_{12}$, $\varepsilon_5 = \varepsilon_{13}$, $\varepsilon_6 = \varepsilon_{23}$). As an alternative notation for wood, it is common to substitute L (longitudinal) for 1, T (tangential) for 2, and R (radial) for 3. The components of the constitutive matrix, C_{ij} , are listed here in terms of the nine independent elastic constants of an orthotropic material:

$$\begin{aligned} C_{11} &= \frac{E_{11}(1 - \nu_{23}\nu_{32})}{\Delta}, \\ C_{22} &= \frac{E_{22}(1 - \nu_{31}\nu_{13})}{\Delta}, \\ C_{33} &= \frac{E_{33}(1 - \nu_{12}\nu_{21})}{\Delta}, \\ C_{12} &= \frac{(\nu_{21} + \nu_{31}\nu_{23})E_{11}}{\Delta}, \\ C_{13} &= \frac{(\nu_{31} + \nu_{21}\nu_{32})E_{11}}{\Delta}, \\ C_{23} &= \frac{(\nu_{32} + \nu_{12}\nu_{31})E_{22}}{\Delta}, \\ C_{44} &= G_{12}, \\ C_{55} &= G_{13}, \\ C_{66} &= G_{23}, \\ \Delta &= 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}. \end{aligned} \quad (23.143.2)$$

The following identity, relating the dependent (minor Poisson's ratios ν_{21} , ν_{31} , and ν_{32}) and independent elastic constants, is obtained from symmetry considerations of the constitutive matrix:

$$\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j} \quad \text{for } i, j = 1, 2, 3. \quad (23.143.3)$$

One common assumption is that wood materials are transversely isotropic. This means that the properties in the tangential and radial directions are modeled the same, *i.e.* $E_{22} = E_{33}$, $G_{12} = G_{13}$, and $\nu_{12} = \nu_{13}$. This reduces the number of independent elastic constants to five, E_{11} , E_{22} , ν_{12} , G_{12} , and G_{23} . Further, Poisson's ratio in the isotropic plane, ν_{23} , is not an independent quantity. It is calculated from the isotropic relation: $\nu = (E - 2G)/2G$ where $E = E_{22} = E_{33}$ and $G = G_{23}$. Transverse isotropy is a reasonable assumption because the difference between the tangential and radial properties of wood (particularly Southern yellow pine and Douglas fir) is small in comparison with the difference between the tangential and longitudinal properties.

The yield surfaces parallel and perpendicular to the grain are formulated from six ultimate strength measurements obtained from uniaxial and pure-shear tests on wood specimens:

X_T	Tensile strength parallel to the grain
X_C	Compressive strength parallel to the grain
Y_T	Tensile strength perpendicular to the grain
Y_C	Compressive strength perpendicular to the grain
$S_{ }$	Shear strength parallel to the grain
S_{\perp}	Shear strength perpendicular to the grain

Here X and Y are the strengths parallel and perpendicular to the grain, respectively, and S are the shear strengths. The formulation is based on the work of Hashin [1980].

For the parallel modes, the yield criterion is composed of two terms involving two of the five stress invariants of a transversely isotropic material. These invariants are $I_1 = \sigma_{11}$ and $I_4 = \sigma_{12}^2 + \sigma_{13}^2$. This criterion predicts that the normal and shear stresses are mutually weakening, *i.e.* the presence of shear stress reduces the strength below that measured in uniaxial stress tests. Yielding occurs when $f_{||} \geq 0$, where:

$$f_{||} = \frac{\sigma_{11}^2}{X^2} + \frac{(\sigma_{12}^2 + \sigma_{13}^2)}{S_{||}^2} - 1 \quad X = \begin{cases} X_t & \text{for } \sigma_{11} > 0 \\ X_c & \text{for } \sigma_{11} < 0 \end{cases} \quad (23.143.4)$$

For the perpendicular modes, the yield criterion is also composed of two terms involving two of the five stress invariants of a transversely isotropic material. These invariants are $I_2 = \sigma_{22} + \sigma_{33}$ and $I_3 = \sigma_{23}^2 - \sigma_{22}\sigma_{33}$. Yielding occurs when $f_{\perp} \geq 0$, where:

$$f_{\perp} = \frac{(\sigma_{22} + \sigma_{33})^2}{Y^2} + \frac{(\sigma_{23}^2 - \sigma_{22}\sigma_{33})}{S_{\perp}^2} - 1 \quad Y = \begin{cases} Y_t & \text{for } \sigma_{22} + \sigma_{33} > 0 \\ Y_c & \text{for } \sigma_{22} + \sigma_{33} < 0 \end{cases} \quad (23.143.5)$$

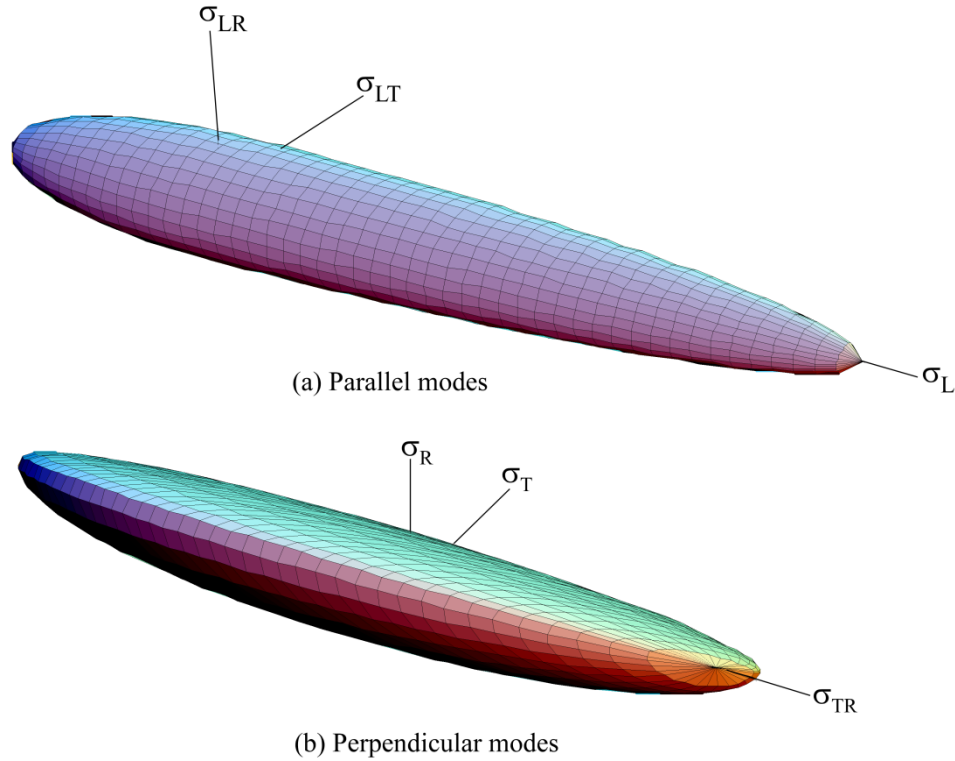


Figure 23.143.1. The yield criteria for wood produces smooth surfaces in stress space.

Each yield criterion is plotted in 3D in [Figure 23.143.1](#) in terms of the parallel and perpendicular stresses. Each criterion is a smooth surface (no corners).

The plasticity algorithms limit the stress components once the yield criteria in [Murry 2002] are satisfied. This is done by returning the trial elastic stress state back to the yield surface. The stress and strain tensors are partitioned into elastic and plastic parts. Partitioning is done with a return mapping algorithm which enforces the plastic consistency condition.

Separate plasticity algorithms are formulated for the parallel and perpendicular modes by enforcing separate consistency conditions. The solution of each consistency condition determines the consistency parameters, $\Delta\lambda_{||}$ and $\Delta\lambda_{\perp}$. The $\Delta\lambda$ solutions, in turn, determine the stress updates. No input parameters are required.

The stresses are readily updated from the total strain increments and the consistency parameters, as follows:

$$\bar{\sigma}_{ij}^{n+1} = \sigma_{ij}^{*n+1} - C_{ijkl} \Delta\lambda \left. \frac{\partial f}{\partial \sigma_{kl}} \right|_n \quad (23.143.6)$$

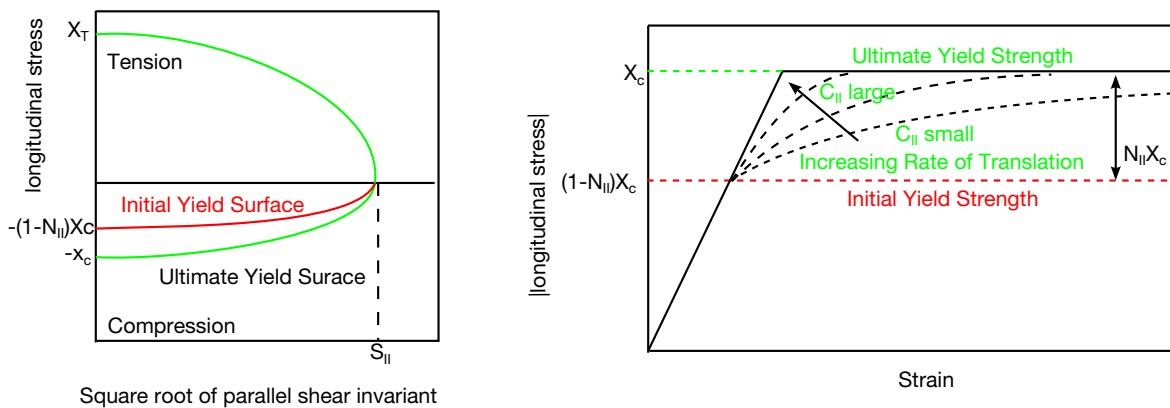
$$\sigma_{ij}^{*n+1} = \sigma_{ij}^n + C_{ijkl} \Delta\epsilon_{kl} \quad (23.143.7)$$

Here n denotes the n^{th} time step in the finite element analysis, and σ_{ij}^* are the trial elastic stress updates calculated from the total strain increments, $\Delta\epsilon_{ij}$, prior to application of plasticity. Each normal stress update depends on the consistency parameters and yield surface functions for both the parallel ($\Delta\lambda = \Delta\lambda_{\parallel}$ and $f = f_{\parallel}$) and perpendicular ($\Delta\lambda = \Delta\lambda_{\perp}$ and $f = f_{\perp}$) modes. Each shear stress update depends on just one consistency parameter and yield surface function. If neither parallel nor perpendicular yielding occurs ($f_{\parallel}^* < 0$ and $f_{\perp}^* < 0$) then $\Delta\lambda = 0$ and the stress update is trivial: $\hat{\sigma}_{ij}^{n+1} = \sigma_{ij}^{*n+1}$.

Wood exhibits pre-peak nonlinearity in compression parallel and perpendicular to the grain. Separate translating yield surface formulations are modeled for the parallel and perpendicular modes, which simulate gradual changes in moduli. Each initial yield surface hardens until it coincides with the ultimate yield surface. The initial location of the yield surface determines the onset of plasticity. The rate of translation determines the extent of the nonlinearity.

For each mode (parallel and perpendicular), the user inputs two parameters: the initial yield surface location in uniaxial compression, N , and the rate of translation, c . Say the user wants pre-peak nonlinearity to initiate at 70% of the peak strength. The user will input $N = 0.3$ so that $1 - N = 0.7$. If the user wants to harden rapidly, then a large value of c is input, like $c = 1$ msec. If the user wants to harden gradually, then a small value of c is input, like $c = 0.2$ msec.

The state variable that defines the translation of the yield surface is known as the *back stress*, and is denoted by α_{ij} . Hardening is modeled in compression, but not shear, so the only back stress required for the parallel modes is α_{11} . The value of the back stress is $\alpha_{11} = 0$ upon initial yield and $\alpha_{11} = -N_{\parallel} X_c$ at ultimate yield (in uniaxial compression). The maximum back stress occurs at ultimate yield and is equal to the total translation of the yield surface in stress space. The back stress components required for the



(a) Initial and ultimate yield surfaces

(b) Stress-strain behavior

Figure 23.143.2. Pre-peak nonlinearity in compression is modeled with translating yield surfaces that allow the user to specify the hardening response.

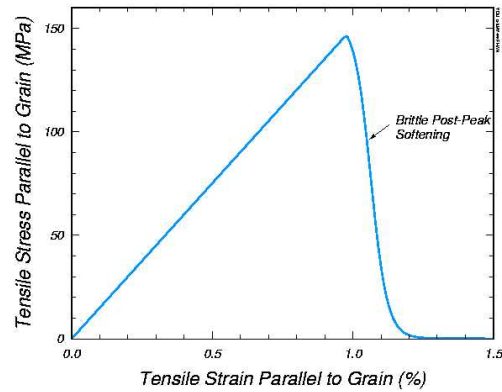
perpendicular modes are α_{22} and α_{33} . The value of the backstress sum is $\alpha_{22} + \alpha_{33} = 0$ upon initial yield and $\alpha_{22} + \alpha_{33} = -N_{\perp} Y_c$ at ultimate yield (biaxial compression without shear).

Separate damage formulations are modeled for the parallel and perpendicular modes. These formulations are loosely based on the work of Simo and Ju [1987]. If failure occurs in the parallel modes, then all six stress components are degraded uniformly. This is because parallel failure is catastrophic, and will render the wood useless. If failure occurs in the perpendicular modes, then only the perpendicular stress components are degraded. This is because perpendicular failure is not catastrophic: we expect the wood to continue to carry load in the parallel direction. Based on these assumptions, the following degradation model is implemented:

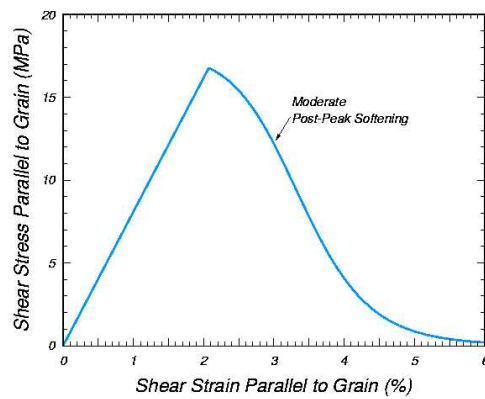
$$\begin{aligned}
 d_m &= \max(d(\tau_{\parallel}), d(\tau_{\perp})), \\
 d_{\parallel} &= d(\tau_{\parallel}), \\
 \sigma_{11} &= (1 - d_{\parallel})\bar{\sigma}_{11}, \\
 \sigma_{22} &= (1 - d_m)\bar{\sigma}_{22}, \\
 \sigma_{33} &= (1 - d_m)\bar{\sigma}_{33}, \\
 \sigma_{12} &= (1 - d_{\parallel})\bar{\sigma}_{12}, \\
 \sigma_{13} &= (1 - d_{\parallel})\bar{\sigma}_{13}, \\
 \sigma_{23} &= (1 - d_m)\bar{\sigma}_{23}.
 \end{aligned} \tag{23.143.8}$$

Here, each scalar damage parameter, d , transforms the stress tensor associated with the undamaged state, $\bar{\sigma}_{ij}$, into the stress tensor associated with the damaged state, σ_{ij} . The stress tensor $\bar{\sigma}_{ij}$ is calculated by the plasticity algorithm prior to application of the damage model. Each damage parameter ranges from zero for no damage and approaches unity for maximum damage. Thus $1 - d$ is a reduction factor associated with the amount of damage. Each damage parameter evolves as a function of a strain energy-type term. Mesh size dependency is regulated via a length scale based on the element size (cube root of volume). Damage-based softening is brittle in tension, less brittle in shear, and ductile (no softening) in compression, as demonstrated in [Figure 23.143.1](#).

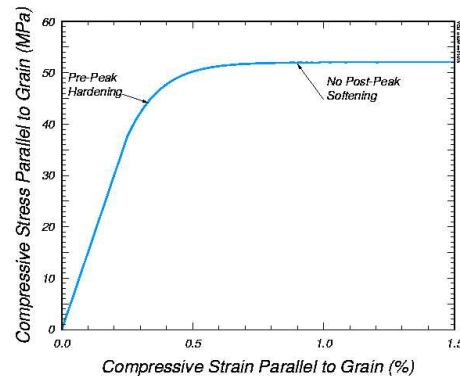
Element erosion occurs when an element fails in the parallel mode, and the parallel damage parameter exceeds $d_{\parallel} = 0.99$. Elements do not automatically erode when an element fails in the perpendicular mode. A flag is available, which, when set, allows elements to erode when the perpendicular damage parameter exceeds $d_{\perp} = 0.99$. Setting this flag is not recommended unless excessive perpendicular damage is causing computational difficulties.



(a) Tensile softening.



(b) Shear softening.



(c) Compressive yielding.

Figure 19.143.3. Softening response modeled for parallel modes of Southern yellow pine.

Data available in the literature for pine [Reid & Peng, 1997] indicates that dynamic strength enhancement is more pronounced in the perpendicular direction than in the parallel direction. Therefore, separate rate effects formulations are modeled for the parallel

and perpendicular modes. The formulations increase strength with increasing strain rate by expanding each yield surface:

$$\begin{aligned}\sigma_{11} &= X + E_{11} \dot{\epsilon} \eta_{\parallel} && \text{Parallel} \\ \sigma_{22} &= Y + E_{22} \dot{\epsilon} \eta_{\perp} && \text{Perpendicular}\end{aligned}\quad (23.143.9)$$

Here X and Y are the static strengths, σ_{11} and σ_{22} are the dynamic strengths, and $E_{11} \dot{\epsilon} \eta_{\parallel}$ and $E_{22} \dot{\epsilon} \eta_{\perp}$ are the excess stress components. The excess stress components depend on the value of the fluidity parameter, η , as well as the stiffness and strain rate. The user inputs two values, η_0 and n , to define each fluidity parameter:

$$\begin{aligned}\eta_{\parallel} &= \frac{\eta_{0\parallel}}{\dot{\epsilon}^{n_{\parallel}}}, \\ \eta_{\perp} &= \frac{\eta_{0\perp}}{\dot{\epsilon}^{n_{\perp}}}.\end{aligned}\quad (23.143.10)$$

The two parameter formulation [Murray, 1997] allows the user to model a nonlinear variation in dynamic strength with strain rate. Setting $n = 0$ allows the user to model a linear variation in dynamic strength with strain rate.

23.144 Material Model 144: Pitzer Crushable Foam

The logarithmic volumetric strain is defined in terms of the relative volume, V , as:

$$\gamma = -\ln(V). \quad (23.144.1)$$

In defining the curves the stress and strain pairs should be positive values starting with a volumetric strain value of zero.

Viscous damping in the model follows an implementation identical to that of material type 57.

23.147 Material Model 147: FHWA Soil Model

A brief discussion of the FHWA soil model is given. The elastic properties of the soil are isotropic. The implementation of the modified Mohr-Coulomb plasticity surface is based on the work of Abbo and Sloan [1995]. The model is extended to include excess pore water effects, strain softening, kinematic hardening, strain rate effects, and element deletion.

The modified yield surface is a hyperbola fitted to the Mohr-Coulomb surface. At the crossing of the pressure axis (zero shear strength) the modified surface is a smooth surface and it is perpendicular to the pressure axis. The yield surface is given as

$$F = -P\sin\phi + \sqrt{J_2 K(\theta)^2 + AHYP^2 \sin^2\phi} - c\cos\phi = 0, \quad (23.147.1)$$

where P is the pressure, ϕ is the internal friction angle, $K(\theta)$ is a function of the angle in

$$F = -P\sin\phi + \sqrt{J_2 K(\theta)^2 + AHYP^2 \sin^2\phi} - c\cos\phi = 0, \quad (23.147.2)$$

deviatoric plane, $\sqrt{J_2}$ is the square root of the second invariant of the stress deviator, c is the amount of cohesion and

$$\cos 3\theta = \frac{3\sqrt{3}J_3}{2J_2^{\frac{3}{2}}}, \quad (23.147.3)$$

J_3 is the third invariant of the stress deviator, $AHYP$ is a parameter for determining how close to the standard Mohr-Coulomb yield surface the modified surface is fitted. If the user defined parameter, $AHYP$, is input as zero, the standard Mohr-Coulomb surface is recovered. The parameter $ahyp$ should be set close to zero, based on numerical considerations, but always less than $c \cot\phi$. It is best not to set the cohesion, c , to very small values as this causes excessive iterations in the plasticity routines.

To generalize the shape in the deviatoric plane, we have changed the standard Mohr-Coulomb $K(\theta)$ function to a function used by Klisinski [1985]

$$K(\theta) = \frac{4(1 - e^2)\cos^2\theta + (2e - 1)^2}{2(1 - e^2)\cos\theta + (2e - 1)[4(1 - e^2)\cos^2\theta + 5e^2 - 4e]^{\frac{1}{2}}}, \quad (23.147.4)$$

where e is a material parameter describing the ratio of triaxial extension strength to triaxial compression strength. If e is set equal to 1, then a circular cone surface is formed. If e is set to 0.55, then a triangular surface is found, $K(\theta)$ is defined for $0.5 < e \leq 1.0$.

To simulate non-linear strain hardening behavior the friction, angle ϕ is increased as a function of the effective plastic strain,

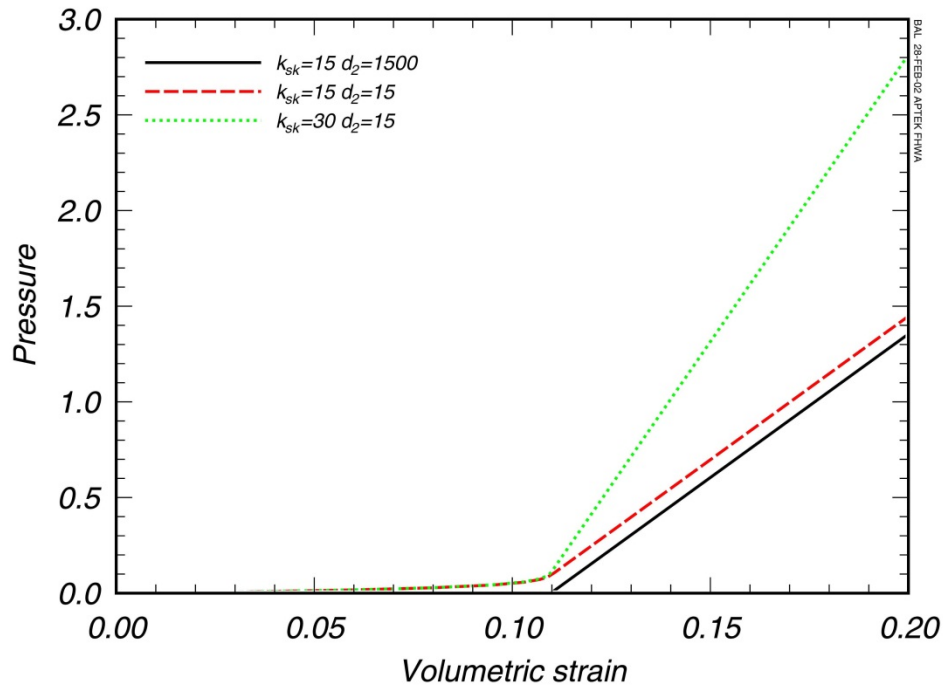


Figure 23.147.1. Pressure versus volumetric strain showing the effects of D1 parameter.

$$\Delta\varphi = E_t \left(1 - \frac{\varphi - \varphi_{\text{init}}}{A_n \varphi_{\text{max}}} \right) \Delta\varepsilon_{\text{eff plas}}. \quad (23.147.5)$$

where $\varepsilon_{\text{eff plas}}$ is the effective plastic strain. A_n is the fraction of the peak strength internal friction angle where nonlinear behavior begins, $0 < A_n \leq 1$. The input parameter E_t determines the rate of the nonlinear hardening.

To simulate the effects of moisture and air voids including excess pore water pressure, both the elastic and plastic behaviors can be modified. The bulk modulus is

$$K = \frac{K_i}{1 + K_i D_1 n_{\text{cur}}}. \quad (23.147.6)$$

where

K_i = initial bulk modulus

n_{cur} = current porosity = $\text{Max}[0, (w - \varepsilon_v)]$

w = volumetric strain corresponding to the volume of air voids = $n(1 - S)$

ε_v = total volumetric strain

D_1 = material constant controlling the stiffness before the air voids are collapsed

n = porosity of the soil = $\frac{e}{1 + e}$

e = void ratio = $\frac{\gamma_{\text{sp}}(1 + m_c)}{\rho} - 1$

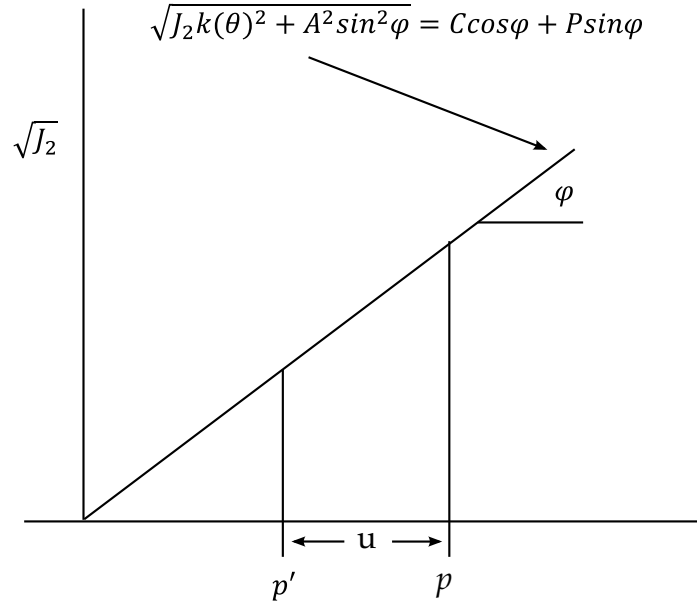


Figure 23.147.2. The effect on pressure due to pore water pressure.

$$S = \text{degree of saturation} = \frac{\rho m_c}{n(1 + m_c)}$$

and ρ , γ , m_c are the soil density, specific gravity, and moisture content, respectively.

Figure 23.147.1 shows the effect of the D_1 parameter on the pressure-volumetric strain relationship (bulk modulus). The bulk modulus will always be a monotonically increasing value, i.e.,

$$K_{j+1} = \begin{cases} \frac{K_i}{1 + K_i D_1 n_{\text{cur}}} & \text{if } \varepsilon_{vj+1} > \varepsilon_{vj} \\ K_j & \text{if } \varepsilon_{vj+1} \leq \varepsilon_{vj} \end{cases} \quad (23.147.7)$$

Note that the model is following the standard practice of assuming compressive stresses and strains are positive. If the input parameter D_1 is zero, then the standard linear elastic bulk modulus behavior is used.

To simulate the loss of shear strength due to excess pore water effects, the model uses a standard soil mechanics technique [Holtz and Kovacs, 1981] of reducing the total pressure, P , by the excess pore water pressure, u , to get an “effective pressure”, P' ; therefore,

$$P' = P - u. \quad (23.147.8)$$

Figure 23.147.2 shows pore water pressure will affect the algorithm for the plasticity surface. The excess pore water pressure reduces the total pressure, which will lower the shear strength, $\sqrt{J_2}$. A large excess pore water pressure can cause the effective pressure to become zero.

To calculate the pore water pressure, u , the model uses an equation similar to the equation used for the moisture effects on the bulk modulus.

$$u = \frac{K_{sk}}{1 + K_{sk} D_2 n_{cur}} \varepsilon_v, \quad (23.147.9)$$

where

K_{sk} = bulk modulus for soil without air voids (skeletal bulk modulus)

n_{cur} = current porosity = $\text{Max}[0, (w - \varepsilon_v)]$

w = volumetric strain corresponding to the volume of air voids = $n(1 - S)$

ε_v = total volumetric strain

D_2 = material constant controlling the pore water pressure before the air voids are collapsed to $D_2 \geq 0$

n = porosity of the soil = $\frac{e}{1 + e}$

e = void ratio = $\frac{\gamma_{sp}(1 + m_c)}{\rho} - 1$

S = degree of saturation = $\frac{\rho m_c}{n(1 + m_c)}$

and ρ, γ, m_c are the soil density, specific gravity, and moisture content, respectively. The pore water pressure will not be allowed to become negative, $u \geq 0$.

Figure 23.147.3 is a plot of the pore pressure versus volumetric strain for different parameter values. With the D_2 parameter set relatively high compared to K_{sk} there is no pore pressure until the volumetric strain is greater than the strains associated with the air voids. However, as D_2 is lowered, the pore pressure starts to increase before the air voids are totally collapsed. The K_{sk} parameter affects the slope of the post-void collapse pressure - volumetric behavior.

The parameter D_2 can be found from Skempton pore water pressure parameter B , where B is defined as [Holtz and Kovacs, 1981]:

$$B = \frac{1}{1 + n \frac{K_{sk}}{K_v}}, \quad (23.147.10)$$

$$D_2 = \frac{1 - B}{BK_{sk}[n(1 - S)]}.$$

To simulate strain softening behavior the FHWA soil model uses a continuum damage algorithm. The strain-based damage algorithm is based on the work of J. W. Ju and J. C. Simo [1987, 1989]. They proposed a strain based damage criterion, which is uncoupled from the plasticity algorithm.

For the damage criterion,

$$\zeta = -\frac{1}{K_i} \int \bar{P} d\varepsilon_{pv}, \quad (23.147.11)$$

where \bar{P} is the pressure and ε_{pv} is the plastic volumetric strain, the damaged stress is found from the undamaged stresses.

$$\sigma = (1 - d)\bar{\sigma}, \quad (23.147.12)$$

where d is the isotropic damage parameter. The damage parameter is found at step $j + 1$ as:

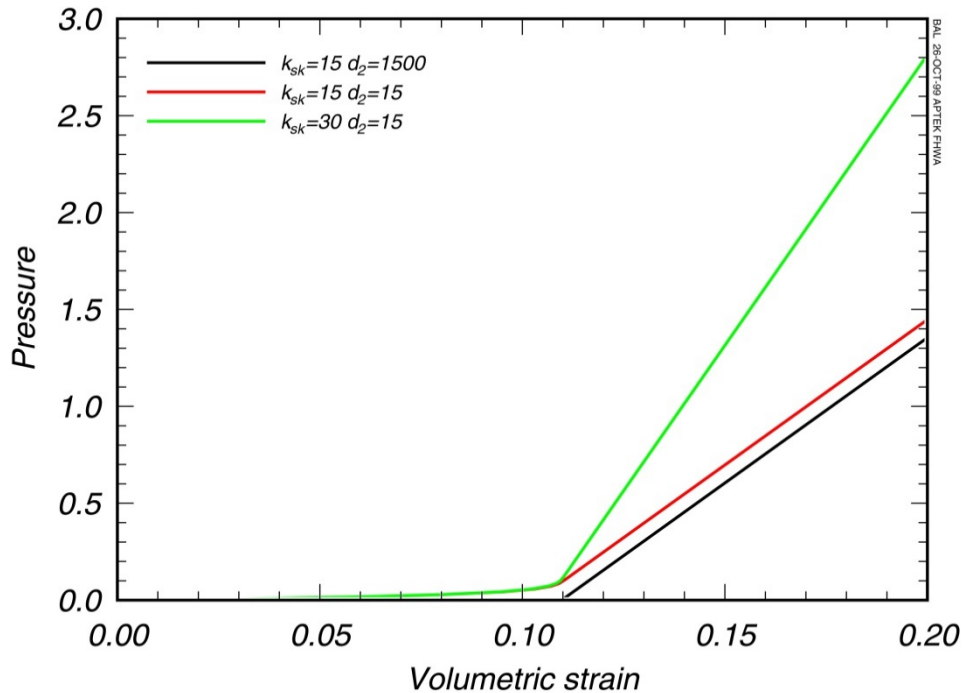


Figure 23.147.3. The effects of D_2 and K_{sk} parameters on pore water pressure.

$$\begin{aligned} d_{j+1} &= d_j & \text{if } \tilde{\zeta}_{j+1} \leq r_j \\ d_{j+1} &= \frac{\tilde{\zeta}_{j+1} - \tilde{\zeta}_0}{\alpha - \tilde{\zeta}_0} & \text{if } \tilde{\zeta}_{j+1} > r_j \end{aligned} \quad (23.147.13)$$

where r_t is a damage threshold surface, $r_{j+1} = \max\{r_j, \tilde{\zeta}_{j+1}\}$, and $\tilde{\zeta}_0 = r_0$ (DINT). The mesh sensitivity parameter, α , will be described below.

Typically, the damage, d , varies from 0 to a maximum of 1. However, some soils can have a residual strength that is pressure dependent. The residual strength is represented by ϕ_{res} , the minimum internal friction angle.

The maximum damage allowed is related to the internal friction angle of residual strength by:

$$d_{\text{max}} = \frac{\sin\phi - \sin\phi_{\text{res}}}{\sin\phi}, \quad (23.147.14)$$

If $\phi_{\text{res}} > 0$, then d_{max} , the maximum damage, will not reach 1, and the soil will have some residual strength.

When material models include strain softening, special techniques must be used to prevent mesh sensitivity. Mesh sensitivity is the tendency of the finite element model/analysis to produce significantly different results as the element size is reduced. The mesh sensitivity occurs because the softening in the model concentrates in one element. As the element size is reduced the failure becomes localized in smaller volumes, which causes less energy to be dissipated by the softening leading to instabilities or at least mesh sensitive behavior.

To eliminate or reduce the effects of strain softening mesh sensitivity, the softening parameter, α (the strain at full damage), must be modified as the element size changes. The FHWA soil model uses an input parameter, “void formation”, G_f , that is like fracture energy material property for metals. The void formation parameter is the area under the softening region of the pressure volumetric strain curve times the cube root of the element volume, $V^{1/3}$.

$$G_f = V^{1/3} \int_{\tilde{\zeta}_0}^{\alpha} P d\epsilon_v = \frac{P_{\text{peak}}(\alpha - \tilde{\zeta}_0)V^{1/3}}{2}, \quad (23.147.15)$$

with $\tilde{\zeta}_0$, the volumetric strain at peak pressure (strain at initial damage, DINT). Then α can be found as a function of the volume of the element V :

$$\alpha = \frac{2G_f}{K\tilde{\zeta}_0 V^{1/3}} + \tilde{\zeta}_0. \quad (23.147.16)$$

If G_f is made very small relative to $K\tilde{\zeta}_0 V^{1/3}$, then the softening behavior will be brittle.

Strain-rate enhanced strength is simulated by a two-parameter Devaut-Lions viscoplastic update algorithm, developed by Murray [1997]. This algorithm interpolates between the elastic trial stress (beyond the plasticity surface) and the inviscid stress. The inviscid stresses ($\bar{\sigma}$) are on the plasticity surface.

$$\bar{\sigma}_{vp} = (1 - \zeta)\bar{\sigma} + \zeta\bar{\sigma}_{trial}, \quad (23.147.17)$$

where $\zeta = \frac{\eta}{\Delta t + \eta}$, and $\eta = \left(\frac{\gamma_r}{\dot{\epsilon}}\right)^{(vn-1)/vn}$.

As ζ approaches 1, then the viscoplastic stress becomes the elastic trial stress. Setting the input value $\gamma_r = 0$ eliminates any strain-rate enhanced strength effects.

The model allows element deletion, if needed. As the strain softening (damage) increases, the effective stiffness of the element can get very small, causing severe element distortion and hourglassing. The element can be “deleted” to remedy this behavior. There are two input parameters that affect the point of element deletion. DAMLEV is the damage threshold where element deletion will be considered. EPSPRMAX is the maximum principal strain where element will be deleted. Therefore,

$$d \geq \text{DAMLEV} \quad \text{and} \quad \epsilon_{prmax} > \text{EPSPRMAX}, \quad (23.147.18)$$

for element deletion to occur. If DAMLEV is set to zero, there is no element deletion. Care must be taken when employing element deletion to assure that the internal forces are very small (element stiffness is zero) or significant errors can be introduced into the analysis.

The keyword option, **NEBRASKA**, gives the soil parameters used to validate the material model with experiments performed at University of Nebraska at Lincoln. The units for this default inputs are milliseconds, kilograms, and millimeters. There are no required input parameters except material id (MID). If different units are desired the unit conversion factors that need to multiply the default parameters can be input.

23.154 Material Model 154: Deshpande-Fleck Foam

$$\Phi = \hat{\sigma} - \sigma_Y, \quad (23.154.1)$$

The equivalent stress, $\hat{\sigma}$, is given by:

$$\hat{\sigma}^2 = \frac{\sigma_{VM}^2 + \alpha^2 \sigma_m^2}{1 + (\alpha/3)^2}, \quad (23.154.2)$$

where, σ_{VM} , is the von Mises effective stress,

$$\sigma_{VM} = \sqrt{\frac{3}{2} \sigma^{\text{dev}} : \sigma^{\text{dev}}}, \quad (23.154.3)$$

and, σ_m and σ^{dev} , is the mean and deviatoric stress

$$\begin{aligned} \sigma_m &= \text{tr}(\sigma) \\ \sigma^{\text{dev}} &= \sigma - \sigma_m \mathbf{I}. \end{aligned} \quad (23.154.4)$$

The yield stress σ_Y can be expressed as

$$\sigma_Y = \sigma_p + \gamma \frac{\hat{\varepsilon}}{\varepsilon_D} + \alpha_2 \left(\frac{1}{1 - (\hat{\varepsilon}/\varepsilon_D)^\beta} \right), \quad (23.154.5)$$

Here, σ_p , α_2 , γ and β are material parameters. The densification strain, ε_D , is defined as

$$\varepsilon_D = -\ln \left(\frac{\rho_f}{\rho_{f0}} \right), \quad (23.154.6)$$

where ρ_f is the foam density and ρ_{f0} is the density of the virgin material.

23.156 Material Model 156: Muscle

The material behavior of the muscle model is adapted from *MAT_SPRING_MUSCLE, the spring muscle model and treated here as a standard material. The initial length of muscle is calculated automatically. The force, relative length and shortening velocity are replaced by stress, strain and strain rate. A new parallel damping element is added.

The strain and normalized strain rate are defined respectively as

$$\begin{aligned}\varepsilon &= \frac{l}{l_o} - 1 = L - 1 \\ \dot{\varepsilon} &= \frac{\dot{l}}{l_o \dot{\varepsilon}_{\max}} = \frac{V^M}{l_o * (SRM * SFR)} = \frac{V^M}{(l_o * SRM) * SFR} = \frac{V^M}{V_{\max} * SFR} = V,\end{aligned}\quad (23.156.1)$$

where l_o is the original muscle length.

From the relation above, it is known:

$$l_o = \frac{l_0}{1 + \varepsilon_0}, \quad (23.156.2)$$

where $\varepsilon_0 = \text{SNO}$; l_0 = muscle length at time 0. Stress of Contractile Element is:

$$\sigma_1 = \sigma_{\max} a(t) f(\varepsilon) g(\dot{\varepsilon}), \quad (23.156.3)$$

where $\sigma_{\max} = \text{PIS}$; $a(t) = \text{ALM}$; $f(\varepsilon) = \text{SVS}$; $g(\dot{\varepsilon}) = \text{SVR}$. Stress of Passive Element is:

$$\sigma_2 = \sigma_{\max} h(\varepsilon). \quad (23.156.4)$$

For exponential relationship:

$$h(\varepsilon) = \begin{cases} 0 & \varepsilon \leq 0 \\ \frac{1}{\exp(c) - 1} \left[\exp\left(\frac{c\varepsilon}{L_{\max}}\right) - 1 \right] & \varepsilon > 0 \quad c \neq 0 \\ \varepsilon / L_{\max} & \varepsilon > 0 \quad c = 0 \end{cases} \quad (23.156.5)$$

where $L_{\max} = 1 + \text{SSM}$; and $c = \text{CER}$. Stress of Damping Element is:

$$\sigma_3 = D \dot{\varepsilon}_{\max} \dot{\varepsilon} \quad (23.156.6)$$

Total Stress is:

$$\sigma = \sigma_1 + \sigma_2 + \sigma_3. \quad (23.156.7)$$

23.158 Material Model 158: Rate Sensitive Composite Fabric

See material type 58, Laminated Composite Fabric, for the treatment of the composite material.

Rate effects are taken into account through a Maxwell model using linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.158.1)$$

where $g_{ijkl}(t - \tau)$ is the relaxation function for different stress measures. This stress is added to the stress tensor determined from the strain energy functional. Since we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \sum_{m=1}^N G_m e^{-\beta_m t}. \quad (23.158.2)$$

We characterize this in the input by the shear moduli, G_i , and the decay constants, β_i . An arbitrary number of terms, not exceeding 6, may be used when applying the viscoelastic model. The composite failure is not directly affected by the presence of the viscous stress tensor.

23.159 Material Model 159: Continuous Surface Cap Model

This is a cap model with a smooth intersection between the shear yield surface and hardening cap, as shown in [Figure 23.159.1](#). The initial damage surface coincides with the yield surface. Rate effects are modeled with viscoplasticity. See [Murray 2007] for a more complete description of the material model.

Stress Invariants. The yield surface is formulated in terms of three stress invariants: J_1 is the first invariant of the stress tensor, J'_2 is the second invariant of the deviatoric stress tensor, and J'_3 is the third invariant of the deviatoric stress tensor. The invariants are defined in terms of the deviatoric stress tensor, S_{ij} and pressure, P , as follows:

$$\begin{aligned} J_1 &= 3P, \\ J'_2 &= \frac{1}{2} S_{ij} S_{ij}, \\ J'_3 &= \frac{1}{3} S_{ij} S_{jk} S_{ki}. \end{aligned} \quad (23.159.1)$$

Plasticity Surface. The three invariant yield function is based on these three invariants, and the cap hardening parameter, k , as follows:

$$f(J_1, J'_2, J'_3, \kappa) = J'_2 - \Re^2 F_f^2 F_c. \quad (23.159.2)$$

Here F_f is the shear failure surface, F_c is the hardening cap, and \Re is the Rubin three-invariant reduction factor. The cap hardening parameter κ is the value of the pressure invariant at the intersection of the cap and shear surfaces.

Trial elastic stress invariants are temporarily updated via the trial elastic stress tensor, σ^T . These are denoted J_1^T , J_2^T and J_3^T . Elastic stress states are modeled when

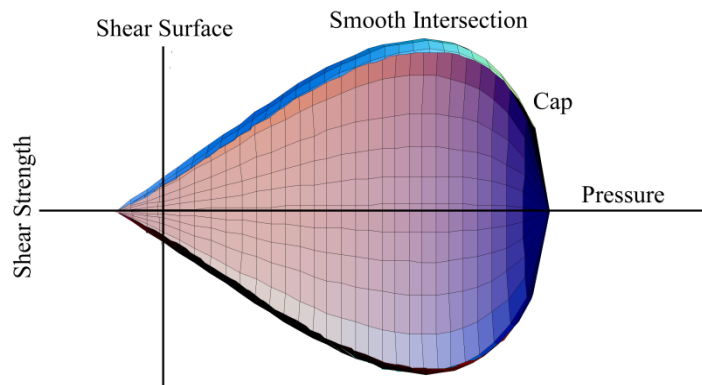


Figure 23.159.1. General Shape of the concrete model yield surface in two-dimensions.

$f(J_1, J'_2, J'_3, \kappa^T) < 0$. Elastic-plastic stress states are modeled when $f(J_1, J'_2, J'_3, \kappa^T) > 0$. In this case, the plasticity algorithm returns the stress state to the yield surface such that $f(J_1^P, J'_2{}^P, J'_3{}^P, \kappa^P) = 0$. This is accomplished by enforcing the plastic consistency condition with associated flow.

Shear Failure Surface. The strength of concrete is modeled by the shear surface in the tensile and low confining pressure regimes:

$$F_f(J_1) = \alpha - \lambda e^{-\beta J_1} + \theta J_1. \quad (23.159.3)$$

Here the values of α , β , λ , and θ are selected by fitting the model surface to strength measurements from triaxial compression (txc) tests conducted on plain concrete cylinders.

Rubin Scaling Function. Concrete fails at lower values of $\sqrt{3J'_2}$ (principal stress difference) for triaxial extension (txe) and torsion (tor) tests than it does for txc tests conducted at the same pressure. The Rubin scaling function \mathfrak{R} determines the strength of concrete for any state of stress relative to the strength for txc, via $\mathfrak{R}F_f$. Strength in torsion is modeled as Q_1F_f . Strength in txe is modeled as Q_2F_f , where:

$$\begin{aligned} Q_1 &= \alpha_1 - \lambda_1 e^{-\beta_1 J_1} + \theta_1 J_1, \\ Q_2 &= \alpha_2 - \lambda_2 e^{-\beta_2 J_1} + \theta_2 J_1. \end{aligned} \quad (23.159.4)$$

Cap Hardening Surface. The strength of concrete is modeled by a combination of the cap and shear surfaces in the low to high confining pressure regimes. The cap is used to model plastic volume change related to pore collapse (although the pores are not explicitly modeled). The isotropic hardening cap is a two-part function that is either unity or an ellipse:

$$F_c(J_1, \kappa) = 1 - \frac{[J_1 - L(\kappa)][|J_1 - L(\kappa)| + J_1 - L(\kappa)]}{2[X(\kappa) - L(\kappa)]^2}, \quad (23.159.5)$$

where $L(\kappa)$ is defined as:

$$L(\kappa) = \begin{cases} \kappa & \text{if } \kappa > \kappa_0 \\ \kappa_0 & \text{otherwise} \end{cases}. \quad (23.159.6)$$

The equation for F_c is equal to unity for $J_1 \leq L(\kappa)$. It describes the ellipse for $J_1 > L(\kappa)$. The intersection of the shear surface and the cap is at $J_1 = \kappa$. κ_0 is the value of J_1 at the initial intersection of the cap and shear surfaces before hardening is engaged (before the cap moves). The equation for $L(\kappa)$ restrains the cap from retracting past its initial location at κ_0 .

The intersection of the cap with the J_1 axis is at $J_1 = X(\kappa)$. This intersection depends upon the cap ellipticity ratio R , where R is the ratio of its major to minor axes:

$$X(\kappa) = L(\kappa) + RF_f(L(\kappa)). \quad (23.159.7)$$

The cap moves to simulate plastic volume change. The cap expands ($X(\kappa)$ and κ increase) to simulate plastic volume compaction. The cap contracts ($X(\kappa)$ and κ decrease) to simulate plastic volume expansion, called dilation. The motion (expansion and contraction) of the cap is based upon the hardening rule:

$$\varepsilon_v^p = W(1 - e^{-D_1(X-X_0)-D_2(X-X_0)^2}). \quad (23.159.8)$$

Here ε_v^p the plastic volume strain, W is the maximum plastic volume strain, and D_1 and D_2 are model input parameters. X_0 is the initial location of the cap when $\kappa = \kappa_0$.

The five input parameters (X_0 , W , D_1 , D_2 , and R) are obtained from fits to the pressure-volumetric strain curves in isotropic compression and uniaxial strain. X_0 determines the pressure at which compaction initiates in isotropic compression. R combined with X_0 , determines the pressure at which compaction initiates in uniaxial strain. D_1 and D_2 determine the shape of the pressure-volumetric strain curves. W determines the maximum plastic volume compaction.

Shear Hardening Surface. In unconfined compression, the stress-strain behavior of concrete exhibits nonlinearity and dilation prior to the peak. Such behavior is modeled with an initial shear yield surface, $N_H F_f$, which hardens until it coincides with the ultimate shear yield surface, F_f . Two input parameters are required. One parameter, N_H , initiates hardening by setting the location of the initial yield surface. A second parameter, C_H , determines the rate of hardening (amount of nonlinearity).

Damage. Concrete exhibits softening in the tensile and low to moderate compressive regimes.

$$\sigma_{ij}^d = (1 - d)\sigma_{ij}^{vp}, \quad (23.159.9)$$

A scalar damage parameter, d , transforms the viscoplastic stress tensor without damage, denoted σ^{vp} , into the stress tensor with damage, denoted σ^d . Damage accumulation is based upon two distinct formulations, which we call brittle damage and ductile damage. The initial damage threshold is coincident with the shear plasticity surface, so the threshold does not have to be specified by the user.

Ductile Damage. Ductile damage accumulates when the pressure (P) is compressive and an energy-type term, τ_c , exceeds the damage threshold, τ_{0c} . Ductile damage accumulation depends upon the total strain components, ε_{ij} , as follows:

$$\tau_c = \sqrt{\frac{1}{2} \sigma_{ij} \varepsilon_{ij}} \quad (23.159.10)$$

The stress components σ_{ij} are the elasto-plastic stresses (with kinematic hardening) calculated before application of damage and rate effects.

Brittle Damage. Brittle damage accumulates when the pressure is tensile and an energy-type term, τ_t , exceeds the damage threshold, τ_{0t} . Brittle damage accumulation depends upon the maximum principal strain, ε_{\max} , as follows:

$$\tau_t = \sqrt{E\varepsilon_{\max}^2}. \quad (23.159.11)$$

Softening Function. As damage accumulates, the damage parameter d increases from an initial value of zero, towards a maximum value of one, via the following formulations:
Brittle Damage:

$$d(\tau_1) = \frac{0.999}{D} \left(\frac{1 + D}{1 + D \exp[-C(\tau_t - \tau_{0t})]} - 1 \right). \quad (23.159.12)$$

Ductile Damage:

$$d(\tau_1) = \frac{d_{\max}}{B} \left(\frac{1 + B}{1 + B \exp[-A(\tau_c - \tau_{0c})]} - 1 \right). \quad (23.159.13)$$

The damage parameter that is applied to the six stresses is equal to the current maximum of the brittle or ductile damage parameter. The parameters A and B or C and D set the shape of the softening curve plotted as stress-displacement or stress-strain. The parameter d_{\max} is the maximum damage level that can be attained. It is internally calculated and is less than one at moderate confining pressures (see [Murray 2007] for a more detailed description). The compressive softening parameter, A , may also be reduced with confinement, using the input parameter PMOD, as follows:

$$A = A(d_{\max} + 0.001)^{\text{PMOD}}. \quad (23.159.14)$$

Regulating Mesh Size Sensitivity. The concrete model maintains constant fracture energy, regardless of element size. The fracture energy is defined here as the area under the stress-displacement curve from peak strength to zero strength. This is done by internally formulating the softening parameters A and C in terms of the element length, l (cube root of the element volume), the fracture energy, G_f , the initial damage threshold, τ_{0t} or τ_{0c} , and the softening shape parameters, D or B .

The fracture energy is calculated from up to five user-specified input parameters (G_{fc} , G_{ft} , G_{fs} , pwrc, pwrc). The user specifies three distinct fracture energy values. These are the fracture energy in uniaxial tensile stress, G_{ft} , pure shear stress, G_{fs} , and uniaxial compressive stress, G_{fc} . The model internally selects the fracture energy from equations which interpolate between the three fracture energy values as a function of the stress state (expressed via two stress invariants). The interpolation equations depend upon the user-specified input powers PWRC and PWRT, as follows.

$$(23.159.15)$$

if the pressure is compressive

$$G_f = G_{fs} + \text{trans}(G_{fc} - G_{fs}) \text{ where } \text{trans} = \left(\frac{J_1}{\sqrt{3}J'_2} \right)^{\text{PWRC}}$$

The internal parameter trans is limited to range between 0 and 1.

Element Erosion. An element loses all strength and stiffness as $d \rightarrow 1$. To prevent computational difficulties with very low stiffness, element erosion is available as a user option. An element erodes when $d > 0.99$ and the maximum principal strain is greater than a user supplied input value, 1-erode.

Viscoplastic Rate Effects. At each time step, the viscoplastic algorithm interpolates between the elastic trial stress, σ_{ij}^T , and the inviscid stress (without rate effects), σ_{ij}^P , to set the viscoplastic stress (with rate effects), σ_{ij}^{VP} :

$$\sigma_{ij}^{VP} = (1 - \gamma)\sigma_{ij}^T + \gamma\sigma_{ij}^P, \quad (23.159.16)$$

with $\gamma = \frac{\Delta t/\eta}{1 + \Delta t/\eta}$.

This interpolation depends upon the effective fluidity coefficient, η , and the time step, Δt . The effective fluidity coefficient is internally calculated from five user-supplied input parameters and interpolation equations:

if the pressure is tensile

$$\eta = \eta_s + \text{trans}(\eta_t - \eta_s) \quad \text{trans} = \left(\frac{-J_1}{\sqrt{3}J'_2} \right)^{\text{pwrt}}$$

if the pressure is compressive

$$\eta = \eta_s + \text{trans}(\eta_c - \eta_s) \quad \text{trans} = \left(\frac{J_1}{\sqrt{3}J'_2} \right)^{\text{pwrc}} \quad (23.159.17)$$

$$\eta_t = \frac{\eta_{0t}}{\dot{\epsilon} N_t} \quad \eta_c = \frac{\eta_{0c}}{\dot{\epsilon} N_c} \quad \eta_s = \text{SRATE } \eta_t$$

The input parameters are η_{0t} and N_t for fitting uniaxial tensile stress data, η_{0c} and N_c for fitting the uniaxial compressive stress data, and SRATE for fitting shear stress data. The effective strain rate is $\dot{\epsilon}$.

This viscoplastic model may predict substantial rate effects at high strain rates ($\dot{\epsilon} > 100$). To limit rate effects at high strain rates, the user may input overstress limits in tension (OVERT) and compression (OVERC). These input parameters limit calculation of the fluidity parameter, as follows:

$$\text{If } E\dot{\epsilon}\eta > \text{OVER, then } \eta = \frac{\text{over}}{E\dot{\epsilon}} \quad (23.159.18)$$

where OVER = OVERT when the pressure is tensile, and OVER = OVERC when the pressure is compressive.

The user has the option of increasing the fracture energy as a function of effective strain rate via the REPOW input parameter, as follows:

$$G_f^{\text{rate}} = G_f \left(1 + \frac{E\dot{\epsilon}\eta}{r^s \sqrt{E}} \right)^{\text{REPOW}} \quad (23.159.19)$$

Here G_f^{rate} is the fracture energy enhanced by rate effects, and r^s is the internally calculated damage threshold before application of rate effects (see [Murray 2007]). The term in brackets is greater than, or equal to one, and is the approximate ratio of the dynamic to static strength.

23.161 Material Models 161 and 162: Composite MSC

The unidirectional and fabric layer failure criteria and the associated property degradation models for material 161 are described as follows. All the failure criteria are expressed in terms of stress components based on ply level stresses ($\sigma_a, \sigma_b, \sigma_c, \tau_{ab}, \tau_{bc}, \tau_{ca}$) and the associated elastic moduli are ($E_a, E_b, E_c, G_{ab}, G_{bc}, G_{ca}$). Note that for the unidirectional model, a, b and c denote the fiber, in-plane transverse and out-of-plane directions, respectively, while for the fabric model, a, b and c denote the in-plane fill, in-plane warp and out-of-plane directions, respectively.

Unidirectional Lamina Model

Three criteria are used for fiber failure, one in tension/shear, one in compression and another one in crush under pressure. They are chosen in terms of quadratic stress forms as follows:

Tensile/shear fiber mode:

$$f_1 = \left(\frac{\langle \sigma_a \rangle}{S_{aT}} \right)^2 + \left(\frac{\tau_{ab}^2 + \tau_{ca}^2}{S_{FS}^2} \right) - 1 = 0. \quad (23.161.1)$$

Compression fiber mode:

$$f_2 = \left(\frac{\langle \sigma'_a \rangle}{S_{aC}} \right)^2 - 1 = 0, \quad \sigma'_a = -\sigma_a + \left\langle -\frac{\sigma_b + \sigma_c}{2} \right\rangle. \quad (23.161.2)$$

Crush mode:

$$f_3 = \left(\frac{\langle p \rangle}{S_{FC}} \right)^2 - 1 = 0, \quad p = -\frac{\sigma_a + \sigma_b + \sigma_c}{3}. \quad (23.161.3)$$

where $\langle \rangle$ are Macaulay brackets, S_{aT} and S_{aC} are the tensile and compressive strengths in the fiber direction, and S_{FS} and S_{FC} are the layer strengths associated with the fiber shear and crush failure, respectively.

Matrix mode failures must occur without fiber failure, and hence they will be on planes parallel to fibers. For simplicity, only two failure planes are considered: one is perpendicular to the planes of layering and the other one is parallel to them. The matrix failure criteria for the failure plane perpendicular and parallel to the layering planes, respectively, have the forms:

Perpendicular matrix mode:

$$f_4 = \left(\frac{\langle \sigma_b \rangle}{S_{bT}} \right)^2 + \left(\frac{\tau_{bc}}{S'_{bc}} \right)^2 + \left(\frac{\tau_{ab}}{S_{ab}} \right)^2 - 1 = 0. \quad (23.161.4)$$

Parallel matrix mode (Delamination):

$$f_5 = S^2 \left\{ \left(\frac{\langle \sigma_c \rangle}{S_{bT}} \right)^2 + \left(\frac{\tau_{bc}}{S_{bc}''} \right)^2 + \left(\frac{\tau_{ca}}{S_{ca}} \right)^2 \right\} - 1 = 0, \quad (23.161.5)$$

where S_{bT} is the transverse tensile strength. Based on the Coulomb-Mohr theory, the shear strengths for the transverse shear failure and the two axial shear failure modes are assumed to be the forms,

$$\begin{aligned} S_{ab} &= S_{ab}^{(0)} + \tan(\varphi) \langle -\sigma_b \rangle, \\ S'_{bc} &= S_{bc}^{(0)} + \tan(\varphi) \langle -\sigma_b \rangle, \\ S_{ca} &= S_{ca}^{(0)} + \tan(\varphi) \langle -\sigma_c \rangle, \\ S_{bc}'' &= S_{bc}^{(0)} + \tan(\varphi) \langle -\sigma_c \rangle, \end{aligned} \quad (23.161.6)$$

where φ is a material constant as $\tan(\varphi)$ is similar to the coefficient of friction, and $S_{ab}^{(0)}$, $S_{ca}^{(0)}$ and $S_{bc}^{(0)}$ are the shear strength values of the corresponding tensile modes.

Failure predicted by the criterion of f_4 can be referred to as transverse matrix failure, while the matrix failure predicted by f_5 , which is parallel to the layer, can be referred as the delamination mode when it occurs within the elements that are adjacent to the ply interface. Note that a scale factor S is introduced to provide better correlation of delamination area with experiments. The scale factor S can be determined by fitting the analytical prediction to experimental data for the delamination area.

When fiber failure in tension/shear mode is predicted in a layer by f_1 , the load carrying capacity of that layer is completely eliminated. All the stress components are reduced to zero instantaneously (100 time steps to avoid numerical instability). For compressive fiber failure, the layer is assumed to carry a residual axial load, while the transverse load carrying capacity is reduced to zero. When the fiber compressive failure mode is reached due to f_2 , the axial layer compressive strength stress is assumed to reduce to a residual value S_{RC} ($=SFFC * S_{AC}$). The axial stress is then assumed to remain constant, i.e., $\sigma_a = -S_{RC}$, for continuous compressive loading, while the subsequent unloading curve follows a reduced axial modulus to zero axial stress and strain state. When the fiber crush failure occurs, the material is assumed to behave elastically for compressive pressure, $p > 0$, and to carry no load for tensile pressure, $p < 0$.

When a matrix failure (delamination) in the a-b plane is predicted, the strength values for $S_{ca}^{(0)}$ and $S_{bc}^{(0)}$ are set to zero. This results in reducing the stress components σ_c , τ_{bc} and τ_{ca} to the fractured material strength surface. For tensile mode, $\sigma_c > 0$, these stress components are reduced to zero. For compressive mode, $\sigma_c < 0$, the normal stress σ_c is assumed to deform elastically for the closed matrix crack. Loading on the failure envelop, the shear stresses are assumed to 'slide' on the fractured strength surface (frictional shear stresses) like in an ideal plastic material, while the subsequent unloading shear stress-strain path follows reduced shear moduli to the zero shear stress and strain state for both τ_{bc} and τ_{ca} components.

The post failure behavior for the matrix crack in the a-c plane due to f_4 is modeled in the same fashion as that in the a-b plane as described above. In this case, when failure occurs, $S_{ab}^{(0)}$ and $S_{bc}^{(0)}$ are reduced to zero instantaneously. The post fracture response is then governed by failure criterion of f_5 with $S_{ab}^{(0)} = 0$ and $S_{bc}^{(0)} = 0$. For tensile mode, σ_b , τ_{ab} and τ_{bc} are zero. For compressive mode, $\sigma_b < 0$, σ_b is assumed to be elastic, while τ_{ab} and τ_{bc} 'slide' on the fracture strength surface as in an ideal plastic material, and the unloading path follows reduced shear moduli to the zero shear stress and strain state. It should be noted that τ_{bc} is governed by both the failure functions and should lie within or on each of these two strength surfaces.

Fabric Lamina Model

The fiber failure criteria of Hashin for a unidirectional layer are generalized to characterize the fiber damage in terms of strain components for a plain weave layer. The fill and warp fiber tensile/shear failure are given by the quadratic interaction between the associated axial and shear stresses, i.e.

$$f_6 = \left(\frac{\langle \sigma_a \rangle}{S_{aT}} \right)^2 + \frac{(\tau_{ab}^2 + \tau_{ca}^2)}{S_{aFS}^2} - 1 = 0, \quad (23.161.7)$$

$$f_7 = \left(\frac{\langle \sigma_b \rangle}{S_{bT}} \right)^2 + \frac{(\tau_{ab}^2 + \tau_{bc}^2)}{S_{bFS}^2} - 1 = 0, \quad (23.161.8)$$

where S_{aT} and S_{bT} are the axial tensile strengths in the fill and warp directions, respectively, and S_{aFS} and S_{bFS} are the layer shear strengths due to fiber shear failure in the fill and warp directions. These failure criteria are applicable when the associated σ_a or σ_b is positive. It is assumed $S_{aFS} = SFS$, and

$$S_{bFS} = SFS * \frac{S_{bT}}{S_{aT}}. \quad (23.161.9)$$

When σ_a or σ_b is compressive, it is assumed that the in-plane compressive failure in both the fill and warp directions are given by the maximum stress criterion, i.e.

$$f_8 = \left[\frac{\langle \sigma'_a \rangle}{S_{aC}} \right]^2 - 1 = 0, \quad \sigma'_a = -\sigma_a + \langle -\sigma_c \rangle, \quad (23.161.10)$$

$$f_9 = \left[\frac{\langle \sigma'_b \rangle}{S_{bC}} \right]^2 - 1 = 0, \quad \sigma'_b = -\sigma_b + \langle -\sigma_c \rangle. \quad (23.161.11)$$

where S_{aC} and S_{bC} are the axial compressive strengths in the fill and warp directions, respectively. The crush failure under compressive pressure is

$$f_{10} = \left(\frac{\langle p \rangle}{S_{FC}} \right)^2 - 1 = 0, \quad p = - \frac{\sigma_a + \sigma_b + \sigma_c}{3}. \quad (23.161.12)$$

A plain weave layer can fail under in-plane shear stress without the occurrence of fiber breakage. This in-plane matrix failure mode is given by

$$f_{11} = \left(\frac{\tau_{ab}}{S_{ab}} \right)^2 - 1 = 0, \quad (23.161.13)$$

where S_{ab} is the layer shear strength due to matrix shear failure.

Another failure mode, which is due to the quadratic interaction between the thickness stresses, is expected to be mainly a matrix failure. This through the thickness matrix failure criterion is

$$f_{12} = S^2 \left\{ \left(\frac{\langle \sigma_c \rangle}{S_{cT}} \right)^2 + \left(\frac{\tau_{bc}}{S_{bc}} \right)^2 + \left(\frac{\tau_{ca}}{S_{ca}} \right)^2 \right\} - 1 = 0, \quad (23.161.14)$$

where S_{cT} is the through the thickness tensile strength, and S_{bc} , and S_{ca} are the shear strengths assumed to depend on the compressive normal stress σ_c , i.e.,

$$\begin{Bmatrix} S_{ca} \\ S_{bc} \end{Bmatrix} = \begin{Bmatrix} S_{ca}^{(0)} \\ S_{bc}^{(0)} \end{Bmatrix} + \tan(\varphi) \langle -\sigma_c \rangle. \quad (23.161.15)$$

When failure predicted by this criterion occurs within elements that are adjacent to the ply interface, the failure plane is expected to be parallel to the layering planes, and, thus, can be referred to as the delamination mode. Note that a scale factor S is introduced to provide better correlation of delamination area with experiments. The scale factor S can be determined by fitting the analytical prediction to experimental data for the delamination area.

Similar to the unidirectional model, when fiber tensile/shear failure is predicted in a layer by f_6 or f_7 , the load carrying capacity of that layer in the associated direction is completely eliminated. For compressive fiber failure due to f_8 or f_9 , the layer is assumed to carry a residual axial load in the failed direction, while the load carrying capacity transverse to the failed direction is assumed unchanged. When the compressive axial stress in a layer reaches the compressive axial strength S_{aC} or S_{bC} , the axial layer stress is assumed to be reduced to the residual strength S_{aRC} or S_{bRC} where $S_{aRC} = \text{SFFC} * S_{aC}$ and $S_{bRC} = \text{SFFC} * S_{bC}$. The axial stress is assumed to remain constant, i.e., $\sigma_a = -S_{aCR}$ or $\sigma_b = -S_{bCR}$, for continuous compressive loading, while the subsequent unloading curve follows a reduced axial modulus. When the fiber crush failure has occurred, the material is assumed to behave elastically for compressive pressure, $p > 0$, and to carry no load for tensile pressure, $p < 0$.

When the in-plane matrix shear failure is predicted by f_{11} the axial load carrying capacity within a failed element is assumed unchanged, while the in-plane shear stress is assumed to be reduced to zero.

For through the thickness matrix (delamination) failure given by equations f_{12} , the in-plane load carrying capacity within the element is assumed to be elastic, while the strength values for the tensile mode, $S_{ca}^{(0)}$ and $S_{bc}^{(0)}$, are set to zero. For tensile mode, $\sigma_c > 0$, the through the thickness stress components are reduced to zero. For compressive mode, $\sigma_c < 0$, σ_c is assumed to be elastic, while τ_{bc} and τ_{ca} 'slide' on the fracture strength surface as in an ideal plastic material, and the unloading path follows reduced shear moduli to the zero shear stress and strain state.

The effect of strain-rate on the layer strength values of the fiber failure modes is modeled by the strain-rate dependent functions for the strength values $\{I_{RT}\}$ as

$$\{S_{RT}\} = \{S_0\} \left(1 + C_{rate1} \ln \frac{\{\dot{\epsilon}\}}{\dot{\epsilon}_0} \right), \quad (23.161.16)$$

$$\{S_{RT}\} = \begin{Bmatrix} S_{aT} \\ S_{aC} \\ S_{bT} \\ S_{bC} \\ S_{FC} \\ S_{FS} \end{Bmatrix} \text{ and } \{\dot{\epsilon}\} = \begin{Bmatrix} |\dot{\epsilon}_a| \\ |\dot{\epsilon}_a| \\ |\dot{\epsilon}_b| \\ |\dot{\epsilon}_b| \\ |\dot{\epsilon}_c| \\ (\dot{\epsilon}_{ca}^2 + \dot{\epsilon}_{bc}^2)^{\frac{1}{2}} \end{Bmatrix}, \quad (23.161.17)$$

where C_{rate} is the strain-rate constants, and $\{S_0\}$ are the strength values of $\{S_{RT}\}$ at the reference strain-rate $\dot{\epsilon}_0$.

Damage Model

The damage model is a generalization of the layer failure model of Material 161 by adopting the MLT damage mechanics approach, Matzenmiller et al. [1995], for characterizing the softening behavior after damage initiation. Complete model description is given in Yen [2001]. The damage functions, which are expressed in terms of ply level engineering strains, are converted from the above failure criteria of fiber and matrix failure modes by neglecting the Poisson's effect. Elastic moduli reduction is expressed in terms of the associated damage parameters ω_i :

$$E'_i = (1 - \omega_i)E_i \quad (23.161.18)$$

$$\omega_i = 1 - \exp(-r_i^{m_i}/m_i) \quad r_i \geq 0 \quad i = 1, \dots, 6, \quad (23.161.19)$$

where E_i are the initial elastic moduli, E'_i are the reduced elastic moduli, r_i are the damage thresholds computed from the associated damage functions for fiber damage, matrix damage and delamination, and m_i are material damage parameters, which are currently assumed to be independent of strain-rate. The damage function is formulated to account for the overall nonlinear elastic response of a lamina including the initial 'hardening' and the subsequent softening beyond the ultimate strengths.

In the damage model (Material 162), the effect of strain-rate on the nonlinear stress-strain response of a composite layer is modeled by the strain-rate dependent functions for the elastic moduli $\{E_{RT}\}$ as

$$\{E_{RT}\} = \{E_0\} \left(1 + \{C_{rate}\} \ln \frac{\{\dot{\epsilon}\}}{\dot{\epsilon}_0} \right), \quad (23.161.20)$$

$$\{E_{RT}\} = \begin{Bmatrix} E_a \\ E_b \\ E_c \\ G_{ab} \\ G_{bc} \\ G_{ca} \end{Bmatrix}, \{\dot{\epsilon}\} = \begin{Bmatrix} |\dot{\epsilon}_a| \\ |\dot{\epsilon}_b| \\ |\dot{\epsilon}_c| \\ |\dot{\epsilon}_{ab}| \\ |\dot{\epsilon}_{bc}| \\ |\dot{\epsilon}_{ca}| \end{Bmatrix} \text{ and } \{C_{rate}\} = \begin{Bmatrix} C_{rate2} \\ C_{rate2} \\ C_{rate4} \\ C_{rate3} \\ C_{rate3} \\ C_{rate3} \end{Bmatrix}, \quad (23.161.21)$$

where $\{C_{rate}\}$ are the strain-rate constants. $\{E_0\}$ are the modulus values of $\{E_{RT}\}$ at the reference strain-rate $\dot{\epsilon}_0$.

23.163 Material Model 163: Modified Crushable Foam

The volumetric strain is defined in terms of the relative volume, V , as: $\gamma = 1 - V$. The relative volume is defined as the ratio of the current to the initial volume. In place of the effective plastic strain in the D3PLOT database, the integrated volumetric strain is output.

This material is an extension of material 63, *MAT_CRUSHABLE_FOAM. It allows the yield stress to be a function of both volumetric strain rate and volumetric strain. Rate effects are accounted for by defining a table of curves using *DEFINE_TABLE. Each curve defines the yield stress versus volumetric strain for a different strain rate. The yield stress is obtained by interpolating between the two curves that bound the strain rate.

To prevent high frequency oscillations in the strain rate from causing similar high frequency oscillations in the yield stress, a modified volumetric strain rate is used when interpolating to obtain the yield stress. The modified strain rate is obtained as follows. If NCYCLE is > 1 , then the modified strain rate is obtained by a time average of the actual strain rate over NCYCLE solution cycles. For SRCLMT > 0 , the modified strain rate is capped so that during each cycle, the modified strain rate is not permitted to change more than SRCLMT multiplied by the solution time step.

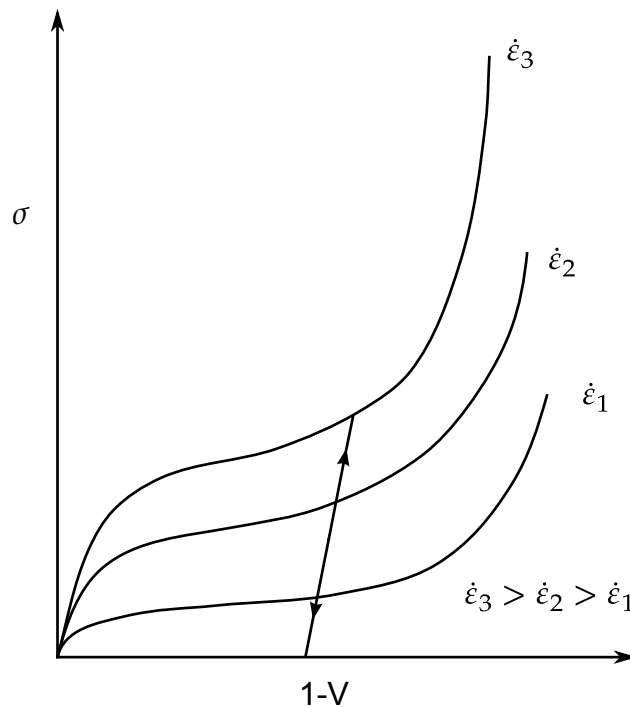


Figure 23.163.1. Rate effects are defined by a family of curves giving yield stress versus volumetric strain.

23.164 Material Model 164: Brain Linear Viscoelastic

The shear relaxation behavior is described by the Maxwell model as:

$$G(t) = G + (G_0 - G_\infty)e^{-\beta t}. \quad (23.164.1)$$

A Jaumann rate formulation is used

$$\overset{\nabla}{\sigma'}_{ij} = 2 \int_0^t G(t - \tau) D'_{ij}(\tau) dt. \quad (23.164.2)$$

where the prime denotes the deviatoric part of the stress rate, $\overset{\nabla}{\sigma'}_{ij}$, and the strain rate D_{ij} . For the Kelvin model the stress evolution equation is defined as:

$$\dot{s}_{ij} + \frac{1}{\tau} s_{ij} = (1 + \delta_{ij}) \left(G_0 + \frac{G_\infty}{\tau} \right) \dot{e}_{ij}. \quad (23.164.3)$$

The strain data as written to the LS-DYNA database may be used to predict damage, see [Bandak 1991].

23.166 Material Model 166: Moment Curvature Beam

Curvature rate can be decomposed into elastic part and plastic part:

$$\dot{\varepsilon} = \dot{\varepsilon}^e + \dot{\varepsilon}^p \Rightarrow \frac{\dot{\varepsilon}}{y} = \frac{\dot{\varepsilon}^e}{y} + \frac{\dot{\varepsilon}^p}{y} \Rightarrow \dot{\kappa} = \dot{\kappa}^e + \dot{\kappa}^p. \quad (23.166.1)$$

Moment rate is the product of elastic bending stiffness and elastic curvature:

$$\begin{aligned} \dot{M} &= \int_A \dot{\sigma} y dA = \int_A E^e \dot{\varepsilon}^e y dA = \int_A E^e \dot{\kappa}^e y^2 dA = \int_A E^e (\dot{\kappa} - \dot{\kappa}^p) y^2 dA \\ &= E^e (\dot{\kappa} - \dot{\kappa}^p) \int_A y^2 dA = (EI)^e (\dot{\kappa} - \dot{\kappa}^p). \end{aligned} \quad (23.166.2)$$

Plastic flow rule: $\psi = |M|$ (Isotropic hardening)

$$\dot{\kappa}^p = \dot{\lambda} \frac{\partial \psi}{\partial M} = \dot{\lambda} \text{sign}(M), \quad \dot{\kappa}^p = \sqrt{\dot{\kappa}^p \dot{\kappa}^p} = \dot{\lambda}. \quad (23.166.3)$$

Yield condition:

$$f = |M| - M_Y(\bar{\kappa}^p) = 0. \quad (23.166.4)$$

Loading and unloading conditions:

$$\dot{\lambda} \geq 0, f \leq 0, \dot{\lambda} f = 0. \quad (23.166.5)$$

Consistency condition:

$$\begin{aligned} \dot{f} &= 0 \Rightarrow \dot{M} \text{sign}(M) - \frac{\partial M_Y}{\partial \bar{\kappa}^p} \dot{\kappa}^p = 0 \\ \Rightarrow \dot{\lambda} &\equiv \dot{\kappa}^p \\ &= \frac{M \text{sign}(M)}{(EI)^p} = \frac{(EI)^e}{(EI)^p} (\dot{\kappa} - \dot{\kappa}^p) \text{sign}(M) \\ &= \frac{(EI)^e}{(EI)^p} [\dot{\kappa} - \dot{\lambda} \text{sign}(M)] \text{sign}(M) \\ \Rightarrow \dot{\lambda} &\equiv \dot{\kappa} = \frac{(EI)^e \dot{\kappa} \text{sign}(M)}{(EI)^p + (EI)^e} \end{aligned} \quad (23.166.6)$$

Moment rate is also the product of tangential bending stiffness and total curvature:

$$\dot{M} = (EI)^{ep} \dot{\kappa}. \quad (23.166.7)$$

Elastic, plastic, and tangential stiffnesses are obtained from user-defined curves:

$$(EI)^{ep} = \frac{dM}{d\kappa}, \quad (EI)^p = \frac{dM}{d\bar{\kappa}^p}. \quad (23.166.8)$$

Both are obtained from user-defined curves.

$$(EI)^e = \frac{(EI)^{ep}(EI)^p}{(EI)^p - (EI)^{ep}}. \quad (23.166.9)$$

For Torsion-Twist, simply replace M by T , κ by β , (EI) by (GJ) . For Force-Strain, simply replace M by N , κ by ε , (EI) by (EA) .

23.169 Material Model 169: Arup Adhesive

The through-thickness direction is identified from the smallest dimension of each element. It is expected that this dimension will be much smaller than in-plane dimensions (typically 2mm compared with 10mm).

In-plane stresses are set to zero: it is assumed that the stiffness and strength of the substrate is large compared with that of the adhesive, given the relative thicknesses. If the substrate is modeled with shell elements, it is expected that these will lie at the mid-surface of the substrate geometry. Therefore the solid elements representing the adhesive will be thicker than the actual bond. The yield and failure surfaces are treated as a power-law combination of direct tension and shear across the bond: $(\sigma/\sigma_{\max})^{\text{PWRT}} + (\tau/\tau_{\max})^{\text{PWRS}} = 1.0$ at yield. The stress-displacement curves for tension and shear are shown in the diagrams below. In both cases, G_c is the area under the curve. Because of the algorithm used, yielding in tension across the bond does not require strains in the plane of the bond – unlike the plasticity models, plastic flow is not treated as volume-conserving.

The Plastic Strain output variable has a special meaning:

- $0 < ps < 1$: ps is the maximum value of the yield function experienced since time zero
- $1 < ps < 2$: the element has yielded and the strength is reducing towards failure – yields at $ps = 1$, fails at $ps = 2$.

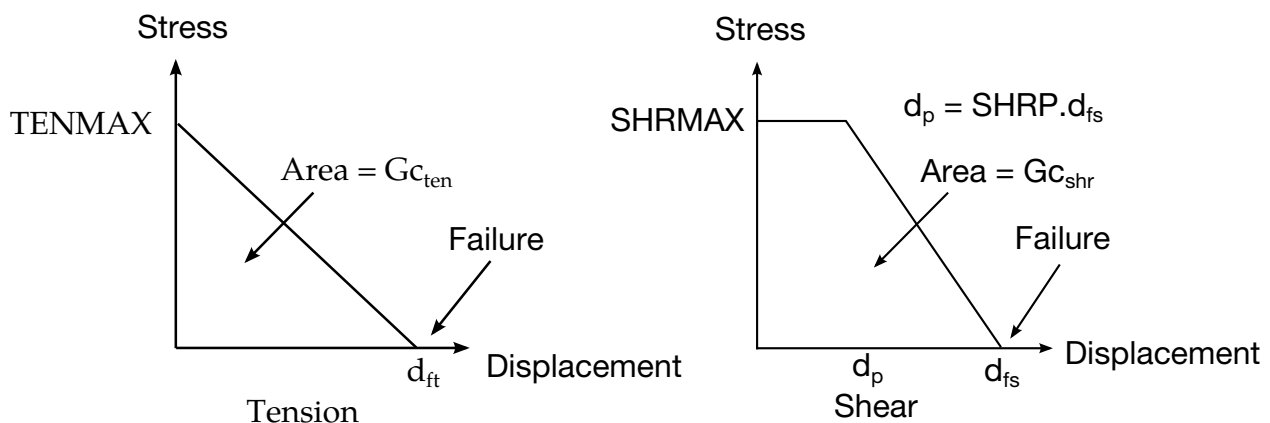


Figure 23.169.1.

23.170 Material Model 170: Resultant Anisotropic

The in-plane elastic matrix for in-plane plane stress behavior is given by:

$$\mathbf{C}_{\text{in plane}} = \begin{bmatrix} Q_{11p} & Q_{12p} & 0 & 0 & 0 \\ Q_{12p} & Q_{22p} & 0 & 0 & 0 \\ 0 & 0 & Q_{44p} & 0 & 0 \\ 0 & 0 & 0 & Q_{55p} & 0 \\ 0 & 0 & 0 & 0 & Q_{66p} \end{bmatrix}. \quad (23.170.1)$$

The terms Q_{ijp} are defined as:

$$\begin{aligned} Q_{11p} &= \frac{E_{11p}}{1 - \nu_{12p}\nu_{21p}}, \\ Q_{22p} &= \frac{E_{22p}}{1 - \nu_{12p}\nu_{21p}}, \\ Q_{12p} &= \frac{\nu_{12p}E_{11p}}{1 - \nu_{12p}\nu_{21p}}, \\ Q_{44p} &= G_{12p}, \\ Q_{55p} &= G_{23p}, \\ Q_{66p} &= G_{31p}. \end{aligned} \quad (23.170.2)$$

The elastic matrix for bending behavior is given by:

$$\mathbf{C}_{\text{bending}} = \begin{bmatrix} Q_{11b} & Q_{12b} & 0 \\ Q_{12b} & Q_{22b} & 0 \\ 0 & 0 & Q_{44b} \end{bmatrix}. \quad (23.170.3)$$

The terms Q_{ijb} are similarly defined.

23.175 Material Model 175: Viscoelastic Maxwell

Rate effects are taken into account through linear viscoelasticity by a convolution integral of the form:

$$\sigma_{ij} = \int_0^t g_{ijkl}(t - \tau) \frac{\partial \varepsilon_{kl}}{\partial \tau} d\tau, \quad (23.175.1)$$

where $g_{ijkl}(t - \tau)$ is the relaxation function for different stress measures. This stress is added to the stress tensor determined from the strain energy functional.

If we wish to include only simple rate effects, the relaxation function is represented by six terms from the Prony series:

$$g(t) = \sum_{m=1}^N G_m e^{-\beta_m t}. \quad (23.175.2)$$

We characterize this in the input by shear moduli, G_i , and the decay constants, β_i . An arbitrary number of terms, up to 6, may be used when applying the viscoelastic model.

For volumetric relaxation, the relaxation function is also represented by the Prony series in terms of bulk moduli:

$$k(t) = \sum_{m=1}^N K_m e^{-\beta_{k_m} t}. \quad (23.175.3)$$

The Arrhenius and Williams-Landau-Ferry (WLF) shift functions account for the effects of the temperature on the stress relaxation. A scaled time, t' ,

$$t' = \int_0^t \Phi(T) dt, \quad (23.175.4)$$

is used in the relaxation function instead of the physical time. The Arrhenius shift function is

$$\Phi(T) = \exp \left(-A \left\{ \frac{1}{T} - \frac{1}{T_{\text{REF}}} \right\} \right), \quad (23.175.5)$$

and the Williams-Landau-Ferry shift function is

$$\Phi(T) = \exp \left(-A \frac{T - T_{\text{REF}}}{B + T - T_{\text{REF}}} \right). \quad (23.175.6)$$

23.176 Material Model 176: Quasilinear Viscoelastic

The equations for this model are given as:

$$\begin{aligned}\sigma(t) &= \int_0^t G(t-\tau) \frac{\partial \sigma_\varepsilon[\varepsilon(\tau)]}{\partial \varepsilon} \frac{\partial \varepsilon}{\partial \tau} d\tau, \\ G(t) &= \sum_{i=1}^n G_i e^{-\beta_i t}, \\ \sigma_\varepsilon(\varepsilon) &= \sum_{i=1}^k C_i \varepsilon^i,\end{aligned}\tag{23.176.1}$$

where G is the shear modulus. In place of the effective plastic strain in the D3PLOT database, the effective strain is output:

$$\varepsilon^{\text{effective}} = \sqrt{\frac{2}{3} \varepsilon_{ij} \varepsilon_{ij}}.\tag{23.176.2}$$

The polynomial for instantaneous elastic response should contain only odd terms if symmetric tension-compression response is desired.

23.177 Material Models 177 and 178: Hill Foam and Viscoelastic Hill Foam

23.177.1 Hyperelasticity Using the Principal Stretch Ratios

Material types 177 and 178 in LS-DYNA are highly compressible Ogden models combined with viscous stress contributions. The latter model also allows for an additive viscoelastic stress contribution. As for the rate independent part, the constitutive law is determined by a strain energy function that is expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij} , as well as the constitutive tensor of interest, C_{ijkl}^{TC} , they are first calculated in the principal basis after which they are transformed back to the “base frame”, or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchhoff stress components are given by

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} \quad (\text{no sum}), \quad (23.177.1)$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau_{kl}^E. \quad (23.177.2)$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1} \tau_{ij}, \quad (23.177.3)$$

where $J = \lambda_1 \lambda_2 \lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchhoff stress in the principal basis can be expressed as

$$\left. \begin{aligned} C_{ijij}^{TKE} &= \lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} - 2 \tau_{ii}^E \delta_{ij} \\ C_{ijij}^{TKE} &= \frac{\lambda_j^2 \tau_{ii}^E - \lambda_i^2 \tau_{jj}^E}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \\ C_{ijij}^{TKE} &= \frac{\lambda_i}{2} \left(\frac{\partial \tau_{ii}^E}{\partial \lambda_i} - \frac{\partial \tau_{ii}^E}{\partial \lambda_j} \right), \quad i \neq j, \lambda_i = \lambda_j \end{aligned} \right\} \quad (\text{no sum}). \quad (23.177.4)$$

These components are transformed to the standard basis according to

$$C_{ijkl}^{TK} = q_{ip} q_{jq} q_{kr} q_{ls} C_{pqrs}^{TKE}, \quad (23.177.5)$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through

$$C_{ijkl}^{TC} = J^{-1} C_{ijkl}^{TK}. \quad (23.177.6)$$

23.177.2 Hill's Strain Energy Function

The strain energy function for materials 177 and 178 is given by

$$W = \sum_{m=1}^n \frac{\mu_m}{\alpha_m} \left[\lambda_1^{\alpha_m} + \lambda_2^{\alpha_m} + \lambda_3^{\alpha_m} - 3 + \frac{1}{n} (J^{-n\alpha_m} - 1) \right]. \quad (23.177.7)$$

where n , μ_m and α_m are material parameters. To apply the formulas in the previous section, we require

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} = \sum_{m=1}^n \frac{\mu_m}{J} (\lambda_i^{\alpha_m} - J^{-n\alpha_m}). \quad (23.177.8)$$

Proceeding with the constitutive tensor, we have

$$\lambda_j \frac{\partial \tau_{ii}^E}{\partial \lambda_j} = \sum_{m=1}^n \mu_m \alpha_m (\lambda_i^{\alpha_m} \delta_{ij} + n J^{-n\alpha_m}). \quad (23.177.9)$$

In addition to the hyperelastic stress described above, a viscous stress is added. Converting to Voigt notation, this stress can be written,

$$\sigma = \mathbf{C}\mathbf{D}, \quad (23.177.10)$$

where σ denotes Cauchy stress, \mathbf{D} is the rate-of-deformation and \mathbf{C} is an isotropic constitutive matrix representing the viscosity. In element m , the constitutive matrix depends on the element deformation according to

$$\mathbf{C} = \frac{d_m}{J} \mathbf{C}_0, \quad (23.177.11)$$

where d_m is the diameter⁵ of element m and \mathbf{C}_0 is a constitutive matrix that depends only on the material parameters. The stress contribution to the internal force can be written

$$f^{\text{int}} = \int_{\Omega_m} \mathbf{B}^T \sigma d\Omega_m, \quad (23.177.12)$$

and the corresponding material time derivative is

$$\dot{f}^{\text{mat}} = \int_{\Omega_m} \mathbf{B}^T \sigma^{\nabla T} d\Omega_m. \quad (23.177.13)$$

⁵ Experiments indicate that d_m is the smallest dimension of the element.

Here Ω_m is the current configuration of element m , \mathbf{B} is the strain-displacement matrix and ∇T denotes the Truesdell rate of Cauchy stress. The aim is to identify the material tangent modulus through

$$\dot{f}^{\text{mat}} = \int_{\Omega_m} \mathbf{B}^T \mathbf{C}^{\text{mat}} \mathbf{B} d\Omega_m \dot{u}, \quad (23.177.14)$$

for the viscous stress with \dot{u} being the nodal velocity. The Truesdell rate of the viscous stress can be written,

$$\sigma^{\nabla T} = \dot{\mathbf{C}}\mathbf{D} + \mathbf{C}\dot{\mathbf{D}} + \text{tr}(\mathbf{D})\sigma - \mathbf{L}\sigma - \sigma\mathbf{L}^T, \quad (23.177.15)$$

where \mathbf{L} is the velocity gradient. The terms on the right hand side can be treated as follows.

For the first term, we can assume that $d_m \propto J^{1/3}$ and then approximate

$$\dot{\mathbf{C}} = -\frac{2}{3}\text{tr}(\mathbf{D})\mathbf{C}. \quad (23.177.16)$$

Using [Equation \(23.177.10\)](#), [Equation \(23.177.13\)](#), the first term on the right hand side of [Equation \(23.177.15\)](#), [Equation \(23.177.16\)](#) and the expression

$$\mathbf{D} = \mathbf{B}\dot{\mathbf{u}}, \quad (23.177.17)$$

a material tangent modulus contribution can be identified in [Equation \(23.177.14\)](#) as

$$-\frac{2}{3}\sigma\delta^T, \quad (23.177.18)$$

where δ denotes the identity matrix in Voigt notation.

For the second term in [Equation \(23.177.15\)](#), we differentiate [Equation \(23.177.17\)](#) to see that

$$\dot{\mathbf{D}} = \dot{\mathbf{B}}\dot{\mathbf{u}} + \mathbf{B}\ddot{\mathbf{u}}. \quad (23.177.19)$$

Post-poning the treatment of the first term, the second of these two terms can be treated easily as this gives the following contribution to the material time derivative

$$\int_{\Omega_m} \mathbf{B}^T \mathbf{C} \mathbf{B} d\Omega_m \dot{u} \frac{\gamma}{\beta\Delta t}, \quad (23.177.20)$$

where γ and β are parameters in the Newmark scheme and Δt is the time step. From this expression, a material tangent modulus can through [Equation \(23.177.14\)](#) be identified as,

$$\mathbf{C}^{\text{mat}} = \frac{\gamma}{\beta\Delta t} \mathbf{C}. \quad (23.177.21)$$

The third term in [Equation \(23.177.15\)](#) contributes to the material tangent modulus as

$$\sigma \delta^T \quad (23.177.22)$$

resulting in a material tangent modulus given so far by

$$\frac{\gamma}{\beta \Delta t} \mathbf{C} + \frac{1}{3} \sigma \delta^T. \quad (23.177.23)$$

23.177.3 Viscous Stress

From the remaining terms, i.e., the last two terms in [Equation \(23.177.15\)](#) and the first term in [Equation \(23.177.19\)](#), we see it impossible to identify contributions to a material tangent modulus. We believe that these terms must be treated in some other manner. We are thus left with two choices, either to approximate these terms within the existing framework or to attempt a thorough implementation of the correct tangent stiffness using a different, and probably demanding, approach. We reason as follows.

Since this stress contribution is viscous and proportional to the mesh size, it is our belief that it serves as a stabilizing stress in the occurrence of a coarse mesh and/or large deformation rates, and really has little or nothing to do with the actual material models. If only the simulation process is slow (which it often is in an implicit analysis) and/or the mesh is sufficiently fine, this stress should be negligible compared to the other stress(es). With this in mind, we feel that it is not crucial to derive an exact tangent for this stress but we can be satisfied with an approximation. Even if attempting a more thorough derivation of the tangent stiffness, we would most certainly have to make approximations along the way. Hence we do not see this as an attractive approach.

In the implementation we have simply neglected all terms involving stresses since the experience from earlier work is that such terms generally have a negative effect on the tangent if they are not absolutely correct. In addition, most of the terms involving stresses contribute to a nonsymmetric tangent stiffness, which cannot be supported by LS-DYNA at the moment. Hence the material tangent modulus for the viscous stress is given by [Equation \(23.177.21\)](#). We are aware of that this may be a crude approximation, and if experiments show that it is a poor one, we will take a closer look at it.

In material type 178, the viscous stress acts only in the direction of the principal stretches and in compression. With \mathbf{C} being an isotropic tensor, we evaluate the tangent stiffness modulus in the principal basis according to [Equation \(23.177.21\)](#), modify it to account for the mentioned conditions and then transform it back to the global frame of reference.

23.177.4 Viscoelastic Stress Contribution

For material 178, an optional viscoelastic stress contribution can be added. The evolution of this stress in time can be stated as

$$\sigma_{ij}^{\nabla} = \sum_{m=1}^{12} 2G_m s_{ij}^{m\nabla}, \quad (23.177.24)$$

where

$$s_{ij}^{m\nabla} = D_{ij} - \beta_m s_{ij}^m, \quad (23.177.25)$$

Here G_m and β_m are material constants, and D_{ij} is the rate-of-deformation tensor. Referring to Borrvall [2002], we state that the tangent stiffness modulus for this stress contribution can be written

$$C_{ijkl} = \sum_{m=1}^{12} G_m (\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \quad (23.177.26)$$

Just as for the viscous stress, this stress acts only in the direction of the principal stretches. Hence the tangent modulus is formed in the principal basis, modified to account for this condition and then transformed back to the global frame of reference.

23.177.5 Material Tangent Modulus for the Fully Integrated Brick

To avoid locking tendencies for the fully integrated brick element in LS-DYNA, the stress is modified as

$$\sigma_{S/R} = \sigma + (p - \bar{p})\mathbf{I}, \quad (23.177.27)$$

where p is the pressure and \bar{p} is the mean pressure in the element. This affects the tangent stiffness since one has to take into account that the pressure is constant in the element. Deriving the material time derivative of the internal force results in

$$\begin{aligned} \dot{f}^{\text{mat}} = & \int_{\Omega_m} \mathbf{B}^T \mathbf{C} \mathbf{B} d\Omega_m \dot{u} + \int_{\Omega_m} (p - \bar{p}) \mathbf{B}^T (\mathbf{I} \otimes \mathbf{I}) \mathbf{B} d\Omega_m \dot{u} \\ & - 2 \int_{\Omega_m} (p - \bar{p}) \mathbf{B}^T \mathbf{B} d\Omega_m \dot{u} + \int_{\Omega_m} (\dot{p} - \dot{\bar{p}}) \mathbf{B}^T d\Omega_m. \end{aligned} \quad (23.177.28)$$

To implement this tangent, the last term is the most difficult to deal with as it involves the time derivative (or variation) of the pressure. For certain types of material models, for instance material type 77 in LS-DYNA, the pressure is a function of the relative volume

$$p = p(J), \quad (23.177.29)$$

and with the approximation

$$\bar{p} = p(\bar{J}), \quad (23.177.30)$$

the last term can be evaluated to

$$\int_{\Omega_m} J p'(J) \mathbf{B}^T (\mathbf{I} \otimes \mathbf{I}) \mathbf{B} d\Omega_m \dot{u} - \int_{\Omega_m} \bar{J} p'(\bar{J}) \bar{\mathbf{B}}^T (\mathbf{I} \otimes \mathbf{I}) \bar{\mathbf{B}} d\Omega_m \dot{u}, \quad (23.177.31)$$

and a symmetric tangent stiffness can quite easily be implemented. We have here used \bar{J} and $\bar{\mathbf{B}}$ for the mean values of J and \mathbf{B} , respectively. For other types of material models, such as the ones described in this document or material type 27 in LS-DYNA, the expression for the pressure is more complicated. A characterizing feature is that a non-zero pressure can occur under constant volume. This will in general complicate the implementation of the last term and will also contribute to a non-symmetric tangent stiffness that cannot be handled in LS-DYNA at the moment. For material 27, neglecting this had a tremendous impact on the performance of the implicit solution procedure, (see material type 27). For the current material models, it seems to be of less importance, and we believe that this is due to the higher compressibility allowed.

23.177.6 Viscous damping

Viscous damping in the model follows an implementation identical to that of material type 57.

23.179 Material Models 179 and 180: Low Density Synthetic Foam

Material types 179 and 180 in LS-DYNA are highly compressible synthetic foam models with no Poisson's ratio effects combined with an optional visco-elastic and a stabilizing viscous stress contribution. The tensile behavior of the materials is linear where the stress cannot exceed a user prescribed cutoff stress. In compression the materials show a hysteresis on unloading similar to material 57. In addition, the first load cycle damages the material so that the stress level on reloading is significantly reduced. For material 179 the damage is isotropic while it is orthotropic for material 180. Viscous damping in the model follows an implementation identical to that of material type 57.

23.179.1 Hyperelasticity Using the Principal Stretch Ratios

As for the rate independent part of the stress, the constitutive law is mainly determined by a strain energy function that is expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij} , as well as the constitutive tensor of interest, C_{ijkl}^{TC} , they are first calculated in the principal basis after which they are transformed back to the "base frame", or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here.

The principal Kirchhoff stress components are given by

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} \quad (\text{no sum}), \quad (23.179.1)$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik} q_{jl} \tau_{kl}^E. \quad (23.179.2)$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by ,

$$\sigma_{ij} = J^{-1} \tau_{ij}, \quad (23.179.3)$$

where $J = \lambda_1 \lambda_2 \lambda_3$ is the relative volume change.

The constitutive tensor that relates the rate of deformation to the Truesdell (convected) rate of Kirchhoff stress can in the principal basis be expressed as

$$\begin{aligned}
C_{ijij}^{\text{TKE}} &= \lambda_j \frac{\partial \tau_{ii}^{\text{E}}}{\partial \lambda_j} - 2\tau_{ii}^{\text{E}} \delta_{ij} \\
C_{ijij}^{\text{TKE}} &= \frac{\lambda_j^2 \tau_{ii}^{\text{E}} - \lambda_i^2 \tau_{jj}^{\text{E}}}{\lambda_i^2 - \lambda_j^2}, \quad i \neq j, \lambda_i \neq \lambda_j \quad (\text{no sum}). \\
C_{ijij}^{\text{TKE}} &= \frac{\lambda_i}{2} \left(\frac{\partial \tau_{ii}^{\text{E}}}{\partial \lambda_i} - \frac{\partial \tau_{ii}^{\text{E}}}{\partial \lambda_j} \right), \quad i \neq j, \lambda_i = \lambda_j
\end{aligned} \tag{23.179.4}$$

These components are transformed to the standard basis according to

$$C_{ijkl}^{\text{TK}} = q_{ip} q_{jq} q_{kr} q_{ls} C_{pqrs}^{\text{TKE}}, \tag{23.179.5}$$

and finally the constitutive tensor relating the rate of deformation to the Truesdell rate of Cauchy stress is obtained through.

$$C_{ijkl}^{\text{TC}} = J^{-1} C_{ijkl}^{\text{TK}}. \tag{23.179.6}$$

23.179.2 Strain Energy Function

The strain energy function for materials 179 and 180 is given by

$$W = \sum_{m=1}^3 w(\lambda_m), \tag{23.179.7}$$

where

$$w(\lambda) = \begin{cases} s \left(\lambda - 1 - \frac{s}{2E} \right) & \text{if } \lambda \geq \frac{s}{E} + 1 \\ \frac{E}{2} (\lambda - 1)^2 & \text{if } 1 \leq \lambda < \frac{s}{E} + 1 \\ \int_1^\lambda f_s(1 - \mu) d\mu & \text{otherwise} \end{cases} \tag{23.179.8}$$

Here s is the nominal tensile cutoff stress and E is the stiffness coefficient relating a change in principal stretch to a corresponding change in nominal stress. The function $f_s(\leq 0)$ gives the nominal compressive stress as a function of the strain in compression for the second and all subsequent load cycles and is supplied by the user. To apply the formulas in the previous section, we require

$$\tau_{ii}^{\text{E}} = \lambda_i \frac{\partial w}{\partial \lambda_i} = \begin{cases} s \lambda_i & \text{if } \lambda_i \geq \frac{s}{E} + 1 \\ E \lambda_i (\lambda_i - 1) & \text{if } 1 \leq \lambda_i < \frac{s}{E} + 1 \\ \lambda_i f_s(1 - \lambda_i) & \text{otherwise} \end{cases} \tag{23.179.9}$$

Proceeding with the constitutive tensor, we have

$$\lambda_j \frac{\partial \tau_{ij}^E}{\partial \lambda_j} = \delta_{ij} \begin{cases} s\lambda_i & \text{if } \lambda_i \geq \frac{s}{\bar{E}} + 1 \\ E\lambda_i(2\lambda_i - 1) & \text{if } 1 \leq \lambda_i < \frac{s}{\bar{E}} + 1 \\ \lambda_i(f_s(1 - \lambda_i) - \lambda_i f'_s(1 - \lambda_i)) & \text{otherwise} \end{cases} \quad (23.179.10)$$

23.179.3 Modeling of the Hysteresis

The hyperelastic part of the Cauchy stress is scaled by a factor κ given by

$$\kappa = \frac{E}{\bar{E}}, \quad (23.179.11)$$

where

$$E = \int J \sigma : d\epsilon, \quad (23.179.12)$$

is the stored energy in the material and

$$\bar{E} = E_{\max} \exp(-\beta(t - s)). \quad (23.179.13)$$

Here s stands for the time point when E has its maximum E_{\max} in the interval $[0, t]$. The factor κ is introduced to model the hysteresis that characterizes this material (and material 57). The decay coefficient β is introduced to get a reloading curve similar to the original loading curve.

This factor κ is treated as a constant in the determination of the tangent stiffness matrix.

23.179.4 Viscous Stress

In addition to the hyperelastic stress described above, a viscous stress is added. Converting to Voigt notation, this stress can be written

$$\sigma = \mathbf{C}\mathbf{D}, \quad (23.179.14)$$

where σ denotes Cauchy stress, \mathbf{D} is the rate-of-deformation and \mathbf{C} is an isotropic constitutive matrix representing the viscosity. In element m , the constitutive matrix depends on the element deformation according to

$$\mathbf{C} = \frac{d_m}{J} \mathbf{C}_0, \quad (23.179.15)$$

where d_m is the diameter⁶ of element m and \mathbf{C}_0 is a constitutive matrix that depends only on the material parameters. Following material models 177 and 178 we use the following material tangent stiffness for this stress contribution

⁶ Experiments indicate that d_m is the smallest dimension of the element.

$$\mathbf{C}^{\text{mat}} = \frac{\gamma}{\beta \Delta t} \mathbf{C}, \quad (23.179.16)$$

where γ and β are parameters in the Newmark scheme and Δt is the time step.

23.179.5 Viscoelastic Stress Contribution

An optional viscoelastic stress contribution can be added. The evolution of this stress in time can be stated as

$$\sigma_{ij}^{\nabla} = E_d s_{ij}^{\nabla}, \quad (23.179.17)$$

where

$$s_{ij}^{\nabla} = D_{ij} - \beta_1 s_{ij}. \quad (23.179.18)$$

Here E_d and β_1 are material constants, D_{ij} is the rate-of-deformation tensor and ∇ stands for an objective rate. Referring to material models 177 and 178, we state that the tangent stiffness modulus for this stress contribution can be written

$$C_{ijkl} = \frac{E_d}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}). \quad (23.179.19)$$

This stress acts only in the direction of the principal stretches. Hence the tangent modulus is formed in the principal basis, modified to account for this condition and then transformed back to the global frame of reference.

23.179.6 Stress Corresponding to First Load Cycle

We define a contribution to the principal Kirchhoff stress as

$$^1 \tau_{ii}^E = \lambda_i \{g_s(1 - \lambda_i) - f_s(1 - \lambda_i)\} \xi. \quad (23.179.20)$$

When the damage is isotropic the factor ξ is given by

$$\xi = \max \left(0, 1 - \frac{\varepsilon_h}{0.0001 + \varepsilon_m} \right). \quad (23.179.21)$$

where ε_h is the damage parameter that is initially zero and ε_m is the maximum compressive volumetric strain during the entire simulation thus far. Damage evolves when the material is in compression and unloads

$$\Delta \varepsilon_h = \begin{cases} 0 & \text{if } J \geq 1 \\ \max(0, \Delta J) & \text{otherwise} \end{cases} \quad (23.179.22)$$

where J is the jacobian of the deformation. The first load cycle will result in a total stress that follows load curve g_s since there is no damage. After a complete load cycle, i.e., unloading has occurred, the material is completely damaged, i.e., $\varepsilon_h \approx \varepsilon_m$, and the nominal stress will for the second and subsequent load cycles be given by the load curve f_s .

In the orthotropic case the principal Kirchhoff stress contribution is instead given by

$$^1 \tau_{ii}^E = \lambda_i \{g_s(1 - \lambda_i) - f_s(1 - \lambda_i)\} \zeta_i. \quad (23.179.23)$$

For the damage to be orthotropic we introduce a symmetric and positive definite damage tensor ε_h^{ij} . This tensor is initially the zero tensor corresponding to no damage. The evolution of damage begins with a half step Jaumann rotation of the tensor to maintain objectivity. After that the local increment is performed. As for the isotropic case, damage evolves in compression in combination with unloading. We introduce the local damage increment as

$$\Delta \varepsilon_{loc}^{ij} = \delta_{ij} \begin{cases} 0 & \text{if } \lambda_i \geq 1 \\ \max(0, \Delta \lambda_i) & \text{otherwise} \end{cases} \quad (23.179.24)$$

which is a diagonal tensor. The global damage tensor increment is given by

$$\Delta \varepsilon_h^{ij} = q_{ik} q_{jl} \Delta \varepsilon_{loc}^{kl}, \quad (23.179.25)$$

which is used to increment the damage tensor ε_h^{ij} . The factor ζ_i is now given by

$$\zeta_i = \max \left(0, 1 - \frac{q_{ki} q_{li} \varepsilon_h^{kl}}{0.0001 + \varepsilon_m} \right). \quad (23.179.26)$$

where the quantity ε_m in the orthotropic case is the maximum compressive principal strain in any direction during the simulation thus far. As for the isotropic case, the material is completely damaged after one load cycle and reloading will follow load curve f_s . In addition, the directions corresponding to no loading will remain unaffected.

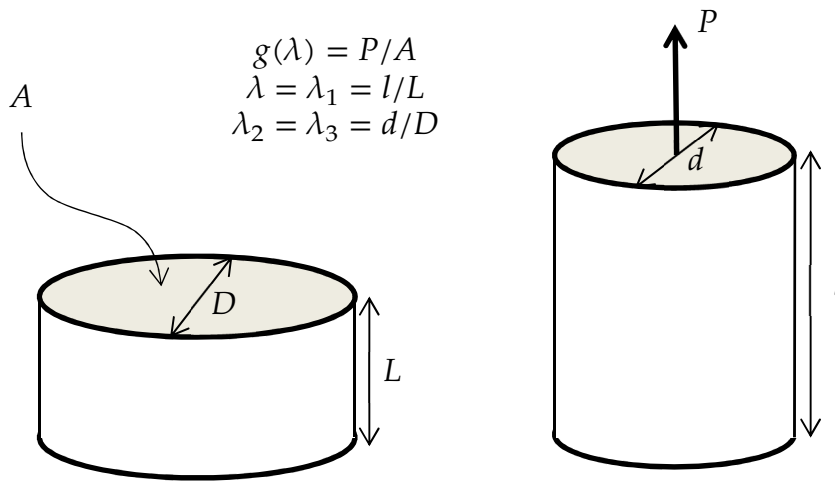
The factors ζ and ζ_i are treated as constants in the determination of the tangent stiffness so the contribution is regarded as hyperelastic and follows the exposition given in Section 19.179.1.

The reason for not differentiating the coefficients κ , ζ and ζ_i is that they are always non-differentiable. Their changes depend on whether the material is loaded or unloaded, i.e., the direction of the load. Even if they were differentiable their contributions would occasionally result in a non-symmetric tangent stiffness matrix and any attempt to symmetrize this would probably destroy its properties. After all, we believe that the one-dimensional nature and simplicity of this foam will be enough for good convergence properties even without differentiating these coefficients.

23.181 Material Model 181: Simplified Rubber/Foam

Material type 181 in LS-DYNA is a simplified “quasi”-hyperelastic rubber or foam model defined by a single uniaxial load curve or by a family of curves at discrete strain rates. The term “quasi” is used because there is really no strain energy function for determining the stresses used in this model. Rather the stress response mimics the gradient of the strain energy potential in the Ogden rubber (see material 77). For deriving the tangent stiffness matrix we use the formulas as if a strain energy function were present, with appropriate modifications.

This model is equipped with various features related to dissipation and damage, but not all of those are described in detail.



23-123 Uniaxial test parameters.

23.181.1 Hyperelasticity Using the Principal Stretch Ratios

A hyperelastic constitutive law is determined by a strain energy function that we assume is expressed in terms of the principal stretches, i.e., $W = W(\lambda_1, \lambda_2, \lambda_3)$. To obtain the Cauchy stress σ_{ij} this is first calculated in the principal basis after which it is transformed back to the “base frame”, or standard basis. The complete set of formulas is given by Crisfield [1997] and is for the sake of completeness recapitulated here. For the following discussion we refer to [figure 23-123](#)

The principal Kirchhoff stress components are given by

$$\tau_{ii}^E = \lambda_i \frac{\partial W}{\partial \lambda_i} \text{ (no sum),} \quad (23.181.1)$$

that are transformed to the standard basis using the standard formula

$$\tau_{ij} = q_{ik}q_{jl}\tau_{kl}^E \quad (23.181.2)$$

The q_{ij} are the components of the orthogonal tensor containing the eigenvectors of the principal basis. The Cauchy stress is then given by

$$\sigma_{ij} = J^{-1}\tau_{ij}, \quad (23.181.3)$$

where $J = \lambda_1\lambda_2\lambda_3$ is the relative volume change.

Now, the Ogden strain energy potential results in a Kirchhoff stress on the form

$$\tau_{ii}^E = f(\tilde{\lambda}_i) + K_m(J - 1) - \frac{1}{3}\sum_{k=1}^3 f(\tilde{\lambda}_k) \quad (23.4)$$

for a (large) bulk modulus K_m and where $\tilde{\lambda}_i = \lambda_i/J^{1/3}$ are the isochoric stretches. In the Ogden material, f has a specific form *a priori* that requires a least square approximation for fitting test data. This is of course a restriction and the idea in material 181 is to let f be determined directly from input data. The ansatz for the compressible foam option is to let

$$\tau_{ii}^E = f(\lambda_i) - f\left(J^{-\frac{\nu}{1-2\nu}}\right) \quad (23.5)$$

for a given Poisson's ratio ν , a decision that will be made clear below. So assume that $g(\lambda)$ is the curve providing the engineering stress as function of stretch in a uniaxial test, see [figure 23-123](#), then the principal Kirchhoff stresses are

$$\begin{aligned} \tau_{11}^E &= \lambda g(\lambda) \\ \tau_{22}^E &= \tau_{33}^E = 0 \end{aligned} \quad (23.6)$$

What follows is the determination of the internal function f for the rubber and foam option.

23.181.1.1 Determination of f , rubber option

For incompressibility we deduce that the principal stretches are

$$\begin{aligned} \lambda_1 &= \lambda \\ \lambda_2 &= \lambda_3 = \lambda^{-1/2} \end{aligned} \quad (23.7)$$

which coincide with the isochoric counterparts. Using these expressions when equating (23.4) and (23.6), one must determine f from

$$\begin{aligned} \lambda g(\lambda) &= \frac{2}{3}\left(f(\lambda) - f(\lambda^{-1/2})\right) + K_m(J - 1) \\ 0 &= \frac{1}{3}\left(f(\lambda^{-1/2}) - f(\lambda)\right) + K_m(J - 1). \end{aligned} \quad (23.8)$$

By subtracting these two equations to eliminate the influence of the pressure we get

$$\lambda g(\lambda) = f(\lambda) - f\left(\lambda^{-\frac{1}{2}}\right) \quad (23.9)$$

that can be rewritten as

$$f(\lambda) = \lambda g(\lambda) + f(\lambda^{-1/2}). \quad (23.10)$$

This in turn can be recursively expanded as

$$f(\lambda) = \lambda g(\lambda) + \lambda^{-1/2} g(\lambda^{-1/2}) + \lambda^{1/4} g(\lambda^{1/4}) + \dots + f(\lambda^{(-1/2)^n}) \quad (23.11)$$

and by letting n be large enough the function f can be determined since the last term tends to zero.

23.181.1.2 Determination of f , foam option

Similarly, for a given Poisson's ratio ν , the principal stretches in a uniaxial tension test are

$$\begin{aligned} \lambda_1 &= \lambda \\ \lambda_2 &= \lambda_3 = \lambda^{-\nu} \end{aligned} \quad (23.12)$$

and using (23.5) and (23.6) the equation to solve is now

$$\lambda g(\lambda) = f(\lambda) - f(\lambda^{-\nu}) \quad (23.13)$$

Note that the equation corresponding to the second of (23.6) vanishes because of the ansatz in (23.5). The same technique as for the rubber option is used, recursive expansion gives

$$f(\lambda) = \lambda g(\lambda) + \lambda^{-\nu} g(\lambda^{-\nu}) + \lambda^{\nu^2} g(\lambda^{\nu^2}) + \dots + f(\lambda^{(-\nu)^n}) \quad (23.14)$$

which for a large n gives a sufficiently accurate representation of f .

23.181.2 Some Remarks

23.181.2.1 Strain rates

The function f introduced in the previous section depends not only on the stretches but for some choices of input also on the strain rate. In this case each test curve $g_i(\lambda)$ corresponding to a particular strain rate $\dot{\epsilon}_i$ is converted to an internal function $f_i(\lambda)$ following the procedure described in the previous section. These internal functions are then used for determining the response for a given strain rate $\dot{\epsilon}$ by interpolation. Strain rates are treated in various ways depending on user defined parameters and we refer to Section and the Keyword Manual for more info.

23.181.2.2 Modeling of the Frequency Independent Damping

An elastic-plastic stress σ_d is added to model the frequency independent damping properties of rubber. This stress is deviatoric and determined by the shear modulus G and the yield stress σ_Y . This part of the stress is updated incrementally as

$$\tilde{\sigma}_d^{n+1} = \sigma_d^n + 2GI^{\text{dev}} \Delta \epsilon, \quad (23.181.15)$$

where $\Delta \epsilon$ is the strain increment. The trial stress is then radially scaled (if necessary) to the yield surface according to

$$\sigma_d^{n+1} = \tilde{\sigma}_d^{n+1} \min \left(1, \frac{\sigma_Y}{\sigma_{\text{eff}}} \right), \quad (23.181.16)$$

where σ_{eff} is the effective von Mises stress for the trial stress $\tilde{\sigma}_d^{n+1}$.

23.187 Material Model 187: Semi-Analytical Model for the Simulation of Polymers

23.187.1 Material law formulation

Choice of a yield surface formulation

All plastics are to some degree anisotropic. The anisotropic characteristic can be due to fibre reinforcement, to the moulding process or it can be load induced in which case the material is at least initially isotropic. Therefore a quadratic form in the stress tensor is often used to describe the yield surface. We restrict the scope of this work to isotropic formulations. However, the choice of this yield surface was made in view of later anisotropic generalisations. In the isotropic case the most general quadratic yield surface can be written as

$$f = \sigma^T \mathbf{F} \sigma + \mathbf{B} \sigma + F_0 \leq 0, \quad (23.187.1)$$

where

$$\begin{aligned} \sigma &= \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{pmatrix}, \\ \mathbf{F} &= \begin{pmatrix} F_{11} & F_{12} & F_{12} & 0 & 0 & 0 \\ F_{12} & F_{11} & F_{12} & 0 & 0 & 0 \\ F_{12} & F_{12} & F_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & F_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & F_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & F_{44} \end{pmatrix}, \\ \mathbf{B} &= \begin{pmatrix} F_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & F_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \quad (23.187.2)$$

Some restrictions apply to the choice of the coefficients. The existence of a stress-free state and the equivalence of pure shear and biaxial tension/compression require respectively

$$F_0 \leq 0 \text{ and } F_{44} = 2(F_{11} - F_{12}). \quad (23.187.3)$$

Although 4 independent coefficients remain in the expression for the isotropic yield surface at this point, however the yield condition is not affected if all coefficients are multiplied by a constant. Consequently only 3 coefficients can be freely chosen and 3 experiments under different states of stress can be fitted by this formulation.

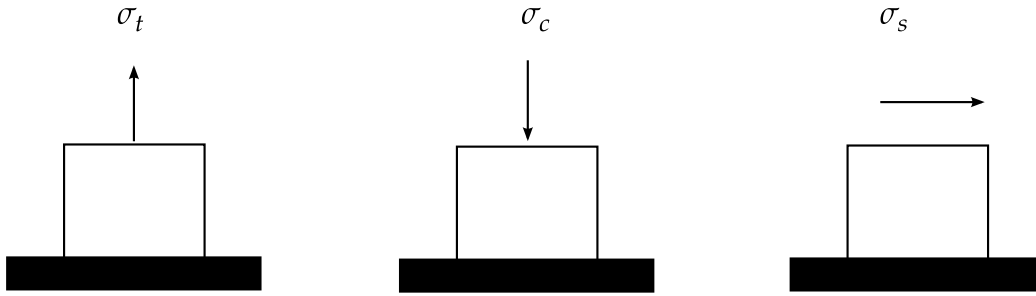


Figure 23.1. Recommended tests for material data in SAMP

Without loss of generality the expression for the yield surface can be reformulated in terms of the first two stress invariants: pressure and von Mises stress:

$$p = -\frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3},$$

$$\sigma_{vm} = \sqrt{\frac{3}{2} \left((\sigma_{xx} + p)^2 + (\sigma_{yy} + p)^2 + (\sigma_{zz} + p)^2 + 2\sigma_{xy}^2 + 2\sigma_{yz}^2 + 2\sigma_{zx}^2 \right)}. \quad (23.187.4)$$

The expression for the yield surface then becomes

$$f = \sigma_{vm}^2 - A_0 - A_1 p - A_2 p^2 \leq 0, \quad (23.187.5)$$

and identification of the coefficients gives

$$A_0 = -F_0, \quad A_1 = 3F_1 \quad \text{and} \quad A_2 = 9(1 - F_{11}), \quad (23.187.6)$$

or equivalently

$$F_0 = -A_0, F_1 = \frac{A_1}{3}, F_{11} = 1 - \frac{A_2}{9}, F_{44} = 3 \quad \text{and} \quad F_{12} = F_{11} - \frac{F_{44}}{2} = -\left(\frac{1}{2} + \frac{A_2}{9}\right). \quad (23.187.7)$$

Since there is no loss of generality, the simpler formulation in invariants is adopted from this point on. In principle the coefficients of the yield surface can now be determined from 3 experiments. Typically we would perform uniaxial tension, uniaxial compression and simple shear tests:

This allows computation of the coefficients in function of the test results:

$$\left. \begin{aligned} 3\sigma_s^2 &= A_0 \\ \sigma_t^2 &= 3\sigma_s^2 - A_1 \frac{\sigma_t}{3} + A_2 \frac{\sigma_t^2}{9} \\ \sigma_c^2 &= 3\sigma_s^2 + A_1 \frac{\sigma_c}{3} + A_2 \frac{\sigma_c^2}{9} \end{aligned} \right\} \Rightarrow \begin{cases} A_0 = 3\sigma_s^2 \\ A_1 = 9\sigma_s^2 \left(\frac{\sigma_c - \sigma_t}{\sigma_c \sigma_t} \right) \\ A_2 = 9 \left(\frac{\sigma_c \sigma_t - 3\sigma_s^2}{\sigma_c \sigma_t} \right) \end{cases}. \quad (23.187.8)$$

Alternatively we can also compute the coefficients relating to the formulation in stress space:

$$\left. \begin{aligned} F_0 + F_1\sigma_t + F_{11}\sigma_t^2 &= 0 \\ F_0 - F_1\sigma_c + F_{11}\sigma_c^2 &= 0 \\ F_0 + F_{44}\sigma_s^2 &= 0 \end{aligned} \right\} \Rightarrow \begin{cases} F_1 = F_0 \left(\frac{1}{\sigma_c} - \frac{1}{\sigma_t} \right) \\ F_{11} = -\frac{F_0}{\sigma_t\sigma_c} \\ F_{44} = -\frac{F_0}{\sigma_s^2} \end{cases} . \quad (23.187.9)$$

Both are easily seen to be equivalent.

Conditions for convexity of the yield surface

Usually the yield surface is required to be convex, i.e.

$$\left. \begin{aligned} f(\sigma_1) &\leq 0 \\ f(\sigma_2) &\leq 0 \\ 0 &\leq \alpha \leq 1 \end{aligned} \right\} \Rightarrow f(\alpha\sigma_1 + (1-\alpha)\sigma_2) \leq 0. \quad (23.187.10)$$

The second derivative of f is computed as

$$f = \sigma^T \mathbf{F} \sigma + \mathbf{B} \sigma + F_0 \quad \rightarrow \quad \frac{\partial^2 f}{\partial \sigma^2} = 2\mathbf{F} \quad (23.187.11)$$

A sufficient condition for convexity in 6D stress space is then that the matrix \mathbf{F} should be positive semidefinite. This means all eigenvalues of \mathbf{F} should be positive or zero. The conditions for convexity will now be examined in physical terms for two cases: plane stress and general 3D.

The plane stress case

In the plane stress case the yield condition reduces to:

$$f = \sigma^T \mathbf{F} \sigma + \mathbf{B} \sigma + F_0, \quad (23.187.12)$$

where

$$\sigma = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} F_{11} & F_{12} & 0 \\ F_{12} & F_{11} & 0 \\ 0 & 0 & F_{44} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} F_1 & 0 & 0 \\ 0 & F_1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (23.187.13)$$

And convexity requires the eigenvalues of \mathbf{F} to be non-negative:

$$\left. \begin{aligned} F_{11} + F_{12} &\geq 0 \\ F_{11} - F_{12} &\geq 0 \\ F_{44} &\geq 0 \end{aligned} \right\} \Rightarrow \begin{cases} 4\sigma_s^2 \geq \sigma_t\sigma_c \\ -F_0 \geq 0 \end{cases}. \quad (23.187.14)$$

The 3D case

In the full 3D case, the convexity condition is generally more stringent. Again we require the eigenvalues of \mathbf{F} to be non-negative, where \mathbf{F} is now the full 6 by 6 matrix:

$$\left. \begin{aligned} F_{11} + 2F_{12} &\geq 0 \\ F_{11} - F_{12} &\geq 0 \\ F_{44} &\geq 0 \end{aligned} \right\} \Rightarrow \begin{cases} 3\sigma_s^2 \geq \sigma_t\sigma_c \\ -F_0 \geq 0 \end{cases}. \quad (23.187.15)$$

Leading to

$$\sigma_s \geq \frac{\sqrt{\sigma_t \sigma_c}}{\sqrt{3}} > \frac{\sqrt{\sigma_t \sigma_c}}{2}. \quad (23.187.16)$$

Alternatively a yield surface containing a linear rather than a quadratic term was implemented in SAMP-1.

$$f = \sigma_{vm} - A_0 - A_1 p - A_2 p^2 \leq 0. \quad (23.187.17)$$

As it will be difficult in general to guarantee a reasonable flow behaviour from three independent measurements in shear, tension and compression, a simplified flow rule has been implemented as the default in SAMP-1. The generally non-associated flow surface is given as:

$$g = \sigma_{vm}^2 + \alpha p^2. \quad (23.187.18)$$

This flow rule is associated if:

$$\begin{aligned} A_1 &= 0, \\ A_2 &= -\alpha \quad (= \text{cte}). \end{aligned} \quad (23.187.19)$$

And clearly leads to a constant value for the plastic Poisson ratio:

$$v_p = \frac{9 - 2\alpha}{18 + 2\alpha} \Rightarrow \alpha = \frac{9}{2} \frac{1 - v_p}{1 + v_p}. \quad (23.187.20)$$

Plausible flow behaviour just means that:

$$0 \leq \alpha \leq \frac{9}{2} \Rightarrow 0 \leq v_p \leq 0.5. \quad (23.187.21)$$

In SAMP-1 the value of the plastic Poisson coefficient is given by the user, either as a constant or as a load curve in function of the uniaxial plastic strain. This allows adjusting the flow rule of the material to measurements of transversal deformation during uniaxial tensile or compressive testing. This can be important for plastics since often a non-isochoric behaviour is measured.

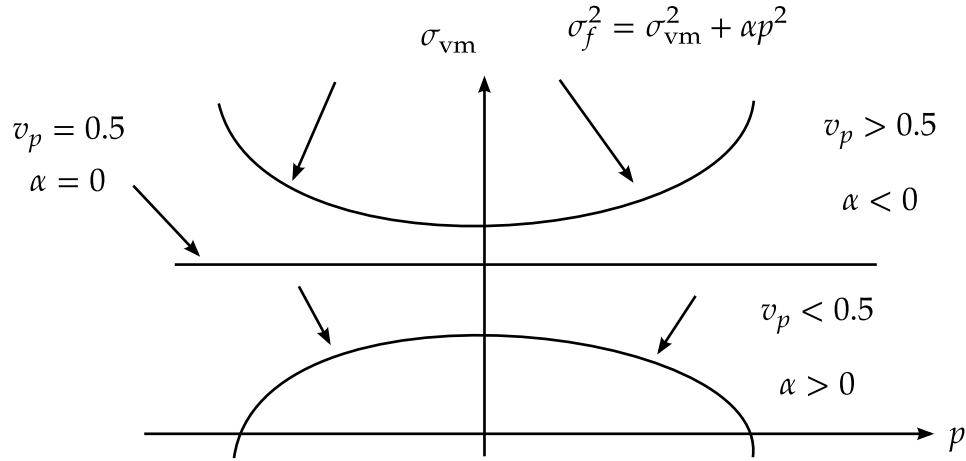


Figure 23.2. Influence of the flow rule on the plastic Poisson ratio

The possible values for the plastic Poisson ratio and the resulting flow behaviour are illustrated in [Figure 23.2](#).

In SAMP-1 the formulation is slightly modified and based on a flow rule given as:

$$g' = \sqrt{\sigma_{vm}^2 + \alpha p^2}. \quad (23.187.22)$$

The plastic strain rate computation is not normalized:

$$\dot{\epsilon}_p = \dot{\lambda} \frac{\partial g'}{\partial \sigma}. \quad (23.187.23)$$

The volumetric and deviatoric plastic strain rates in this case are given as :

$$\begin{aligned} \dot{\epsilon}_{vp} &= \dot{\lambda}(-2\alpha p)/2g' = \frac{\dot{\lambda}(-2\alpha p)}{\sqrt{4\sigma_{vm}^2 + 4\alpha p^2}}, \\ \dot{\epsilon}_{dp} &= \dot{\lambda}3s/2g' = \frac{\dot{\lambda}3s}{\sqrt{4\sigma_{vm}^2 + 4\alpha p^2}}, \end{aligned} \quad (23.187.24)$$

which amounts to a different definition of the plastic consistency parameter which of course has to be considered when equivalent plastic strain values are computed.

23.187.2 Hardening formulation

The hardening formulation is the attractive part of SAMP-1. The formulation is fully tabulated and consequently the user can directly input measurement results from uniaxial tension, uniaxial compression and simple shear tests in terms of load curves giving the yield stress as a function of the corresponding plastic strain. No fitting of coefficients is required. The test results that are reflected in the load curves will be used exactly by SAMP-1 without fitting to any analytical expression. Consequently the hardening will be dependent upon the state of stress and not only upon the plastic strain.

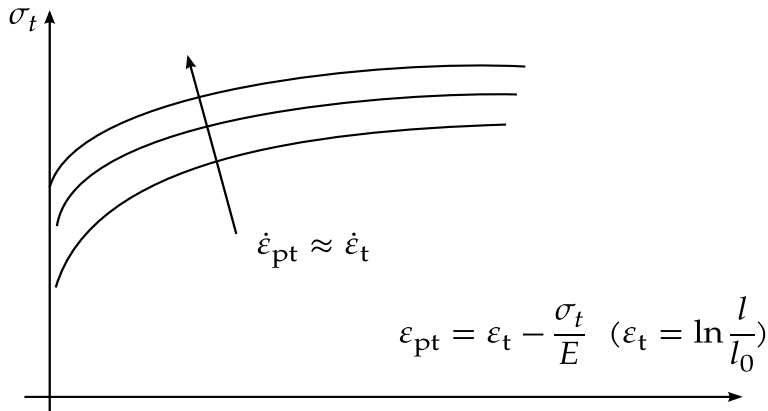


Figure 23.3. Tensile hardening curve from dynamic tensile tests

23.187.3 Rate effects

Plastics are usually highly rate dependent. A proper viscoplastic consideration of the rate effects is therefore important in the numerical treatment of the material law. Data to determine the rate dependency are based on uniaxial dynamic testing. If dynamic tests are available, then the load curve defining the yield stress in uniaxial tension is simply replaced by a table definition containing multiple load curves corresponding to different values of the plastic strain rate. This is illustrated in the [Figure 23.3](#).

23.187.4 Damage and failure

Numerous damage models can be found in the literature. Probably the simplest concept is elastic damage where the damage parameter (usually written as d) is a function of the elastic energy and effectively reduces the elastic modulus of the material. In the case of ductile damage, d is a function of plastic straining and affects the yield stress rather than the elastic modulus. This is equivalent to plastic softening. In more sophisticated damage models, d depends on both the plastic straining and the elastic energy (and maybe other factors) and effects yield stress as well as elastic modulus.

A simple damage model was added to the SAMP-1 material law where the damage parameter d is a function of plastic strain only. A load curve must be provided by the user giving d as a function of the (true) plastic strain under uniaxial tension. The value of the critical damage D_c leading to rupture is then the only other required additional input. The implemented damage model is isotropic.

The implemented model then uses the notion of effective cross section, which is the true cross section of the material minus the cracks that have developed. We will use the following notation:

- $A_0 \rightarrow$ undeformed cross section
- $A \rightarrow$ deformed or current cross section
- $A_0 \rightarrow$ undeformed cross section

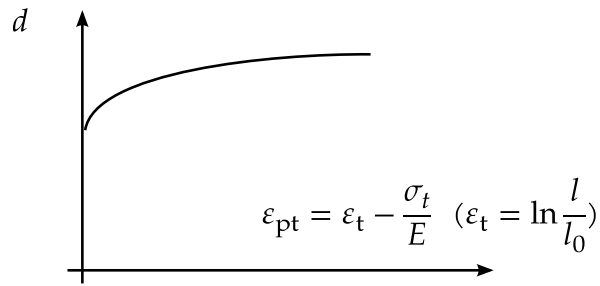


Figure 23.4. Damage parameter from uniaxial tensile test

We define the effective stress as the force divided by the effective cross section:

$$\begin{aligned}\sigma &= \frac{f}{A'}, \\ \sigma_{\text{eff}} &= \frac{f}{A_{\text{eff}}} = \frac{f}{A(1-d)} = \frac{\sigma}{1-d'}\end{aligned}\tag{23.25}$$

which allows defining an effective yield stress:

$$\sigma_{y,\text{eff}} = \frac{\sigma_y}{1-d'}.\tag{23.26}$$

23.196 Material Model 196: General Spring Discrete Beam

If TYPE = 0, elastic behavior is obtained. In this case, if the linear spring stiffness is used, the force, F , is given by:

$$F = F_0 + K\Delta L + D\Delta\dot{L}, \quad (23.196.1)$$

but if the load curve ID is specified, the force is then given by:

$$F = F_0 + K f(\Delta L) \left[1 + C1 \cdot \Delta\dot{L} + C2 \cdot \text{sgn}(\Delta\dot{L}) \ln \left(\max \left\{ 1, \frac{|\Delta\dot{L}|}{DLE} \right\} \right) \right] + D\Delta\dot{L} + g(\Delta L)h(\Delta\dot{L}). \quad (23.196.2)$$

In these equations, ΔL is the change in length

$$\Delta L = \text{current length} - \text{initial length}. \quad (23.196.3)$$

If TYPE = 1, inelastic behavior is obtained. In this case, the yield force is taken from the load curve:

$$F^Y = F_y(\Delta L^{\text{plastic}}), \quad (23.196.4)$$

where L^{plastic} is the plastic deflection. A trial force is computed as:

$$F^T = F^n + K\Delta\dot{L}\Delta t, \quad (23.196.5)$$

and is checked against the yield force to determine F :

$$F = \begin{cases} F^Y & \text{if } F^T > F^Y \\ F^T & \text{if } F^T \leq F^Y \end{cases}. \quad (23.196.6)$$

The final force, which includes rate effects and damping, is given by:

$$F^{n+1} = F \cdot \left[1 + C1 \cdot \Delta\dot{L} + C2 \cdot \text{sgn}(\Delta\dot{L}) \ln \left(\max \left\{ 1, \frac{|\Delta\dot{L}|}{DLE} \right\} \right) \right] + D\Delta\dot{L} + g(\Delta L)h(\Delta\dot{L}). \quad (23.196.7)$$

Unless the origin of the curve starts at (0,0), the negative part of the curve is used when the spring force is negative where the negative of the plastic displacement is used to interpolate, F_y . The positive part of the curve is used whenever the force is positive.

The cross sectional area is defined on the section card for the discrete beam elements, See *SECTION_BEAM. The square root of this area is used as the contact thickness offset if these elements are included in the contact treatment.

24

Equation of State Models

LS-DYNA has 10 equation of state models which are described in this section.

1. Linear Polynomial
2. JWL High Explosive
3. Sack "Tuesday" High Explosive
4. Gruneisen
5. Ratio of Polynomials
6. Linear Polynomial With Energy Deposition
7. Ignition and Growth of Reaction in High Explosives
8. Tabulated Compaction
9. Tabulated
10. Propellant-Deflagration

The forms of the first five equations of state are given in the KOVEC user's manual [Woodruff 1973] as well as below.

24.1 Equation of State Form 1: Linear Polynomial

This polynomial equation of state, linear in the internal energy per initial volume, E , is given by

$$p = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2)E \quad (24.1.1)$$

Here $C_0, C_1, C_2, C_3, C_4, C_5$ and C_6 are user defined constants and

$$\mu = \frac{1}{V} - 1. \quad (24.1.2)$$

where V is the relative volume. In expanded elements, the coefficients of μ^2 are set to zero, i.e.,

$$C_2 = C_6 = 0. \quad (24.1.3)$$

The linear polynomial equation of state may be used to model gas with the gamma law equation of state. This may be achieved by setting:

$$C_0 = C_1 = C_2 = C_3 = C_6 = 0, \quad (24.1.4)$$

and

$$C_4 = C_5 = \gamma - 1, \quad (24.1.5)$$

where γ is the ratio of specific heats. The pressure is then given by:

$$p = (\gamma - 1) \frac{\rho}{\rho_0} E. \quad (24.1.6)$$

Note that the units of E are the units of pressure.

24.2 Equation of State Form 2: JWL High Explosive

The JWL equation of state defines pressure as a function of relative volume, V , and internal energy per initial volume, E , as

$$p = A \left(1 - \frac{\omega}{R_1 V} \right) e^{-R_1 V} + B \left(1 - \frac{\omega}{R_2 V} \right) e^{-R_2 V} + \frac{\omega E}{V}, \quad (24.2.7)$$

where ω , A , B , R_1 and R_2 are user defined input parameters. The JWL equation of state is used for determining the pressure of the detonation products of high explosives in applications involving metal accelerations. Input parameters for this equation are given by Dobratz [1981] for a variety of high explosive materials.

This equation of state is used with the explosive burn (material model 8) material model which determines the lighting time for the explosive element.

24.3 Equation of State Form 3: Sack “Tuesday” High Explosives

Pressure of detonation products is given in terms of the relative volume, V , and internal energy per initial volume, E , as [Woodruff 1973]:

$$p = \frac{A_3}{V^{A_1}} e^{-A_2 V} \left(1 - \frac{B_1}{V} \right) + \frac{B_2}{V} E, \quad (24.3.8)$$

where A_1 , A_2 , A_3 , B_1 and B_2 are user-defined input parameters.

This equation of state is used with the explosive burn (material model 8) material model which determines the lighting time for the explosive element.

24.4 Equation of State Form 4: Gruneisen

The Gruneisen equation of state with cubic shock velocity-particle velocity defines pressure for compressed material as

$$p = \frac{\rho_0 C^2 \mu \left[1 + \left(1 - \frac{\gamma_0}{2} \right) \mu - \frac{a}{2} \mu^2 \right]}{\left[1 - (S_1 - 1) \mu - S_2 \frac{\mu^2}{\mu + 1} - S_3 \frac{\mu^3}{(\mu + 1)^2} \right]^2} + (\gamma_0 + \alpha \mu) E, \quad (24.4.9)$$

where E is the internal energy per initial volume, C is the intercept of the $u_s - u_p$ curve, S_1 , S_2 , and S_3 are the coefficients of the slope of the $u_s - u_p$ curve, γ_0 is the Gruneisen gamma, and a is the first order volume correction to γ_0 . Constants C , S_1 , S_2 , S_3 , γ_0 and a are user defined input parameters. The compression is defined in terms of the relative volume, V , as:

$$\mu = \frac{1}{V} - 1. \quad (24.4.10)$$

For expanded materials as the pressure is defined by:

$$p = \rho_0 C^2 \mu + (\gamma_0 + \alpha \mu) E. \quad (24.4.11)$$

24.5 Equation of State Form 5: Ratio of Polynomials

The ratio of polynomials equation of state defines the pressure as

$$p = \frac{F_1 + F_2 E + F_3 E^2 + F_4 E^3}{F_5 + F_6 E + F_7 E^2} (1 + \alpha \mu), \quad (24.5.12)$$

where

$$F_i = \sum_{j=0}^n A_{ij} m^j, \quad n = 4 \text{ if } i < 3, \quad n = 3 \text{ if } i \geq 3 \quad (24.5.13)$$

$$\mu = \frac{\rho}{\rho_0 - 1}. \quad (24.5.14)$$

In expanded zoned F_1 is replaced by $F'_1 = F_1 + \beta\mu^2$ Constants A_{ij} , α , and β are user input.

24.6 Equation of State Form 6: Linear With Energy Deposition

This polynomial equation of state, linear in the internal energy per initial volume, E , is given by

$$p = C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3 + (C_4 + C_5\mu + C_6\mu^2)E, \quad (24.6.15)$$

Here $C_0, C_1, C_2, C_3, C_4, C_5$ and C_6 are user defined constants and

$$\mu = \frac{1}{V} - 1, \quad (24.6.16)$$

where V is the relative volume. In expanded elements, we set the coefficients of μ^2 to zero, i.e.,

$$C_2 = C_6 = 0. \quad (24.6.17)$$

Internal energy, E , is increased according to an energy deposition rate versus time curve whose ID is defined in the input.

24.7 Equation of State Form 7: Ignition and Growth Model

A JWL equation of state defines the pressure in the unreacted high explosive as

$$P_e = A_e \left(1 - \frac{\omega_e}{R1_e V_e}\right) e^{-R1_e V_e} + B_e \left(1 - \frac{\omega_e}{R2_e V_e}\right) e^{-R2_e V_e} + \frac{\omega_e E}{V_e}, \quad (24.7.18)$$

where V_e is the relative volume, E_e is the internal energy, and the constants $A_e, B_e, \omega_e, R1_e$ and $R2_e$ are input constants. Similarly, the pressure in the reaction products is defined by another JWL form

$$P_p = A_p \left(1 - \frac{\omega_p}{R1_p V_p}\right) e^{-R1_p V_p} + B_p \left(1 - \frac{\omega_p}{R2_p V_p}\right) e^{-R2_p V_p} + \frac{\omega_p E}{V_p}. \quad (24.7.19)$$

The mixture of unreacted explosive and reaction products is defined by the fraction reacted F ($F = 0$ implies no reaction, $F = 1$ implies complete conversion from explosive to products). The pressures and temperature are assumed to be in equilibrium, and the relative volumes are assumed to be additive:

$$V = (1 - F)V_e + FV_p. \quad (24.7.20)$$

The rate of reaction is defined as

$$\frac{\partial F}{\partial t} = I(\text{FCRIT} - F)^y (V_e^{-1} - 1)^3 [1 + G(V_e^{-1} - 1)] + H(1 - F)^y F^x P^z (V_p^{-1} - 1)^m, \quad (24.7.21)$$

where I, G, H, x, y, z and m (generally $m = 0$) are input constants.

The JWL equations of state and the reaction rates have been fitted to one- and two-dimensional shock initiation and detonation data for four explosives: PBX-9404, RX-03-BB, PETN, and cast TNT. The details of calculational method are described by Cochran and Chan [1979]. The detailed one-dimensional calculations and parameters for the four explosives are given by Lee and Tarver [1980]. Two-dimensional calculations with this model for PBX 9404 and LX-17 are discussed by Tarver and Hallquist [1981].

24.8 Equation of State Form 8: Tabulated Compaction

Pressure is positive in compression, and volumetric strain ε_V is positive in tension. The tabulated compaction model is linear in internal energy per unit volume. Pressure is defined by

$$p = C(\varepsilon_V) + \gamma T(\varepsilon_V)E, \quad (24.8.22)$$

during loading (compression). Unloading occurs at a slope corresponding to the bulk modulus at the peak (most compressive) volumetric strain, as shown in [Figure 24.1](#). Re-loading follows the unloading path to the point where unloading began, and then continues on the loading path described by [Equation \(24.8.22\)](#).

24.9 Equation of State Form 9: Tabulated

The tabulated equation of state model is linear in internal energy. Pressure is defined by

$$p = C(\varepsilon_V) + \gamma T(\varepsilon_V)E, \quad (24.9.23)$$

The volumetric strain ε_V is given by the natural algorithm of the relative volume. Up to 10 points and as few as 2 may be used when defining the tabulated functions. The pressure is extrapolated if necessary. Loading and unloading are along the same curve unlike equation of state form 8.

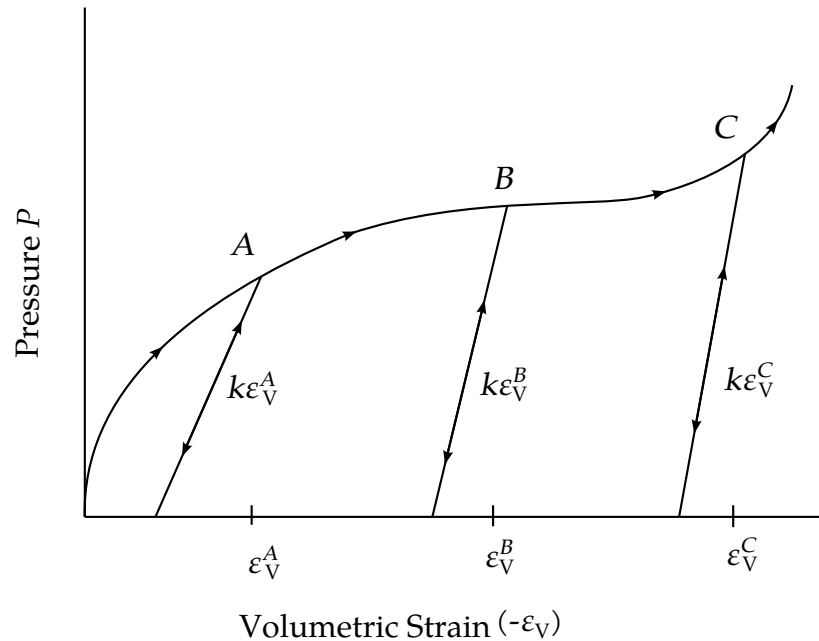


Figure 24.1. Pressure versus volumetric strain curve for equation of state form 8 with compaction. In the compacted states, the bulk unloading modulus depends on the peak volumetric strain.

24.10 Equation of State Form 10: Propellant-Deflagration

A deflagration (burn rate) reactive flow model requires an unreacted solid equation-of-state, a reaction product equation-of-state, a reaction rate law and a mixture rule for the two (or more) species. The mixture rule for the standard ignition and growth model [Lee and Tarver 1980] assumes that both pressures and temperatures are completely equilibrated as the reaction proceeds. However, the mixture rule can be modified to allow no thermal conduction or partial heating of the solid by the reaction product gases. For this relatively slow process of airbag propellant burn, the thermal and pressure equilibrium assumptions are valid. The equations-of-state currently used in the burn model are the JWL, Gruneisen, the van der Waals co-volume, and the perfect gas law, but other equations-of-state can be easily implemented. In this propellant burn, the gaseous nitrogen produced by the burning sodium azide obeys the perfect gas law as it fills the airbag but may have to be modeled as a van der Waal's gas at the high pressures and temperatures produced in the propellant chamber. The chemical reaction rate law is pressure, particle geometry and surface area dependant, as are most high-pressure burn processes. When the temperature profile of the reacting system is well known, temperature dependent Arrhenius chemical kinetics can be used.

Since the airbag propellant composition and performance data are company private information, it is very difficult to obtain the required information for burn rate modeling. However, Imperial Chemical Industries (ICI) Corporation supplied pressure

exponent, particle geometry, packing density, heat of reaction, and atmospheric pressure burn rate data which allowed us to develop the numerical model presented here for their $\text{NaN}_3 + \text{Fe}_2\text{O}_3$ driver airbag propellant. The deflagration model, its implementation, and the results for the ICI propellant are presented in the are described by [Hallquist, et. al., 1990].

The unreacted propellant and the reaction product equations-of-state are both of the form:

$$p = A e^{-R_1 V} + B e^{-R_2 V} + \frac{\omega C_v T}{V - d} \quad (24.10.24)$$

where p is pressure (in Mbars), V is the relative specific volume (inverse of relative density), ω is the Gruneisen coefficient, C_v is heat capacity (in Mbars -cc/cc°K), T is temperature in °K, d is the co-volume, and A, B, R_1 and R_2 are constants. Setting $A = B = 0$ yields the van der Waal's co-volume equation-of-state. The JWL equation-of-state is generally useful at pressures above several kilobars, while the van der Waal's is useful at pressures below that range and above the range for which the perfect gas law holds. Of course, setting $A = B = d = 0$ yields the perfect gas law. If accurate values of ω and C_v plus the correct distribution between "cold" compression and internal energies are used, the calculated temperatures are very reasonable and thus can be used to check propellant performance.

The reaction rate used for the propellant deflagration process is of the form:

$$\begin{aligned} \frac{\partial F}{\partial t} &= Z(1 - F)^y F^x p^w + V(1 - F)^u Frp^s \\ \text{for } 0 < F < F_{\text{limit1}} \\ \text{for } F_{\text{limit2}} < F < 1 \end{aligned} \quad (24.10.25)$$

where F is the fraction reacted ($F = 0$ implies no reaction, $F = 1$ is complete reaction), t is time, and p is pressure (in Mbars), $r, s, u, w, x, y, F_{\text{limit1}}$ and F_{limit2} are constants used to describe the pressure dependence and surface area dependence of the reaction rates. Two (or more) pressure dependent reaction rates are included in case the propellant is a mixture or exhibited a sharp change in reaction rate at some pressure or temperature. Burning surface area dependences can be approximated using the $(1 - F)^y F^x$ terms. Other forms of the reaction rate law, such as Arrhenius temperature dependent $e^{-E/RT}$ type rates, can be used, but these require very accurate temperatures calculations. Although the theoretical justification of pressure dependent burn rates at kilobar type pressures is not complete, a vast amount of experimental burn rate versus pressure data does demonstrate this effect and hydrodynamic calculations using pressure dependent burn accurately simulate such experiments.

The deflagration reactive flow model is activated by any pressure or particle velocity increase on one or more zone boundaries in the reactive material. Such an increase creates pressure in those zones and the decomposition begins. If the pressure is relieved, the reaction rate decreases and can go to zero. This feature is important for short

duration, partial decomposition reactions. If the pressure is maintained, the fraction reacted eventually reaches one and the material is completely converted to product molecules. The deflagration front rates of advance through the propellant calculated by this model for several propellants are quite close to the experimentally observed burn rate versus pressure curves.

To obtain good agreement with experimental deflagration data, the model requires an accurate description of the unreacted propellant equation-of-state, either an analytical fit to experimental compression data or an estimated fit based on previous experience with similar materials. This is also true for the reaction products equation-of-state. The more experimental burn rate, pressure production and energy delivery data available, the better the form and constants in the reaction rate equation can be determined.

Therefore the equations used in the burn subroutine for the pressure in the unreacted propellant

$$P_u = R1 \cdot e^{-R5 \cdot V_u} + R2 \cdot e^{-R6 \cdot V_u} + \frac{R3 \cdot T_u}{V_u - \text{FRER}'} \quad (24.10.26)$$

where V_u and T_u are the relative volume and temperature respectively of the unreacted propellant. The relative density is obviously the inverse of the relative volume. The pressure P_p in the reaction products is given by:

$$P_p = A \cdot e^{-XP1 \cdot V_p} + B \cdot e^{-XP2 \cdot V_p} + \frac{G \cdot T_p}{V_p - \text{CCRIT}} \quad (24.10.27)$$

As the reaction proceeds, the unreacted and product pressures and temperatures are assumed to be equilibrated ($T_u = T_p = T$, $p = P_u = P_p$) and the relative volumes are additive:

$$V = (1 - F) \cdot V_u + F \cdot V_p \quad (24.10.28)$$

where V is the total relative volume. Other mixture assumptions can and have been used in different versions of DYNA2D/3D. The reaction rate law has the form:

$$\begin{aligned} \frac{\partial F}{\partial t} = & \text{GROW1}(p + \text{FREQ})^{em} (F + \text{FMXIG})^{ar1} (1 - F + \text{FMXIG})^{es1} \\ & + \text{GROW2}(p + \text{FREQ})^{en} (F + \text{FMXIG}) \end{aligned} \quad (24.10.29)$$

If F exceeds FMXGR , the GROW1 term is set equal to zero, and, if F is less than FMNGR , the GROW2 term is zero. Thus, two separate (or overlapping) burn rates can be used to describe the rate at which the propellant decomposes.

This equation-of-state subroutine is used together with a material model to describe the propellant. In the airbag propellant case, a null material model (type #10) can be used. Material type #10 is usually used for a solid propellant or explosive when the shear modulus and yield strength are defined. The propellant material is defined by the

material model and the unreacted equation-of-state until the reaction begins. The calculated mixture states are used until the reaction is complete and then the reaction product equation-of-state is used. The heat of reaction, ENQ , is assumed to be a constant and the same at all values of F but more complex energy release laws could be implemented.

25

Artificial Bulk Viscosity

Bulk viscosity is used to treat shock waves. Proposed in one spatial dimension by von Neumann and Richtmyer [1950], the bulk viscosity method is now used in nearly all wave propagation codes. A viscous term q is added to the pressure to smear the shock discontinuities into rapidly varying but continuous transition regions. With this method the solution is unperturbed away from a shock, the Hugoniot jump conditions remain valid across the shock transition, and shocks are treated automatically. In our discussion of bulk viscosity we draw heavily on works by Richtmyer and Morton [1967], Noh [1976], and Wilkins [1980]. The following discussion of the bulk viscosity applies to solid elements since strong shocks are not normally encountered in structures modeled with shell and beam elements.

25.1 Shock Waves

Shock waves result from the property that sound speed increases with increasing pressure. A smooth pressure wave can gradually steepen until it propagates as a discontinuous disturbance called a shock. See [Figure 25.1](#). Shocks lead to jumps in pressure, density, particle velocity, and energy.

Consider a planar shock front moving through a material. The velocity ahead of the shock is u_1 ; the velocity behind is u_2 ; and the shock velocity is u_s . Mass, momentum, and energy are conserved across the front. Application of these conservation laws leads to the well-known Rankine-Hugoniot jump conditions

$$(\rho_2 - \rho_1)u_s = \rho_2 u_2 - \rho_1 u_1 \quad (25.1)$$

$$\rho_2(u_s - u_2)u_2 + \rho_1(u_1 - u_s)u_1 = p_2 - p_1 \quad (25.2)$$

$$(E_2 - E_1)u_s = (E_2 + p_2)u_2 - (E_1 + p_1)u_1 \quad (25.3)$$

where [Equation \(25.3\)](#) is an expression of the energy jump condition using the results of mass conservation, [Equation \(25.1\)](#), and momentum conservation, [Equation \(25.2\)](#). Here,

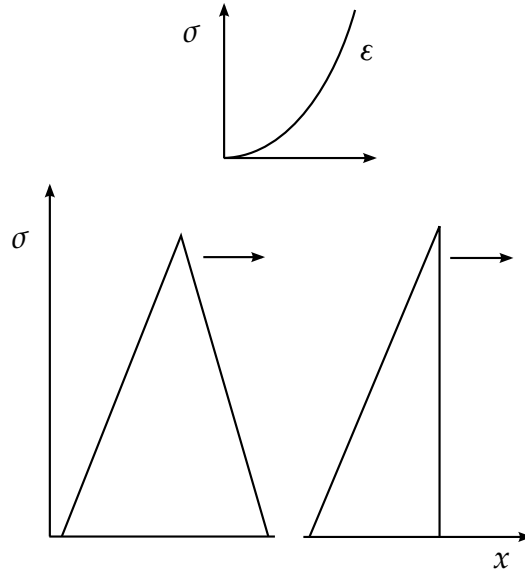


Figure 25.1. If the sound speed increases as the stress increases the traveling wave above will gradually steepen as it moves along the x-coordinate to form a shock wave.

ρ_i is the density, E_i is the energy density, and p_i is the pressure; the subscript, i , is 1 for ahead of the shock and 2 for behind the shock.

The energy equation relating the thermodynamic quantities density, pressure, and energy must be satisfied for all shocks. The equation of state

$$p = p(\rho, e), \quad (25.4)$$

which defines all equilibrium states that can exist in a material and relating the same quantities as the energy equation, must also be satisfied. We may use this equation to eliminate energy from Equation (25.3) and obtain a unique relationship between pressure and compression. This relation, called the Hugoniot, determines all pressure-compression states achievable behind the shock. Shocking takes place along the Rayleigh line and not the Hugoniot (Figure 25.1) and because the Hugoniot curve closely approximates an isentrope, we may often assume the unloading follows the Hugoniot. Combining Equations (25.1) and (25.2) and choosing a coordinate frame such that $u_1 = 0$, we see that the slope of the Rayleigh line is related to the shock speed:

$$u_s = \frac{1}{\rho_1} \sqrt{\frac{p_2 - p_1}{\frac{1}{\rho_1} - \frac{1}{\rho_2}}}. \quad (25.5)$$

For the material of Figure 25.2, increasing pressure increases shock speed.

Consider a γ -law gas with the equation of state

$$p = (\gamma - 1)\rho e, \quad (25.6)$$

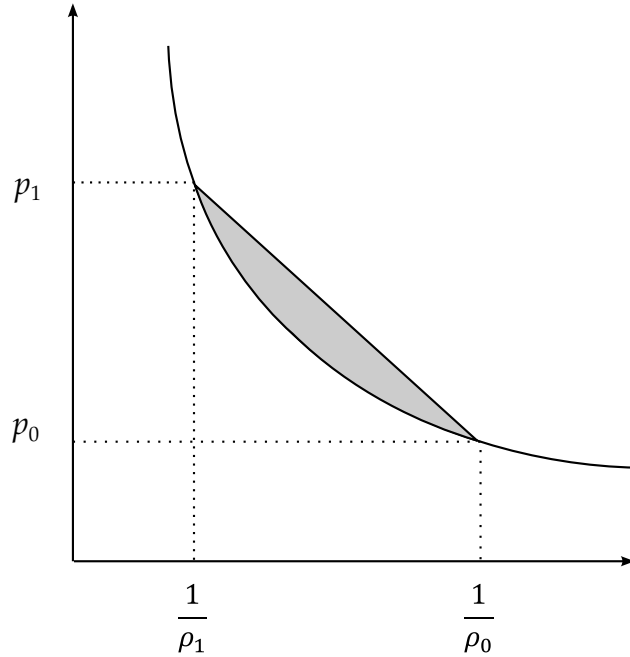


Figure 25.2. Shocking takes place along the Rayleigh line, and release closely follows the Hugoniot. The cross-hatched area is the difference between the internal energy behind the shock and the internal energy lost on release.

where γ is the ratio of specific heats. Using the energy jump condition, we can eliminate e and obtain the Hugoniot

$$\frac{p}{p_0} = p^* = \frac{2V_0 + (\gamma - 1)(V_0 - V)}{2V - (\gamma - 1)(V_0 - V)}, \quad (25.7)$$

where V is the relative volume. [Figure 25.3](#) shows a plot of the Hugoniot and adiabat where it is noted that for $p^* = 1$, the slopes are equal. Thus for weak shocks, Hugoniot and adiabat agree to the first order and can be ignored in numerical calculations. However, special treatment is required for strong shocks, and in numerical calculations this special treatment takes the form of bulk viscosity.

25.2 Bulk Viscosity

In the presence of shocks, the governing partial differential equations can give multiple weak solutions. In their discussion of the Rankine-Hugoniot jump conditions, Richtmyer and Morton [1967] report that the unmodified finite difference (element) equations often will not produce even approximately correct answers. One possible solution is to employ shock fitting techniques and treat the shocks as interior boundary conditions. This technique has been used in one spatial dimension but is too complex to extend to multi-dimensional problems of arbitrary geometry. For these reasons the pseudo-viscosity method was a major breakthrough for computational mechanics.

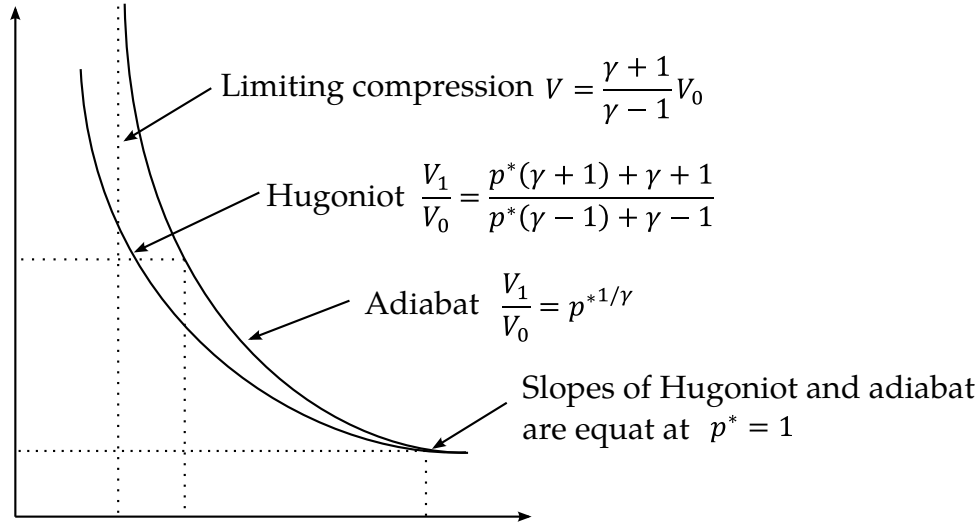


Figure 25.3. Hugoniot curve and adiabat for a g-law gas (from [Noh 1976]).

The viscosity proposed by von Neumann and Richtmyer [1950] in one spatial dimension has the form

$$q = C_0 \rho (\Delta x)^2 \left(\frac{\partial \dot{x}}{\partial x} \right)^2 \quad \text{if } \frac{\partial \dot{x}}{\partial x} < 0$$

$$q = 0 \quad \text{if } \frac{\partial \dot{x}}{\partial x} \geq 0$$
(25.8)

where C_0 is a dimensionless constant and q is added to the pressure in both the momentum and energy equations. When q is used, they proved the following for steady state shocks: the hydrodynamic equations possess solutions without discontinuities; the shock thickness is independent of shock strength and of the same order as the Δx used in the calculations; the q term is insignificant outside the shock layer; and the jump conditions are satisfied. According to Noh, it is generally believed that these properties: “hold for all shocks, and this has been borne out over the years by countless numerical experiments in which excellent agreement has been obtained either with exact solutions or with hydrodynamical experiments.”

In 1955, Landshoff [1955] suggested the addition of a linear term to the q of von Neumann and Richtmyer leading to a q of the form

$$q = C_0 \rho (\Delta x)^2 \left(\frac{\partial \dot{x}}{\partial x} \right)^2 - C_1 \rho a \Delta x \frac{\partial \dot{x}}{\partial x} \quad \text{if } \frac{\partial \dot{x}}{\partial x} < 0$$

$$q = C_0 \rho (\Delta x)^2 \left(\frac{\partial \dot{x}}{\partial x} \right)^2 + C_1 \rho a \Delta x \frac{\partial \dot{x}}{\partial x} \quad \text{if } \frac{\partial \dot{x}}{\partial x} \geq 0,$$
(25.9)

where C_1 is a dimensionless constant and a is the local sound speed. The linear term rapidly damps numerical oscillations behind the shock front (Figure 25.3). A similar form was proposed independently by Noh about the same time.

In an interesting aside, Wilkins [1980] discusses work by Kuropatenko who, given an equation of state, derived a q by solving the jump conditions for pressure in terms of a change in the particle velocity, Δu . For an equation of state of the form,

$$p = K \left(-\frac{\rho}{\rho_0} - 1 \right), \quad (25.10)$$

pressure across the shock front is given by [Wilkins 1980]

$$p = p_0 + \rho_0 \frac{(\Delta u)^2}{2} + \rho_0 |\Delta u| \left[\frac{(\Delta u)^2}{2} + a^2 \right]^{1/2}, \quad (25.11)$$

where a is a sound speed

$$a = \left(\frac{K}{\rho_0} \right)^{1/2}. \quad (25.12)$$

For a strong shock, $\Delta u^2 \gg a^2$, we obtain the quadratic form

$$q = \rho_0 \Delta u^2, \quad (25.13)$$

and for a weak shock, $\Delta u^2 \ll a^2$, the linear form

$$q = \rho_0 a \Delta u, \quad (25.14)$$

Thus linear and quadratic forms for q can be naturally derived. According to Wilkins, the particular expressions for q obtained by Kuropatenko offer no particular advantage over the expressions currently used in most computer programs.

In extending the one-dimensional viscosity formulations to multi-dimensions, most code developers have simply replaced the divergence of the velocity with, $\dot{\epsilon}_{kk}$, the trace of the strain rate tensor, and the characteristic length with the square root of the area A , in two dimensions and the cubic root of the volume v in three dimensions. These changes also give the default viscosities in the LS-DYNA codes:

$$q = \rho \begin{cases} l(C_0 l \dot{\epsilon}_{kk}^2 - C_1 a \dot{\epsilon}_{kk}) & \text{if } \dot{\epsilon}_{kk} < 0 \\ 0 & \text{if } \dot{\epsilon}_{kk} \geq 0 \end{cases} \quad (25.15)$$

where C_0 and C_1 are dimensionless constants which default to 1.5 and 0.06, respectively, where $1 = \sqrt{A}$ in 2D, and $\sqrt[3]{v}$ in 3D, a is the local sound speed, C_0 defaults to 1.5 and C_1 defaults to 0.06.

In converging two- and three-dimensional geometries, the strain rate $\dot{\epsilon}_{kk}$ is negative and the q term in Equation (25.15) is nonzero, even though no shocks may be generated. This results in nonphysical q heating. When the aspect ratios of the elements are poor (far from unity), the use of a characteristic length based on \sqrt{A} or $\sqrt[3]{v}$ can also result in nonphysical q heating and even occasional numerical instabilities. Wilkins uses a bulk viscosity that is based in part on earlier work by Richards [1965] that extends the von

Neumann and Richtmyer formulations in a way that avoids these problems. This latter q may be added in the future if the need arises.

Wilkins' q is defined as:

$$q = C_0 \rho l^2 \left(\frac{ds}{dt} \right)^2 - C_l \rho l a^* \frac{ds}{dt} \quad \text{if } \dot{\epsilon}_{kk} < 0$$

$$q = 0 \quad \text{if } \frac{ds}{dt} \geq 0$$
(25.16)

where l and $\frac{ds}{dt}$ are the thickness of the element and the strain rate in the direction of the acceleration, respectively, and a^* is the sound speed defined by $(p/\rho)^{1/2}$ if $p > 0$. We use the local sound speed in place of a^* to reduce the noise at the low stress levels that are typical of our applications.

Two disadvantages are associated with [Equation \(25.16\)](#). To compute the length parameter and the strain rate, we need to know the direction of the acceleration through the element. Since the nodal force vector becomes the acceleration vector in the explicit integration scheme, we have to provide extra storage to save the direction. In three dimensions our present storage capacity is marginal at best and sacrificing this storage for storing the direction would make it even more so. Secondly, we need to compute l and $\frac{ds}{dt}$ which results in a noticeable increase in computer cost even in two dimensions. For most problems the additional refinement of Wilkins is not needed. However, users must be aware of the pitfalls of [Equation \(25.15\)](#), i.e., when the element aspect ratios are poor or the deformations are large, an anomalous q may be generated.

26

Time Step Control

During the solution we loop through the elements to update the stresses and the right hand side force vector. We also determine a new time step size by taking the minimum value over all elements.

$$\Delta t^{n+1} = a \cdot \min\{\Delta t_1, \Delta t_2, \Delta t_3, \dots, \Delta t_N\}, \quad (26.1)$$

where N is the number of elements. For stability reasons the scale factor a is typically set to a value of .90 (default) or some smaller value.

26.1 Time Step Calculations for Solid Elements

A critical time step size, Δt_e , is computed for solid elements from

$$\Delta t_e = \frac{L_e}{\{[Q + (Q^2 + c^2)^{1/2}]\}} \quad (26.2)$$

where Q is a function of the bulk viscosity linear and quadratic coefficients Q_2 and Q_1 (see *CONTROL_BULK_VISCOSITY):

$$Q = \begin{cases} Q_2 c + Q_1 L_e |\dot{\epsilon}_{kk}| & \text{for } \dot{\epsilon}_{kk} \leq 0 \\ 0 & \text{for } \dot{\epsilon}_{kk} > 0 \end{cases} \quad (26.3)$$

L_e is a characteristic length:

$$\text{8 node solids: } L_e = \frac{v_e}{A_{e\max}}$$

$$\text{4 node tetrahedras: } L_e = \text{minimum altitude}$$

v_e is the element volume, $A_{e\max}$ is the area of the largest side, and c is the adiabatic sound speed:

$$c = \left[\frac{4G}{3\rho_0} + \frac{\partial p}{\partial \rho} \right]_s^{1/2}, \quad (26.4)$$

where ρ is the specific mass density. Noting that:

$$\left(\frac{\partial p}{\partial \rho}\right)_s = \left(\frac{\partial p}{\partial \rho}\right)_U + \left(\frac{\partial p}{\partial U}\right)_\rho \left(\frac{\partial U}{\partial \rho}\right)_s, \quad (26.5)$$

and that along an isentrope the incremental energy, U , in the units of pressure is the product of pressure, p , and the incremental relative volume, dV :

$$dU = -pdV, \quad (26.6)$$

we obtain

$$c = \left[\frac{4G}{3\rho_0} + \left(\frac{\partial p}{\partial \rho}\right)_U + \frac{pV^2}{\rho_0} \left(\frac{\partial p}{\partial U}\right)_\rho \right]^{1/2}. \quad (26.7)$$

For elastic materials with a constant bulk modulus the sound speed is given by:

$$c = \sqrt{\frac{E(1-\nu)}{(1+\nu)(1-2\nu)\rho}} \quad (26.8)$$

where E is Young's modulus, and ν is Poisson's ratio.

In the presence of dynamic viscosity, the time step in (26.1) may be modified as

$$\Delta t_e = \frac{L_e}{\{[Q + \hat{\mu} + ((Q + \hat{\mu})^2 + c^2)^{1/2}]\}}, \quad (26.9)$$

where the bulk viscosity Q is the same as before, and $\hat{\mu}$ is a function of the dynamic viscosity μ (see *MAT_NULL)

$$\hat{\mu} = \frac{4}{3} \frac{\mu}{\rho L_e}. \quad (26.10)$$

See *CONTROL_SOLID for activation of this time-step modification.

26.2 Time Step Calculations for Beam and Truss Elements

For the Hughes-Liu beam and truss elements, the time step size is given by:

$$\Delta t_e = \frac{L_e}{c} \quad (26.11)$$

where L_e is the length of the element and c is the wave speed:

$$c = \sqrt{\frac{E}{\rho}}. \quad (26.12)$$

For the Belytschko beam the time step size given by the longitudinal sound speed is used (Equation (26.11)), unless the bending-related time step size given by [Belytschko and Tsay 1982]

$$\Delta t_e = \frac{0.5L_e}{c \sqrt{3I \left[\frac{3}{12I + AL_e^2} + \frac{1}{AL_e^2} \right]}} \quad (26.13)$$

is smaller, where I and A are the maximum value of the moment of inertia and area of the cross section, respectively.

Comparison of critical time steps of the truss versus the elastic solid element shows that if Poisson's ratio, ν , is nonzero the solid elements give a considerably smaller stable time step size. If we define the ratio, α , as:

$$\alpha = \frac{\Delta t_{\text{continuum}}}{\Delta t_{\text{rod}}} = \frac{C_{\text{rod}}}{C_{\text{continuum}}} = \sqrt{\frac{(1+\nu)(1-2\nu)}{1-\nu}}, \quad (26.14)$$

we obtain the results in Table 22.1 where we can see that as ν approaches .5 $\alpha \rightarrow 0$.

	0	0.2	0.3	0.4	0.45	0.49	0.50
	1.	0.949	0.862	0.683	0.513	0.242	0.0

Table 22.1. Comparison of critical time step sizes for a truss versus a solid element.

26.3 Time Step Calculations for Shell Elements

For the shell elements, the time step size is given by:

$$\Delta t_e = \frac{L_s}{c} \quad (26.15)$$

where L_s is the characteristic length and c is the sound speed:

$$c = \sqrt{\frac{E}{\rho(1-\nu^2)}}. \quad (26.16)$$

Three user options exists for choosing the characteristic length. In the default or first option the characteristic length is given by:

$$L_s = \frac{(1+\beta)A_s}{\max(L_1, L_2, L_3, (1-\beta)L_4)} \quad (26.17)$$

where $\beta = 0$ for quadrilateral and 1 for triangular shell elements, A_s is the area, and L_i , ($i = 1 \dots 4$) is the length of the sides defining the shell elements. In the second option a more conservative value of L_s is used:

$$L_s = \frac{(1 + \beta)A_s}{\max(D_1, D_2)}, \quad (26.18)$$

where $D_i (i = 1, 2)$ is the length of the diagonals. The third option provides the largest time step size and is frequently used when triangular shell elements have very short altitudes. The bar wave speed, Equation (21.10), is used to compute the time step size and L_s is given by

$$L_s = \max \left[\frac{(1 + \beta)A_s}{\max(L_1, L_2, L_3, (1 - \beta)L_4)}, \min(L_1, L_2, L_3, L_4 + \beta 10^{20}) \right]. \quad (26.19)$$

A comparison of critical time steps of truss versus shells is given in Table 22.2 with β defined as:

$$\beta = \frac{\Delta t_{2D\text{-continuum}}}{\Delta t_{\text{rod}}} = \frac{C_{\text{rod}}}{C} = \sqrt{1 - v^2}. \quad (26.20)$$

	0	0.2	0.3	0.4	0.5
	1.0	0.98	0.954	0.917	0.866

Table 22.2. Comparison of critical time step sizes for a truss versus a shell element.

26.4 Time Step Calculations for Solid Shell Elements

A critical time step size, Δt_e is computed for solid shell elements from

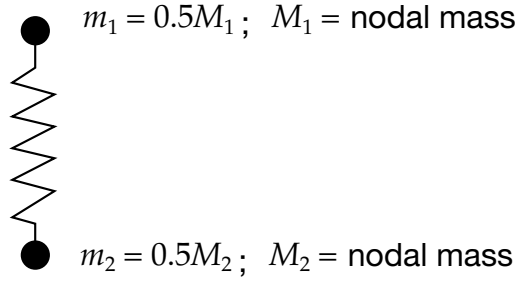
$$\Delta t_e = \frac{v_e}{cA_{e_{\max}}}, \quad (26.21)$$

where v_e is the element volume, $A_{e_{\max}}$ is the area of the largest side, and c is the plane stress sound speed given in [Equation \(26.16\)](#).

26.5 Time Step Calculations for Discrete Elements

For spring elements such as that in [Figure 26.1](#) there is no wave propagation speed c to calculate the critical time step size.

The eigenvalue problem for the free vibration of spring with end masses m_1 and m_2 , and stiffness, k , is given by:

**Figure 26.1.** Lumped spring mass system.

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (26.22)$$

Since the determinant of the characteristic equation must equal zero, we can solve for the maximum eigenvalue:

$$\det \begin{bmatrix} k - \omega^2 m_1 & -k \\ -k & k - \omega^2 m_2 \end{bmatrix} = 0 \rightarrow \omega_{\max}^2 = \frac{k(m_1 + m_2)}{m_1 \cdot m_2}, \quad (26.23)$$

Recalling the critical time step of a truss element:

$$\left. \begin{array}{l} \Delta t \leq \frac{\ell}{c} \\ \omega_{\max} = \frac{2c}{\ell} \end{array} \right\} \Delta t \leq \frac{2}{\omega_{\max}}, \quad (26.24)$$

we get the time step as

$$\Delta t_e = 2 \sqrt{\frac{m_1 m_2}{k(m_1 + m_2)}} = \sqrt{\frac{2m}{k}}, \quad (26.25)$$

where we have introduced

$$m = \frac{2m_1 m_2}{m_1 + m_2}. \quad (26.26)$$

as the harmonic mean of the end masses

Approximating the spring masses by as using 1/2 the actual nodal mass (i.e. a series of springs where each node is shared by two elements), we obtain the time step in terms of the nodal mass:

$$\Delta t_e = \sqrt{\frac{2M_1 M_2}{k(M_1 + M_2)}}. \quad (26.27)$$

The springs used in the contact interface are not checked for stability.

26.6 Time Step Calculations for Cohesive Elements

Cohesive elements connect faces of solid/shell elements, or edges of shell elements, using a cohesive law. The cohesive law relates separation distance between the faces/edges to forces in a spring like fashion.

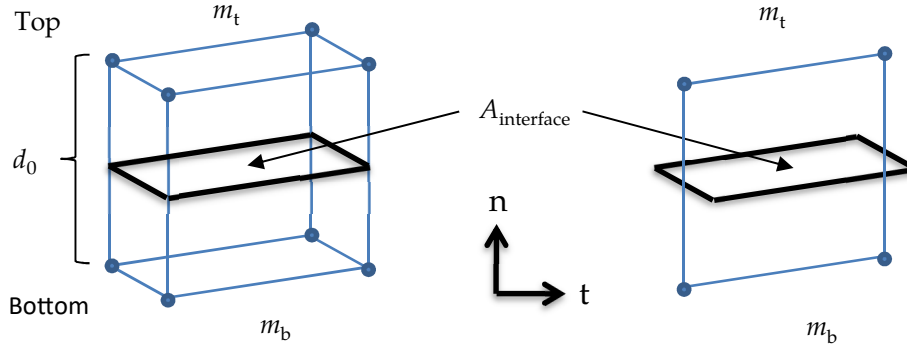


Figure 26.2. Solid and shell cohesive element with initial separation d_0 and top and bottom masses m_t and m_b . The interface area of the shell element is defined using the shell thickness.

The time step for a cohesive element is calculated similarly to a spring, i.e.

$$\Delta t_e = 2 \sqrt{\frac{m_t m_b}{k(m_t + m_b)}} \quad (26.28)$$

where m_t and m_b are the top and bottom masses. These masses are computed using the assembled nodal masses and thus include contributions from neighboring elements. See parameter ICOH on *CONTROL_SOLID for information how these are computed.

The spring stiffness (force per length) can be written as

$$k = A_{\text{interface}} \max(E_n, E_t), \quad (26.29)$$

where E_n and E_t are the normal/tangential cohesive stiffness (force per cubic length) typically given on the material cards.

For a solid cohesive element connecting two shell elements in a regular mesh, the top and bottom masses can be approximated as

$$\begin{aligned} m_t &= A_t h_t \rho_t + \frac{1}{2} M_{\text{coh}}, \\ m_b &= A_b h_b \rho_b + \frac{1}{2} M_{\text{coh}}, \end{aligned} \quad (26.30)$$

where A_* , h_* and ρ_* are the area, thickness, and density of the top and bottom shells, respectively, and (using volume density)

$$M_{\text{coh}} = \rho_{\text{coh}} d_0 A_{\text{interface}} \quad (26.31)$$

is the cohesive element mass contribution.

Assuming $A_t = A_b = A_{\text{interface}}$, the time step can now be estimated as

$$\Delta t_e = 2 \sqrt{\frac{m_t m_b}{k(m_t + m_b)}} \approx 2 \sqrt{\frac{\left(h_t \rho_t + \frac{1}{2} \rho_{\text{coh}} d_0\right) \left(h_b \rho_b + \frac{1}{2} \rho_{\text{coh}} d_0\right)}{\max(E_n, E_t) (h_t \rho_t + h_b \rho_b + \rho_{\text{coh}} d_0)}}. \quad (26.32)$$

That is, the cohesive time step is largely unaffected by the interface area, and in thus also the in-plane resolution of the mesh.

27

Boundary and Loading Conditions

27.1 Pressure Boundary Conditions

Consider pressure loadings on boundary ∂b_1 in Equation (2.4). To carry out the surface integration indicated by the integral

$$\int_{\partial b_1} N^t ds, \quad (27.1)$$

a Gaussian quadrature rule is used. To locate any point of the surface under consideration, a position vector, r , is defined:

$$r = f_1(\xi, \eta)i_1 + f_2(\xi, \eta)i_2 + f_3(\xi, \eta)i_3, \quad (27.2)$$

where

$$f_i(\xi, \eta) = \sum_{j=1}^4 \phi_j x_i^j, \quad (27.3)$$

and i_1, i_2, i_3 are unit vectors in the x_1, x_2, x_3 directions (see [Figure 27.1](#)).

Nodal quantities are interpolated over the four-node linear surface by the functions

$$\phi_i = \frac{1}{4} (1 + \xi \xi_i)(1 + \eta \eta_i), \quad (27.4)$$

so that the differential surface area ds may be written in terms of the curvilinear coordinates as

$$ds = |J| d\xi d\eta, \quad (27.5)$$

where $|J|$ is the surface Jacobian defined by

$$|J| = \left| \frac{\partial r}{\partial \xi} \times \frac{\partial r}{\partial \eta} \right| = (EG - F^2)^{1/2}, \quad (27.6)$$

in which

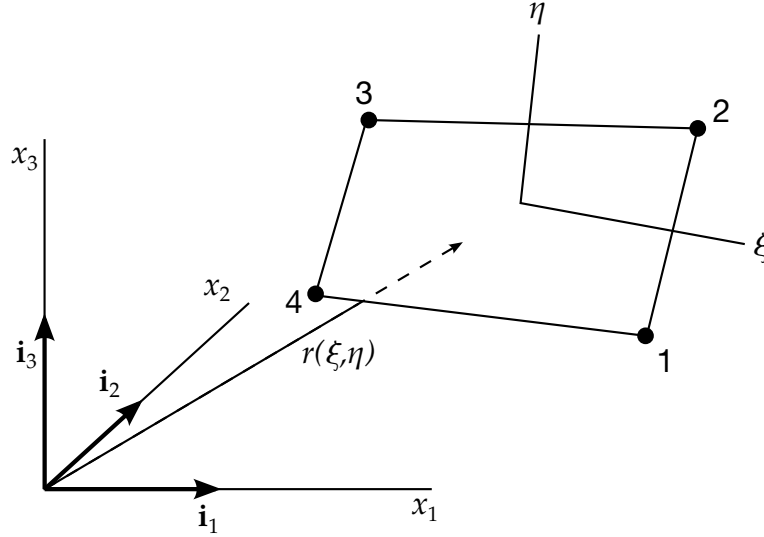


Figure 27.1. Parametric representation of a surface segment.

$$\begin{aligned}
 E &= \frac{\partial r}{\partial \xi} \cdot \frac{\partial r}{\partial \xi}, \\
 F &= \frac{\partial r}{\partial \xi} \cdot \frac{\partial r}{\partial \eta}, \\
 G &= \frac{\partial r}{\partial \eta} \cdot \frac{\partial r}{\partial \eta}.
 \end{aligned} \tag{27.7}$$

A unit normal vector \mathbf{n} to the surface segment is given by

$$\mathbf{n} = |J|^{-1} \left(\frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta} \right), \tag{27.8}$$

and the global components of the traction vector can now be written

$$t_i = n_i \sum_{j=1}^4 \phi_j p_j, \tag{27.9}$$

where p_j is the applied pressure at the j th node.

The surface integral for a segment is evaluated as:

$$\int_{-1}^1 \int_{-1}^1 N^t t |J| d\xi d\eta. \tag{27.10}$$

One such integral is computed for each surface segment on which a pressure loading acts. Note that the Jacobians cancel when [Equations \(27.8\) and \(27.7\)](#) are put into [Equation \(27.10\)](#). [Equation \(27.10\)](#) is evaluated with one-point integration analogous to that employed in the volume integrals. The area of an element side is approximated by $4|J|$ where $|J| = |J(0,0)|$.

27.2 Transmitting Boundaries

Transmitting boundaries are available only for problems that require the modeling of semi-infinite or infinite domains with solid elements and therefore are not available for beam or shell elements. Applications of this capability include problems in geomechanics and underwater structures.

The transmitting or silent boundary is defined by providing a complete list of boundary segments. In the approach used, discussed by Cohen and Jennings [1983] who in turn credit the method to Lysmer and Kuhlemeyer [1969], viscous normal shear stresses in Equation (23.11) are applied to the boundary segments:

$$\sigma_{\text{normal}} = -\rho c_d \mathbf{V}_{\text{normal}} \quad (27.11)$$

$$\sigma_{\text{shear}} = -\rho c_s \mathbf{V}_{\text{tan}}, \quad (27.12)$$

where ρ , c_d , and c_s are the material density, dilatational wave speed, and the shear wave speed of the transmitting media respectively. The magnitude of these stresses is proportional to the particle velocities in the normal, $\mathbf{V}_{\text{normal}}$, and tangential, \mathbf{V}_{tan} , directions. The material associated with each transmitting segment is identified during initialization so that unique values of the constants ρ , c_d , and c_s can be defined automatically.

27.3 Kinematic Boundary Conditions

In this subsection, the kinematic constraints are briefly reviewed. LS-DYNA tracks reaction forces for each type of kinematic constraint and provides this information as output if requested. For the prescribed boundary conditions, the input energy is integrated and included in the external work.

27.4 Displacement Constraints

Translational and rotational boundary constraints are imposed either globally or locally by setting the constrained acceleration components to zero. If nodal single point constraints are employed, the constraints are imposed in a local system. The user defines the local system by specifying a vector \mathbf{u}_l in the direction of the local x-axis \mathbf{x}_l , and a local in-plane vector \mathbf{v}_l . After normalizing \mathbf{u}_l , the local \mathbf{x}_l , \mathbf{y}_l and \mathbf{z}_l axes are given by:

$$\mathbf{x}_l = \frac{\mathbf{u}_l}{\|\mathbf{u}_l\|} \quad (27.13)$$

$$\mathbf{z}_I = \frac{\mathbf{x}_I \times \mathbf{v}_I}{\|\mathbf{x}_I \times \mathbf{v}_I\|} \quad (27.14)$$

$$\mathbf{y}_I = \mathbf{z}_I \times \mathbf{x}_I. \quad (27.15)$$

A transformation matrix \mathbf{q} is constructed to transform the acceleration components to the local system:

$$\mathbf{q} = \begin{bmatrix} \mathbf{x}_I^T \\ \mathbf{y}_I^T \\ \mathbf{z}_I^T \end{bmatrix}, \quad (27.16)$$

and the nodal translational and rotational acceleration vectors \mathbf{a}_I and $\dot{\boldsymbol{\omega}}_I$, for node I are transformed to the local system:

$$\mathbf{a}_{I_1} = \mathbf{q}\mathbf{a}_I \quad (27.17)$$

$$\dot{\boldsymbol{\omega}}_{I_1} = \mathbf{q}\dot{\boldsymbol{\omega}}_I, \quad (27.18)$$

and the constrained components are zeroed. The modified vectors are then transformed back to the global system:

$$\mathbf{a}_I = \mathbf{q}^T \mathbf{a}_{I_1} \quad (27.19)$$

$$\dot{\boldsymbol{\omega}}_I = \mathbf{q}^T \dot{\boldsymbol{\omega}}_{I_1} \quad (27.20)$$

27.5 Prescribed Displacements, Velocities, and Accelerations

Prescribed displacements, velocities, and accelerations are treated in a nearly identical way to displacement constraints. After imposing the zero displacement constraints, the prescribed values are imposed as velocities at time, $t^{n+1/2}$. The acceleration versus time curve is integrated or the displacement versus time curve is differentiated to generate the velocity versus time curve. The prescribed nodal components are then set.

27.6 Body Force Loads

Body force loads are used in many applications. For example, in structural analysis the base accelerations can be applied in the simulation of earthquake loadings, the gun firing of projectiles, and gravitational loads. The latter is often used with dynamic

relaxation to initialize the internal forces before proceeding with the transient response calculation. In aircraft engine design the body forces are generated by the application of an angular velocity of the spinning structure. The generalized body force loads are available if only part of the structure is subjected to such loadings, e.g., a bird striking a spinning fan blade.

For base accelerations and gravity we can fix the base and apply the loading as part of the body force loads element by element according to Equation (22.18)

$$\mathbf{f}_{e\text{body}} = \int_{v_m} \rho \mathbf{N}^T \mathbf{N} \mathbf{a}_{\text{base}} dv = \mathbf{m}_e \mathbf{a}_{\text{base}}, \quad (27.21)$$

where \mathbf{a}_{base} is the base acceleration and \mathbf{m}_e is the element (lumped) mass matrix.

28

Time Integration

28.1 Background

Consider the single degree of freedom damped system in [Figure 28.1](#). The forces acting on mass m are shown in [Figure 28.2](#). The equations of equilibrium are obtained from d'Alembert's principle

$$f_I + f_D + f_{\text{int}} = p(t) \quad (28.1)$$

$$\begin{aligned} f_I &= m\ddot{u} \\ f_D &= c\dot{u} \\ f_{\text{int}} &= ku \end{aligned} \quad (28.2)$$

where u is the displacement, c is the damping coefficient, and k is the linear stiffness. For critical damping $c = c_{\text{cr}}$. The equations of motion for linear behavior lead to a linear ordinary differential equation (ODE):

$$m\ddot{u} + c\dot{u} + ku = p(t) \quad (28.3)$$

but for the nonlinear case the internal force varies as a nonlinear function of the displacement, leading to a nonlinear ode.:

$$m\ddot{u} + c\dot{u} + f_{\text{int}}(u) = p(t) \quad (28.4)$$

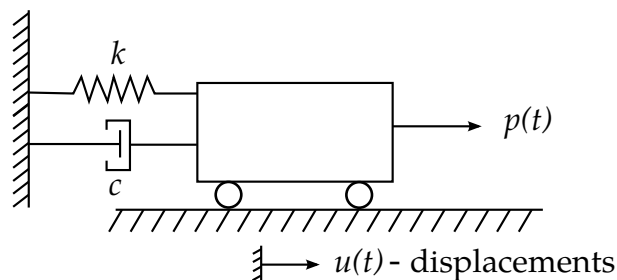


Figure 28.1. Single degree of freedom damped system.

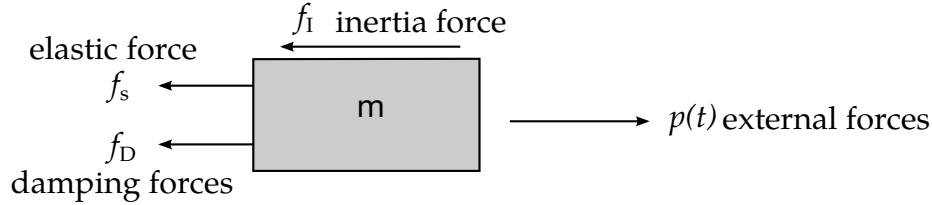


Figure 28.2. Forces acting on mass, m

Analytical solutions of linear ordinary differential equations are available, so instead we consider the dynamic response of the linear system subjected to a harmonic loading. It is convenient to define some commonly used terms:

- Harmonic loading: $p(t) = p_0 \sin \omega t$
- Angular frequency: $\omega = \sqrt{\frac{k}{m}}$ for single degree of freedom
- Natural frequency: $f = \frac{\omega}{2\pi} = \frac{1}{T}$, $T = \text{period}$
- Damping ratio: $\zeta = \frac{c}{c_{cr}} = \frac{c}{2m\omega_n}$
- Damped vibration frequency: $\omega_d = \omega \sqrt{1 - \zeta^2}$
- Applied load frequency: $\beta = \frac{\bar{\omega}}{\omega}$

Assuming $c = 0$, the closed form solution is:

$$u(t) = u_0 \cos \omega t + \frac{\dot{u}_0}{\omega} \sin \omega t + \frac{p_0}{k} \frac{1}{1 - \beta^2} (\sin \bar{\omega} t - \beta \sin \omega t) \quad (28.5)$$

with the initial conditions:

$$\begin{aligned} u_0 &= \text{initial displacement} \\ \dot{u}_0 &= \text{initial velocity} \\ p_0/k &= \text{static displacement} \end{aligned}$$

For nonlinear problems, only numerical solutions are possible. LS-DYNA uses the explicit central difference scheme to integrate the equations of motion.

28.2 The Central Difference Method

The semi-discrete equations of motion at time n are:

$$\mathbf{M} \mathbf{a}^n = \mathbf{P}^n - \mathbf{F}^n + \mathbf{H}^n, \quad (28.6)$$

where \mathbf{M} is the diagonal mass matrix, \mathbf{P}^n accounts for external and body force loads, \mathbf{F}^n is the stress divergence vector, and \mathbf{H}^n is the hourglass resistance. To advance to time t^{n+1} , we use central difference time integration:

$$\mathbf{a}^n = \mathbf{M}^{-1}(\mathbf{P}^n - \mathbf{F}^n + \mathbf{H}^n), \quad (28.7)$$

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a}^n \Delta t^n, \quad (28.8)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \mathbf{v}^{n+1/2} \Delta t^{n+1/2}, \quad (28.9)$$

where

$$\Delta t^{n+1/2} = \frac{(\Delta t^n + \Delta t^{n+1})}{2}, \quad (28.10)$$

and \mathbf{v} and \mathbf{u} are the global nodal velocity and displacement vectors, respectively. We update the geometry by adding the displacement increments to the initial geometry:

$$\mathbf{x}^{n+1} = \mathbf{x}^0 + \mathbf{u}^{n+1}. \quad (28.11)$$

We have found that, although more storage is required to store the displacement vector, the results are much less sensitive to round-off error.

28.3 Stability of Central Difference Method

The stability of the central difference scheme is determined by looking at the stability of a linear system. The system of linear equations is uncoupled into the modal equations where the modal matrix of eigenvectors, $\boldsymbol{\phi}$, are normalized with respect to the mass and linear stiffness matrices \mathbf{K} , and \mathbf{M} , respectively, such that:

$$\begin{aligned} \boldsymbol{\phi}^T \mathbf{M} \boldsymbol{\phi} &= \mathbf{I} \\ \boldsymbol{\phi}^T \mathbf{K} \boldsymbol{\phi} &= \omega^2. \end{aligned} \quad (28.12)$$

With this normalization, we obtain for viscous proportional damping the decoupling of the damping matrix, \mathbf{C} :

$$\boldsymbol{\phi}^T \mathbf{C} \boldsymbol{\phi} = 2\zeta\omega \quad (28.13)$$

The equations of motion in the modal coordinates \mathbf{x} are:

$$\ddot{\mathbf{x}} + 2\zeta\omega\dot{\mathbf{x}} + \omega^2\mathbf{x} = \underbrace{\boldsymbol{\phi}^T \mathbf{p}}_{=\mathbf{Y}}. \quad (28.14)$$

With central differences we obtain for the velocity and acceleration:

$$\dot{x}_n = \frac{x_{n+1} - x_{n-1}}{2\Delta t} \quad (28.15)$$

$$\ddot{x}_n = \frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2}. \quad (28.16)$$

Substituting \dot{x}_n and \ddot{x}_n into equation of motion at time t^n leads to:

$$x_{n+1} = \frac{2 - \omega^2 \Delta t^2}{1 + \zeta \omega \Delta t^2} x_n - \frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t} x_{n-1} + \frac{\Delta t^2}{1 + \zeta \omega \Delta t^2} Y_n, \quad (28.17)$$

$$x_n = x_n, \quad (28.18)$$

which in matrix form leads to

$$\begin{bmatrix} x_{n+1} \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{2 - \omega^2 \Delta t^2}{1 + \zeta \omega \Delta t^2} & -\frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{1 + \zeta \omega \Delta t^2} \\ 0 \end{bmatrix} Y_n, \quad (28.19)$$

or

$$\hat{\mathbf{x}}_{n+1} = \mathbf{A} \hat{\mathbf{x}}_n + \mathbf{L} Y_n, \quad (28.20)$$

where, \mathbf{A} is the time integration operator for discrete equations of motion. After m time steps with $\mathbf{L} = 0$, we obtain:

$$\hat{\mathbf{x}}_m = \mathbf{A}^m \hat{\mathbf{x}}_0. \quad (28.21)$$

As m approaches infinity, \mathbf{A} must remain bounded.

A spectral decomposition of \mathbf{A} gives:

$$\mathbf{A}^m = (\mathbf{P}^T \mathbf{J} \mathbf{P})^m = \mathbf{P}^T \mathbf{J}^m \mathbf{P}, \quad (28.22)$$

where \mathbf{P} is the orthonormal matrix containing the eigenvectors of \mathbf{A} , and \mathbf{J} is the Jordan form with the eigenvalues on the diagonal. The spectral radius, $\rho(\mathbf{A})$, is the largest eigenvalue of $\mathbf{A} = \max [\text{diag. } (\mathbf{J})]$. We know that \mathbf{J}^m , is bounded if and only if:

$$|\rho(\mathbf{A})| \leq 1. \quad (28.23)$$

Consider the eigenvalues of \mathbf{A} :

$$\det \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0, \quad (28.24)$$

$$\lambda^2 - 2A_1 \lambda + A_2 = 0, \quad (28.25)$$

$$\lambda = A_1 \pm \sqrt{A_1^2 - A_2}, \quad (28.26)$$

where

$$A_1 = \frac{1}{2} \text{tr}[\mathbf{A}] = \frac{1}{2} (A_{11} + A_{22}), \quad (28.27)$$

$$A_2 = \det[\mathbf{A}] = A_{11}A_{22} - A_{21}A_{12}.$$

The eigenvalues of \mathbf{A} can either be: 1.) complex conjugate (if $A_1^2 < A_2$) or; 2.) real (if $A_1^2 \geq A_2$). The requirement that $|\lambda| \leq 1$ leads to the following stability conditions

$$\begin{aligned} 1.) \quad & -(A_2 + 1) \leq 2A_1 \leq A_2 + 1 \quad -1 \leq A_2 < 1 \\ 2.) \quad & -1 < A_1 < 1 \quad A_2 = 1 \end{aligned} \quad (28.28)$$

Consider the time integration operator for the undamped equations of motion:

$$\mathbf{A} = \begin{bmatrix} 2 - \omega^2 \Delta t^2 & -1 \\ 1 & 0 \end{bmatrix}, \quad (28.29)$$

giving

$$\begin{aligned} A_1 &= 1 - \frac{\omega^2 \Delta t^2}{2}, \\ A_2 &= 1, \end{aligned} \quad (28.30)$$

The stability condition is then $-1 < A_1 < 1$, and we have a bounded solution for

$$\Delta t < \frac{2}{\omega}, \quad (28.31)$$

as the critical time step.

For the damped equations of motion, we have:

$$\mathbf{A} = \begin{bmatrix} \frac{2 - \omega^2 \Delta t^2}{1 + \zeta \omega \Delta t} & -\frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t} \\ 1 & 0 \end{bmatrix}, \quad (28.32)$$

and

$$\begin{aligned} A_1 &= \frac{1 - \frac{1}{2} \omega^2 \Delta t^2}{1 + \zeta \omega \Delta t}, \\ A_2 &= \frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t}. \end{aligned} \quad (28.33)$$

The stability conditions are

$$\begin{aligned} -\frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t} - 1 &\leq \frac{2 - \omega^2 \Delta t^2}{1 + \zeta \omega \Delta t} \leq \frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t} + 1, \\ -1 &\leq \frac{1 - \zeta \omega \Delta t}{1 + \zeta \omega \Delta t} < 1. \end{aligned} \quad (28.34)$$

The second condition is always true for $\zeta > 0$, and the right inequality of the first condition is true for $\omega > 0$. The left inequality of the first condition is fulfilled if the time step is

$$\Delta t \leq \frac{2}{\omega}, \quad (28.35)$$

which is similar to the undamped case. However, the time step condition now does no longer involve a strict inequality.

In LS-DYNA, the viscous forces are computed using the preceding mid-step velocities. That is, in the equations of motion we use

$$\dot{x}_{n-\frac{1}{2}} = \frac{x_n - x_{n-1}}{\Delta t} \quad (28.36)$$

$$\ddot{x}_n = \frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2}. \quad (28.37)$$

The time integration operator is then

$$\mathbf{A} = \begin{bmatrix} 2 - \omega^2 \Delta t^2 - 2\zeta\omega\Delta t & -(1 - 2\zeta\omega\Delta t) \\ 1 & 0 \end{bmatrix}, \quad (28.38)$$

and

$$\begin{aligned} A_1 &= 1 - \frac{1}{2}\omega^2 \Delta t^2 - \zeta\omega\Delta t, \\ A_2 &= 1 - 2\zeta\omega\Delta t. \end{aligned} \quad (28.39)$$

The stability conditions are

$$\begin{aligned} -(2 - 2\zeta\omega\Delta t) &\leq 2 - \omega^2 \Delta t^2 - 2\zeta\omega\Delta t \leq 2 - 2\zeta\omega\Delta t, \\ -1 &\leq 1 - 2\zeta\omega\Delta t < 1. \end{aligned} \quad (28.40)$$

The right inequalities are always fulfilled in both conditions above. The left inequalities lead to two conditions on the time step; however, the first condition will always be limiting. For a bounded solution the time step must fulfill

$$\Delta t \leq \frac{2}{\omega} \left(\sqrt{1 + \zeta^2} - \zeta \right). \quad (28.41)$$

Thus, when using the mid-step velocity, damping reduces the critical time step size. The time step size is bounded by the largest natural frequency of the structure which, in turn, is bounded by the highest frequency of any individual element in the finite element mesh.

28.4 Subcycling (Mixed Time Integration)

The time step size, Δt , is always limited by a single element in the finite element mesh. The idea behind subcycling is to sort elements based on their step size into groups whose step size is some even multiple of the smallest element step size, $2^{(n-1)}\Delta t$, for integer values of n greater than or equal to 1. For example, in [Figure 28.3](#) the mesh on the right because of the thin row of elements is three times more expensive than the mesh on the left.

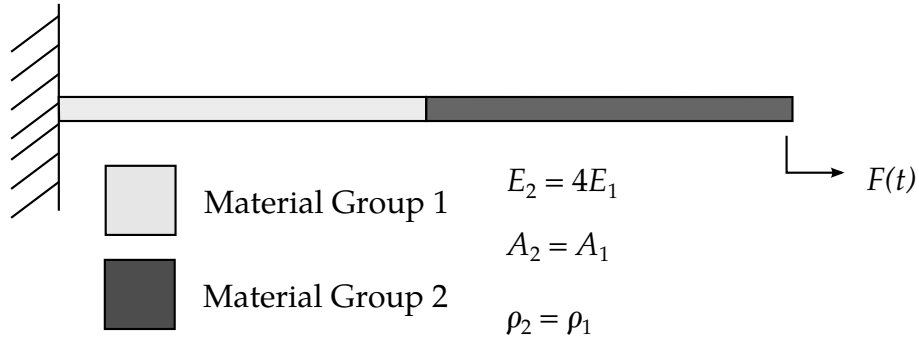


Figure 28.4. Subcycled beam problem from Hulbert and Hughes [1988].

Subcycling in LS-DYNA is based on the linear nodal interpolation partition subcycling algorithm of Belytschko, Yen, and Mullen [1979], and Belytschko [1980]. In their implementation, the steps are:

1. Assign each node, i , a time step size, Δt_i , according to:

$$\Delta t_i = \min \left(2/\omega_j \right)$$

over all elements, j , connected to node i .

2. Assign each element, j , a time step size, Δt_j , according to:

$$\Delta t_j = \min(\Delta t_i)$$

over all nodes, i , of element j

3. Group elements into blocks by time step size for vectorization.

In LS-DYNA we desire to use constant length vectors as much as possible even if it means updating the large elements incrementally with the small time step size. We have found that doing this decreases costs with stability unaffected.

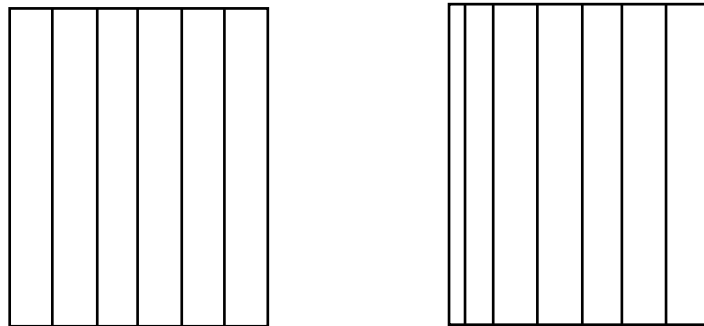


Figure 28.3. The right hand mesh is much more expensive to compute than the left hand due to the presence of the thinner elements.

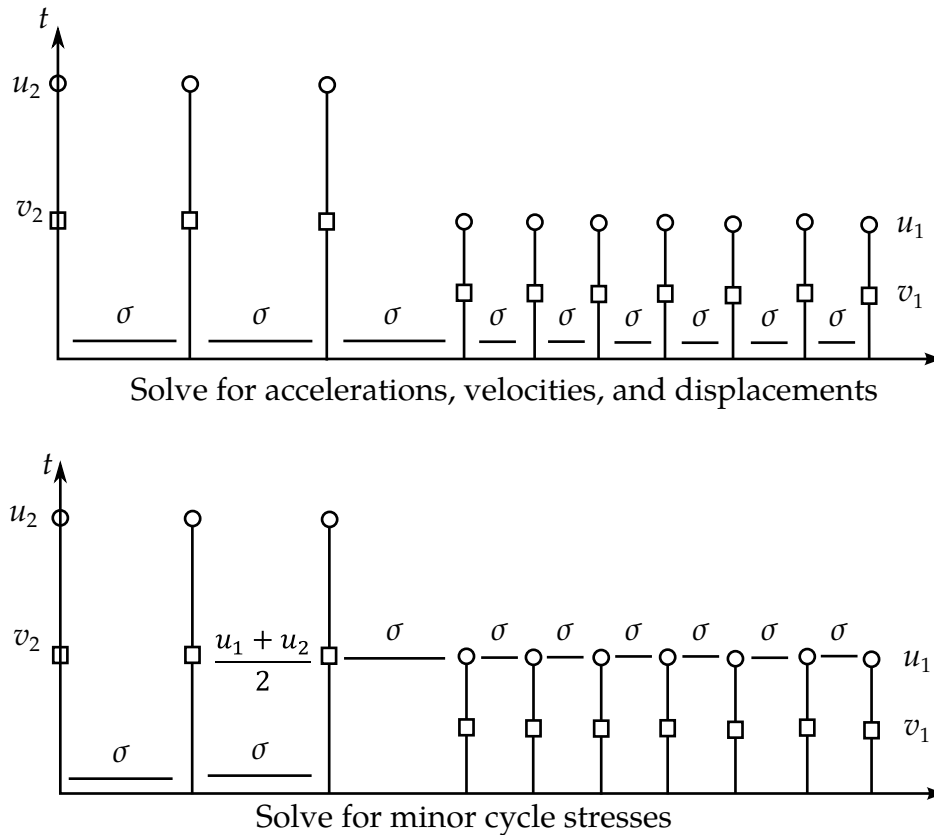


Figure 28.5. Timing diagram for subcycling algorithm based on linear nodal interpolations.

Hulbert and Hughes [1988] reviewed seven subcycling algorithms in which the partitioning is either node or element based. The major differences within the two subcycling methods lie in how elements along the interface between large and small elements are handled, a subject which is beyond the scope of this theoretical manual. Nevertheless, they concluded that three of the methods, including the linear nodal interpolation method chosen for LS-DYNA, provide both stable and accurate solutions for the problems they studied. However, there was some concern about the lack of stability and accuracy proofs for any of these methods when applied to problems in structural mechanics.

The implementation of subcycling currently includes the following element classes and contact options:

- Solid elements

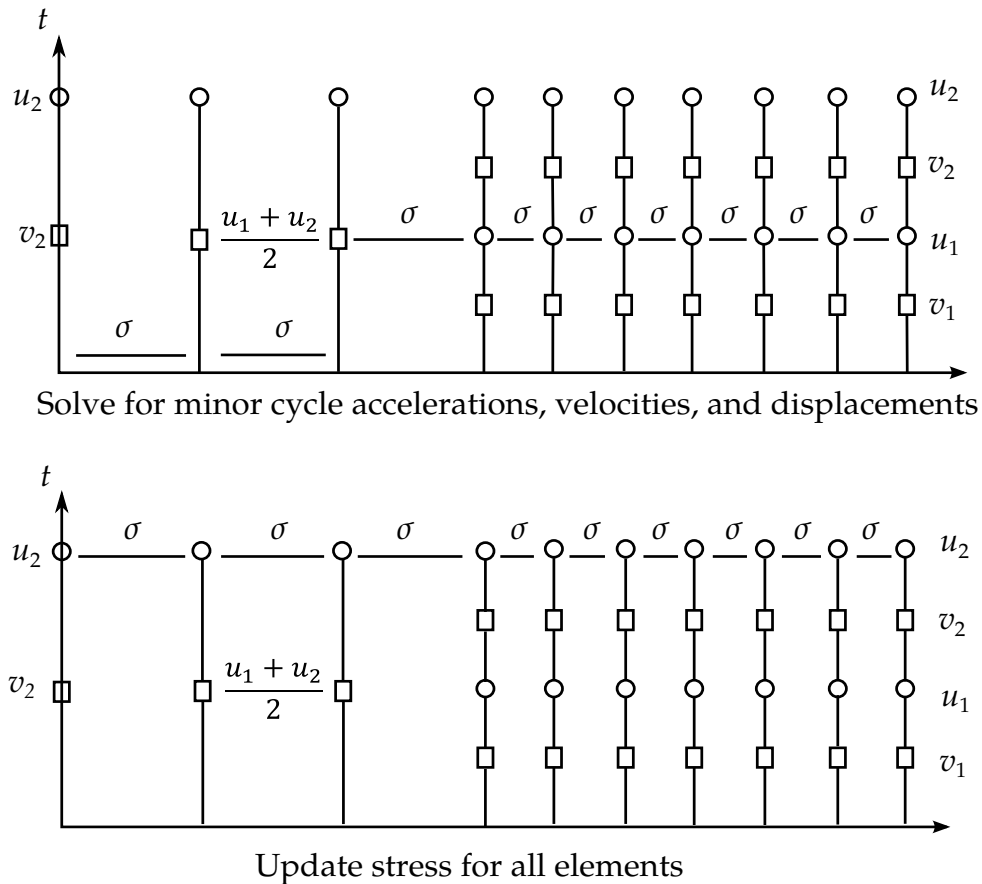


Figure 28.6. Timing diagram for subcycling algorithm based on linear nodal interpolations.

- Beam elements
- Shell elements
- Brick shell elements
- Penalty based contact algorithms

but intentionally excludes discrete elements since these elements generally contribute insignificantly to the calculational costs. The interface stiffnesses used in the contact algorithms are based on the minimum value of the tracked node or reference segment stiffness. Consequently, the time step size determination for elements on either side of the interface is assumed to be decoupled. Thus, scaling penalty values to larger values when subcycling is active can be a dangerous exercise. Nodes that are included in constraint equations, rigid bodies, or in contact with rigid walls are always assigned the smallest time step sizes.

To explain the implementation and the functioning of subcycling, we consider the beam shown in [Figure 28.4](#) where the beam elements on the right (material group 2) have a Courant time step size exactly two times greater than the elements on the left. The nodes attached to material group 2 will be called minor cycle nodes and the rest, major

cycle nodes. At time step $n = mk$ all nodal displacements and element stresses are known, where m is the ratio between the largest and smallest time step size, k is the number of major time steps, and n is the number of minor time steps. In [Figures 28.5](#) and [28.6](#), the update of the state variables to the next major time step $k + 1$ is depicted. The stress state in the element on the material interface in group 1 is updated during the minor cycle as the displacement of the major cycle node shared by this element is assumed to vary linearly during the minor cycle update. This linear variation of the major cycle nodal displacements during the update of the element stresses improves accuracy and stability.

In the timing study results in [Table 1.1](#), fifty solid elements were included in each group for the beam depicted in [Figure 28.4](#), and the ratio between E_2 to E_1 was varied from 1 to 128 giving a major time step size greater than 10 times the minor. Note that as the ratio between the major and minor time step sizes approaches infinity the reduction in cost should approach 50 percent for the subcycled case, and we see that this is indeed the case. The effect of subcycling for the more expensive fully integrated elements is greater as may be expected. The overhead of subcycling for the case where $E_1 = E_2$ is relatively large. This provides some insight into why a decrease in speed is often observed when subcycling is active. For subcycling to have a significant effect, the ratio of the major to minor time step size should be large and the number of elements having the minor step size should be small. In crashworthiness applications the typical mesh is very well planned and generated to have uniform time step sizes; consequently, subcycling will probably give a net increase in costs.

<u>Case 1</u> one point integration with elastic material model	Number of cycles	cpu time(secs)
$E_2 = E_1$	178	4.65
	178	5.36 (+15.%)
$E_2 = 4E_1$	367	7.57
	367	7.13 (-6.0%)
$E_2 = 16E_1$	714	12.17
	715	10.18 (-20.%)
$E_2 = 64E_1$	1417	23.24
	1419	16.39 (-29.%)
$E_2 = 128E_1$	2003	31.89
	2004	22.37 (-30.%)

<u>Case 2</u> eight point integration with orthotropic material model	Number of cycles	cpu time(secs)
$E_2 = E_1$	180	22.09
	180	22.75 (+3.0%)

$E_2 = 4E_1$	369 369	42.91 34.20 (-20.%)
$E_2 = 16E_1$	718 719	81.49 54.75 (-33.%)
$E_2 = 64E_1$	1424 1424	159.2 97.04 (-39.%)
$E_2 = 128E_1$	2034 2028	226.8 135.5 (-40.%)

Table 28.1. Timing study showing effects of the ratio of the major to minor time step size.

The impact of the subcycling implementation in the software has a very significant effect on the internal structure. The elements in LS-DYNA are now sorted three times

- By element number in ascending order
- By material number for large vector blocks
- By connectivity to ensure disjointness for right hand side vectorization which is very important for efficiency.

Sorting by Δt interacts with the second and third sorts and can result in the creation of much smaller vector blocks and a higher cost per element time step. During the simulation elements can continuously change in time step size and resorting may be required to maintain stability. Consequently, we must check for this continuously. Sorting cost, though not high when spread over the entire calculation, can become a factor that results in higher overall cost if done too frequently especially if the factor, m , is relatively small and the ratio of small to large elements is large.

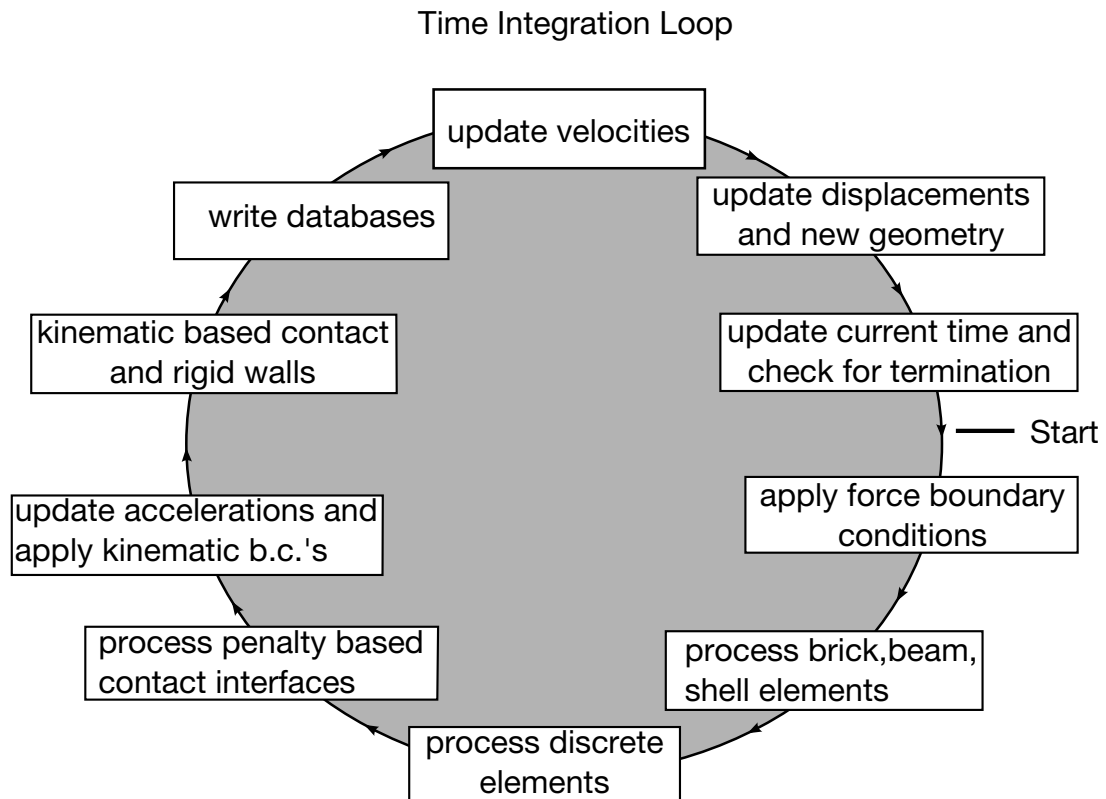


Figure 28.7. The time integration loop in LS-DYNA.

29

Rigid Body Dynamics

A detailed discussion of the rigid body algorithm is presented by Benson and Hallquist [1986] and readers are referred to this publication for more information.

29.1.1 Equations of Motion for the Body

The equations of motion for a rigid body are given by

$$M_\rho \ddot{\mathbf{x}} = \mathbf{f}_\rho^x, \quad (29.1)$$

$$\mathbf{J}_\rho \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}_\rho \boldsymbol{\omega} = \mathbf{f}_\rho^\omega, \quad (29.2)$$

where M_ρ is the *physical* mass, \mathbf{J}_ρ is the *physical* inertia tensor, \mathbf{x} is the location of the center of mass, $\boldsymbol{\omega}$ is the angular velocity of the body, and \mathbf{f}_ρ^x and \mathbf{f}_ρ^ω are the forces and torques applied to the rigid body through *LOAD_RIGID_BODY. These are equations that can be found in a standard text book on rigid body mechanics. The physical properties of a rigid body may come from three sources, these are

1. Integration of the mass density ρ over a region V occupied by the rigid body, for which $M_\rho = \int \rho dV$, and $\mathbf{J}_\rho = \int \rho (\mathbf{y} - \mathbf{x}) \otimes (\mathbf{y} - \mathbf{x}) dV$. The initial rigid body coordinate \mathbf{x} is in this case determined from $\mathbf{x} = \frac{\int \rho \mathbf{y} dV}{M_\rho}$. Here \mathbf{y} is the integrand variable.
2. Specifying properties using *PART_INERTIA, for which M_ρ , \mathbf{J}_ρ and initial coordinate \mathbf{x} is simply specified in the keyword input deck.
3. For a nodal rigid body, *CONSTRAINED_NODAL_RIGID_BODY, the physical properties vanish, i.e., $M_\rho = 0$ and $\mathbf{J}_\rho = \mathbf{0}$, and the position is arbitrary.

Note that these are equations for the body itself, disregarding any interactions it may have with its surroundings.

29.1.2 Rigid Body Node Constraints

All rigid bodies possess *rigid body nodes*, which play a role when rigid bodies in LS-DYNA interact with their surroundings. Rigid body nodes may come from the following.

- 1.The nodes in the finite element mesh for the part specified as rigid through *MAT_-RIGID.
- 2.Extra nodes definitions through *CONSTRAINED_EXTRA_NODES.
- 3.The set used for a nodal rigid body in *CONSTRAINED_NODAL_RIGID_BODY.

We use S to denote the set of nodes to the rigid body, and these are constrained to the rigid body through the equations

$$x_i = x + Q(x_i^0 - x^0), \quad (29.3)$$

$$Q_i = Q, \quad (29.4)$$

for all $i \in S$. Here we have introduced the orientations Q and Q_i of the rigid body and node i , respectively. Furthermore, x_i is the coordinate of node i and we use superscript 0 to denote a quantity at time zero. The time evolution of Q is

$$\dot{Q} = \Omega Q, \quad (29.5)$$

where $\Omega r = \omega \times r$ for an arbitrary vector r and $Q = I$ (identity) at time zero. So [equations \(29.3\) and \(29.4\)](#) can equivalently be put in rate form

$$\dot{x}_i = \dot{x} + \omega \times r_i, \quad (29.6)$$

$$\omega_i = \omega, \quad (29.7)$$

where $r_i = x_i - x$ and understandably ω_i rotational velocity of node i . This also determines the space of admissible virtual displacement for the nodes in the context of work principles, and for this reason we use a compact notation for this equation

$$\begin{bmatrix} \dot{x}_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} I & -R_i \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{x} \\ \omega \end{bmatrix}. \quad (29.8)$$

where $R_i r = r_i \times r$ for an arbitrary vector r .

29.1.3 Equations of Motion for the Nodes

Nodes may have masses m_i , inertias J_i and forces f_i^x and f_i^ω associated with them. The inertia properties may come from

- 1.Mass contributions from deformable elements connected to the rigid body, either through *CONSTRAINED_EXTRA_NODES or *CONSTRAINED_NODAL_RIGID_BODY or simply merged mesh.
- 2.Lumped masses through *ELEMENT_MASS or *ELEMENT_INERTIA.

and the forces may come from

- 1.External loads through *LOAD_NODE or *LOAD_SEGMENT.
- 2.Contacts or fluid structure interaction (FSI) or similar.
- 3.Internal forces on deformable nodes of a contiguous part.

Note that these quantities exclude any contributions from and on the rigid body itself, as these are all collected in M_ρ , J_ρ , f_ρ^x and f_ρ^ω . The motion of the nodes is governed by their own equations of motion

$$m_i \ddot{\mathbf{x}}_i = \mathbf{f}_i^x, \quad (29.9)$$

$$\mathbf{J}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \mathbf{J}_i \boldsymbol{\omega}_i = \mathbf{f}_i^\omega, \quad (29.10)$$

for $i \in S$.

29.1.4 The Generalized Equations of Motion

We seek a set of equations for the rigid body that combines (29.1)-(29.2) and (29.9)-(29.10) by condensing out the dependence of the nodes through (29.6)-(29.7). Differentiating (29.6)-(29.7) with respect to time yields

$$\ddot{\mathbf{x}}_i = \ddot{\mathbf{x}} + \dot{\boldsymbol{\omega}} \times \mathbf{r}_i + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_i, \quad (29.11)$$

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}, \quad (29.12)$$

which can be inserted into (29.9)-(29.10) to yield

$$m_i [\ddot{\mathbf{x}} + \dot{\boldsymbol{\omega}} \times \mathbf{r}_i] = \mathbf{f}_i^x - m_i \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_i, \quad (29.13)$$

$$\mathbf{J}_i \dot{\boldsymbol{\omega}} = \mathbf{f}_i^\omega - \boldsymbol{\omega} \times \mathbf{J}_i \boldsymbol{\omega}. \quad (29.14)$$

This can be compactly written as

$$\begin{bmatrix} m_i & -m_i \mathbf{R}_i \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i^x - m_i \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_i \\ \mathbf{f}_i^\omega - \boldsymbol{\omega} \times \mathbf{J}_i \boldsymbol{\omega} \end{bmatrix}. \quad (29.15)$$

It remains to use the principle of virtual work, employing (29.8), to reduce the number of equations (rigid body and nodes) to the generalized rigid body equations. The result of this endeavor is

$$\left(M_\rho + \sum_{i \in S} m_i \right) \ddot{\mathbf{x}} - \left(\sum_{i \in S} m_i \mathbf{R}_i \right) \dot{\boldsymbol{\omega}} = \mathbf{f}_\rho^x + \sum_{i \in S} (\mathbf{f}_i^x - m_i \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_i), \quad (29.16)$$

$$\begin{aligned} & - \left(\sum_{i \in S} m_i \mathbf{R}_i^T \right) \ddot{\mathbf{x}} + \left(\mathbf{J}_\rho + \sum_{i \in S} \mathbf{J}_i + \sum_{i \in S} m_i \mathbf{R}_i^T \mathbf{R}_i \right) \dot{\boldsymbol{\omega}} \\ & = \mathbf{f}_\rho^\omega + \sum_{i \in S} (\mathbf{f}_i^\omega - \boldsymbol{\omega} \times \mathbf{J}_i \boldsymbol{\omega}) - \sum_{i \in S} \mathbf{R}_i^T (\mathbf{f}_i^x - m_i \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_i) - \boldsymbol{\omega} \times \mathbf{J}_\rho \boldsymbol{\omega}, \end{aligned} \quad (29.17)$$

A simplified expression can be obtained through the variable substitution

$$\mathbf{z} = \mathbf{x} + \frac{1}{M} \sum_{i \in S} m_i \mathbf{r}_i. \quad (29.18)$$

This yields $\ddot{\mathbf{x}} = \ddot{\mathbf{z}} + \mathbf{R}_{\mathbf{z}-\mathbf{x}} \dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_{\mathbf{z}-\mathbf{x}}$ where $\mathbf{r}_{\mathbf{z}-\mathbf{x}} = \mathbf{z} - \mathbf{x}$ and $\mathbf{R}_{\mathbf{z}-\mathbf{x}} \mathbf{r} = \mathbf{r}_{\mathbf{z}-\mathbf{x}} \times \mathbf{r}$ for an arbitrary vector \mathbf{r} . This can be inserted into (29.16) and (29.17) to provide

$$M \ddot{\mathbf{z}} = \mathbf{f}^x, \quad (29.19)$$

$$\mathbf{J} \dot{\boldsymbol{\omega}} = \mathbf{f}^\omega. \quad (29.20)$$

where

$$M = M_\rho + \sum_{i \in S} m_i, \quad (29.21)$$

$$f^x = f_\rho^x + \sum_{i \in S} f_i^x, \quad (29.22)$$

$$J = J_\rho + \sum_{i \in S} J_i + \sum_{i \in S} m_i R_i^T R_i - M R_{z-x}^T R_{z-x}, \quad (29.23)$$

$$f^\omega = -\omega \times J \omega + f_\rho^\omega + \sum_{i \in S} (f_i^\omega + r_i \times f_i^x) - r_{z-x} \times f^x. \quad (29.24)$$

Equations (29.19)-(29.24) are the generalized rigid body equations to be solved for x and Q , cf. (29.5) and (29.18). The mass in (29.21) and inertia tensor in (29.23) are the physical mass and physical inertia augmented by node properties; nodal masses m_i , inertias J_i and locations x_i . We denote M the *algorithmic* mass, which may not reflect what the user intuitively expects when using *MAT_RIGID to make a part rigid. Similarly, J is the algorithmic inertia tensor, and it is worth noting that a nodal rigid body must therefore be connected to deformable elements or otherwise $M = 0$ and $J = 0$ and its whereabouts will be impossible to determine. For no mass scaling, all these properties are constant (except for rotational updates of the inertia tensors) and can essentially be calculated at time zero. If mass scaling is active, the nodal masses and inertias *include* the added mass due to mass scaling and therefore change over time. This means that inertia properties should be recomputed every time step to account for these changes, but the default behavior is that this is done *only* for nodal rigid bodies and not for regular rigid bodies. Presumably this assumes that the influence from nodes is significant for nodal rigid bodies and not so much for regular rigid bodies, which is probably true as long as the number of contiguous nodes is small compared to the total number of nodes in the rigid body. Nevertheless, with RBSMS = 1 on *CONTROL_RIGID, these extra masses are accounted for and equations (29.19)-(29.24) are solved as expressed herein. This amounts to transforming x to z before the update, then update z , and transform back to obtain the new x .

29.1.5 Constraints

Assume that we want to impose a motion at an arbitrary point that is fixed in the body. To this end, we call this coordinate y and it is related to z according to some⁷

$$y = z + Q(y^0 - z^0). \quad (29.25)$$

If we include constraints on rotations, the constraint is written as the system of linear equations

$$C y = c \quad (29.26)$$

$$G \omega = g \quad (29.27)$$

⁷ Note that z^0 in this equation is not necessarily constant over time, as this point is not fixed relative to the body itself, we can however assume so in the context of deriving the resulting equations.

where we assume that C is a constant matrix with CC^T being the identity matrix of some dimension, while c is a vector of the same dimension that depends on time. We have the same assumptions for G and g . Differentiating the first of these twice and using the expression for \dot{y} results in

$$C\ddot{z} + CR_{z-y}\dot{\omega} = \ddot{c} + C(\omega \times \omega \times (z - y)). \quad (29.28)$$

To simplify the notation, we set $D = CR_{z-y}$ and $d = \ddot{c} + C(\omega \times \omega \times (z - y))$ to end up with

$$C\ddot{z} + D\dot{\omega} = d. \quad (29.29)$$

For the constraint on rotations, the same treatment leads to

$$G\dot{\omega} = \dot{g} \quad (29.30)$$

Treating these constraints with Lagrangian multipliers results in the following equations of motion

$$M\ddot{z} + C^T\lambda = f^x \quad (29.31)$$

$$J\dot{\omega} + D^T\lambda + G^T\mu = f^\omega \quad (29.32)$$

that are combined with the constraint equation itself.

29.1.6 Algorithmic Update

From the most recent equations of motion, we have expressions for \ddot{z} and $\dot{\omega}$ in terms of λ and μ

$$\ddot{z} = \frac{1}{M}(f^x - C^T\lambda) \quad (29.33)$$

$$\dot{\omega} = J^{-1}(f^\omega - D^T\lambda - G^T\mu). \quad (29.34)$$

These can be inserted into the constraint equation and solved for λ

$$\lambda = A^{-1}\left(\frac{1}{M}Cf^x + Ff^\omega - d + DH^TB^{-1}\dot{g}\right) \quad (29.35)$$

where $H = GJ^{-1}$ and $B = HG^T$ while

$$F = D(J^{-1} - H^TB^{-1}H) \quad (29.36)$$

$$A = \frac{1}{M}I + FD^T. \quad (29.37)$$

Once λ is obtained, then μ is given by

$$\mu = B^{-1}(Hf^\omega - \dot{g}) - B^{-1}H^TD^T\lambda. \quad (29.38)$$

These are inserted into the previous expressions to completely determine \ddot{z} and $\dot{\omega}$.

For arbitrary orientations of the body, the inertia tensor is transformed each time step based on the incremental rotations using the standard rules of second-order tensors:

$$J = AJ^nA^T \quad (29.39)$$

where J is the updated inertia tensor components in the global frame. The transformation matrix A is not stored since the formulation is incremental but recomputed as explained below. After calculating the rigid body accelerations, the rigid body translational and rotational increment, Δz and $\Delta\theta$, can be calculated using the time step Δt and the explicit time integration update. The translational coordinate is then updated as

$$\mathbf{z} = \mathbf{z}^n + \Delta \mathbf{z} \quad (29.40)$$

and $\Delta \boldsymbol{\theta}$ is used to calculate \mathbf{A} using the Hughes-Winget algorithm,

$$\mathbf{A} = \mathbf{I} + \frac{1}{1 + \frac{\Delta \boldsymbol{\theta}^T \Delta \boldsymbol{\theta}}{4}} \left(\mathbf{I} + \frac{1}{2} \Delta \mathbf{S} \right) \Delta \mathbf{S}, \quad (29.41)$$

$$\Delta \mathbf{S} \mathbf{r} = \Delta \boldsymbol{\theta} \times \mathbf{r}, \quad \forall \mathbf{r}. \quad (29.42)$$

Similarly to \mathbf{J} , we update the orientation of the rigid body as

$$\mathbf{Q} = \mathbf{A} \mathbf{Q}^n \mathbf{A}^T \quad (29.43)$$

and then we can obtain the predicted rigid body center of mass $\tilde{\mathbf{x}}$. As a final step, we need to make sure that the body is satisfying the constraint we may have, and not drifting. To this end we simply correct $\tilde{\mathbf{x}}$ as

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{C}^T \left(\mathbf{c} - \mathbf{C} \left(\tilde{\mathbf{x}} + \mathbf{Q}(\mathbf{y}^0 - \mathbf{x}^0) \right) \right). \quad (29.44)$$

The rigid body nodes are incrementally updated

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta \mathbf{x} + (\mathbf{A} - \mathbf{I}) \mathbf{r}_i^n, \quad (29.45)$$

and the velocities of the nodes are calculated by differencing the coordinates

$$\dot{\mathbf{x}}_i = \frac{(\mathbf{x}_i^{n+1} - \mathbf{x}_i^n)}{\Delta t}. \quad (29.46)$$

A direct integration of the rigid body accelerations into velocity and displacements is not used for two reasons: (1) calculating the rigid body accelerations of the nodes is more expensive than the current algorithm, and (2) the second-order accuracy of the central difference integration method would introduce distortion into the rigid bodies. Since the accelerations are not needed within the program, they are calculated by a post-processor using a difference scheme like the above.

29.2 Rigid Body Joints

29.2.1 Penalty types

The joints for the rigid bodies in LS-DYNA, see [Figure 29.1](#), are implemented using the penalty method. Given a constraint equation $C(x_i, x_j) = 0$, for nodes i and j , the penalty function added to the Lagrangian of the system is $-1/2kC^2(x_i, x_j)$. The resulting nodal forces are:

$$\mathbf{f}_i = -kC(x_i, x_j) \frac{\partial C(x_i, x_j)}{\partial x_i}, \quad (29.47)$$

$$f_j = -kC(x_i, x_j) \frac{\partial C(x_i, x_j)}{\partial x_j}. \quad (29.48)$$

The forces acting at the nodes have to convert into forces acting on the rigid bodies. Recall that velocities of a node i is related to the velocity of the center of mass of a rigid body by Equation (29.6). By using Equation (29.6) and virtual power arguments, it may be shown that the generalized forces are:

$$F_i^x = f_i, \quad (29.49)$$

$$F_i^\omega = e_{ijk} \bar{x}_j f_k, \quad (29.50)$$

which are the forces and moments about the center of mass.

The magnitude of the penalty stiffness k is chosen so that it does not control the stable time step size. For the central difference method, the stable time step Δt is restricted by the condition that,

$$\Delta t = \frac{2}{\Omega}, \quad (29.51)$$

where Ω is the highest frequency in the system. The six vibrational frequencies associated with each rigid body are determined by solving their eigenvalue problems assuming $k = 1$. For a body with m constraint equations, the linearized equations of the translational degrees of freedom are

$$M\ddot{X} + mkX = 0, \quad (29.52)$$

and the frequency is $\sqrt{mk/M}$ where M is the mass of the rigid body. The corresponding rotational equations are

$$\mathbf{J}\ddot{\boldsymbol{\theta}} + \mathbf{K}\boldsymbol{\theta} = 0, \quad (29.53)$$

\mathbf{J} is the inertia tensor and \mathbf{K} is the stiffness matrix for the moment contributions from the penalty constraints. The stiffness matrix is derived by noting that the moment contribution of a constraint may be approximated by

$$\mathbf{F}^x = -k\mathbf{r}_i \times (\boldsymbol{\theta} \times \mathbf{r}_i), \quad (29.54)$$

$$\mathbf{r}_i = \mathbf{x}_i - \mathbf{X}^{\text{cm}}, \quad (29.55)$$

and noting the identity,

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = |\mathbf{A} \cdot \mathbf{C} - \mathbf{A} \otimes \mathbf{C}| \mathbf{B}, \quad (29.56)$$

so that

$$\mathbf{K} = \sum_{i=1}^m k[\mathbf{r}_i \cdot \mathbf{r}_i - \mathbf{r}_i \otimes \mathbf{r}_i]. \quad (29.57)$$

The rotational frequencies are the roots of the equation $\det[\mathbf{K} - \Omega^2 \mathbf{J}] = 0$, which is cubic in Ω^2 . Defining the maximum frequency over all rigid bodies for $k = 1$ as Ω_{\max} , and introducing a time step scale factor TSSF, the equation for k is

$$k \leq \left(\frac{2\text{TSSF}}{\Delta t \Omega_{\max}} \right)^2, \quad (29.58)$$

The joint constraints are defined in terms of the displacements of individual nodes. Regardless of whether the node belongs to a solid element or a structural element, only its translational degrees of freedom are used in the constraint equations.

A spherical joint is defined for nodes i and j by the three constraint equations,

$$x_{1i} - x_{1j} = 0 \quad x_{2i} - x_{2j} = 0 \quad x_{3i} - x_{3j} = 0, \quad (29.59)$$

and a revolute joint, which requires five constraints, is defined by two spherical joints, for a total of six constraint equations. Since a penalty formulation is used, the redundancy in the joint constraint equations is unimportant. A cylindrical joint is defined by taking a revolute joint and eliminating the penalty forces along the direction defined by the two spherical joints. In a similar manner, a planar joint is defined by eliminating the penalty forces that are perpendicular to the two spherical joints.

The translational joint is a cylindrical joint that permits sliding along its axis, but not rotation. An additional pair of nodes is required off the axis to supply the additional constraint. The only force active between the extra nodes acts in the direction normal to the plane defined by the three pairs of nodes.

The universal joint is defined by four nodes. Let the nodes on one body be i and k , and the other body, j and l . Two of them, i and j , are used to define a spherical joint for the first three constraint equations. The fourth constraint equation is,

$$C(x_i, x_j, x_k, x_l) = (x_k - x_i) \cdot (x_i - x_j) = 0, \quad (29.60)$$

and is differentiated to give the penalty forces $f_n = -kC \frac{\partial C}{\partial x_n}$, where n ranges over the four nodes numbers.

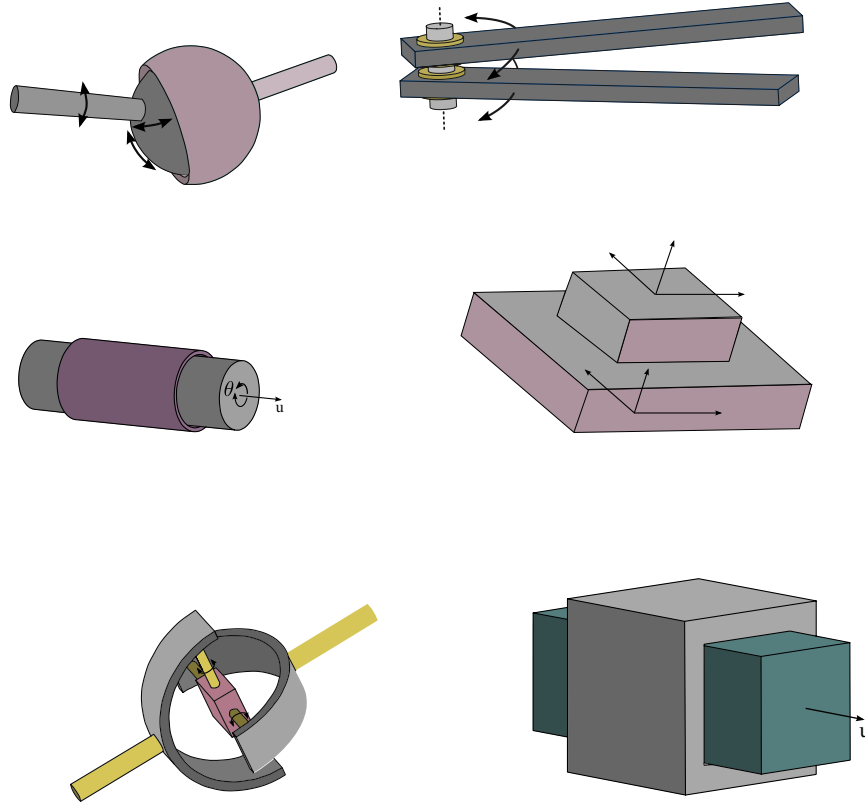


Figure 29.1. Rigid body joints in LS-DYNA.

29.2.2 Lagrange multiplier types

An alternate version of joints is to use constraints and associated Lagrange multipliers. To this end, we recall the numerical update of the rigid body kinematics. Given translational and rotational accelerations \ddot{x} and $\dot{\omega}$, we have

$$\dot{x} = \dot{x}_n + \Delta t_{n+1/2} \ddot{x} \quad (29.61)$$

$$x = x_n + \Delta t_{n+1} \dot{x} \quad (29.62)$$

and

$$\omega = \omega_n + \Delta t_{n+1/2} \dot{\omega} \quad (29.63)$$

$$Q = \left(I - \frac{1}{2} \Delta t_{n+1} R(\omega) \right)^{-1} \left(I + \frac{1}{2} \Delta t_{n+1} R(\omega) \right) Q_n \quad (29.64)$$

where $\Delta t_{n+1/2} = (\Delta t_{n+1} + \Delta t_n)/2$ and $R(\omega)v = \omega \times v$ for an arbitrary vector v . Here $Q = [q_1 \ q_2 \ q_3]$ is the orthonormal system attached to the rigid body, which we for simplicity assume is aligned with whatever joints we are considering. For instance, if two bodies i and j are connected with a joint we let $Q^i = Q^j$ initially, and let the vectors be aligned with the axes of the joint. Also, each body has a point p where the system is assumed to be attached and that defines the origin of the joint. Referring to the two bodies

exemplified, we assume that $p^i = p^j$ initially. The point p is for simplicity expressed as $p = x + d$, where d is a vector attached to the rigid body and updates in analogy with Q , i.e., using the Hughes-Winget formula (29.64).

For bodies with no constraints, the accelerations are given directly by the solution of the equations of motion for each individual body, but now we seek to solve the equations of motion with respect to the combined accelerations \ddot{X} and $\ddot{\Omega}$ of all bodies with joint constraints, subject to these constraints. This can be written as

$$M\ddot{X} - F_x = 0 \quad (29.65)$$

$$J\ddot{\Omega} - F_\theta = 0 \quad (29.66)$$

where we have collected quantities for all bodies of interest to form a complete system. Here M is the mass matrix, J is the inertia matrix while F_x and F_θ are the external forces and torques coming from all other features in the model. The constraints can be written

$$C(\ddot{X}, \ddot{\Omega}) = 0, \quad (29.67)$$

and a Lagrangian system is set up as

$$M\ddot{X} - F_x + \left(\frac{\partial C}{\partial \ddot{X}}\right)^T \Lambda = 0 \quad (29.68)$$

$$J\ddot{\Omega} - F_\theta + \left(\frac{\partial C}{\partial \ddot{\Omega}}\right)^T \Lambda = 0, \quad (29.69)$$

with Λ being the Lagrange multiplier. In the quest for a solution, we linearize this to form a Newton step according to

$$\begin{aligned} M\Delta\ddot{X} + \frac{\partial}{\partial \ddot{X}} \left\{ \left(\frac{\partial C}{\partial \ddot{X}}\right)^T \Lambda \right\} \Delta\ddot{X} + \frac{\partial}{\partial \ddot{\Omega}} \left\{ \left(\frac{\partial C}{\partial \ddot{X}}\right)^T \Lambda \right\} \Delta\ddot{\Omega} + \left(\frac{\partial C}{\partial \ddot{X}}\right)^T \Delta\Lambda \\ = F_x - M\ddot{X} - \left(\frac{\partial C}{\partial \ddot{X}}\right)^T \Lambda \end{aligned} \quad (29.70)$$

$$\begin{aligned} J\Delta\ddot{\Omega} + \frac{\partial}{\partial \ddot{X}} \left\{ \left(\frac{\partial C}{\partial \ddot{\Omega}}\right)^T \Lambda \right\} \Delta\ddot{X} + \frac{\partial}{\partial \ddot{\Omega}} \left\{ \left(\frac{\partial C}{\partial \ddot{\Omega}}\right)^T \Lambda \right\} \Delta\ddot{\Omega} + \left(\frac{\partial C}{\partial \ddot{\Omega}}\right)^T \Delta\Lambda \\ = F_\theta - J\ddot{\Omega} - \left(\frac{\partial C}{\partial \ddot{\Omega}}\right)^T \Lambda. \end{aligned} \quad (29.71)$$

This is accompanied with a linearization of the constraints themselves

$$\frac{\partial C}{\partial \ddot{X}} \Delta\ddot{X} + \frac{\partial C}{\partial \ddot{\Omega}} \Delta\ddot{\Omega} = -C \quad (29.72)$$

to form a solvable system of equations. Given a starting solution $\ddot{X} = 0$, $\ddot{\Omega} = 0$, and $\Lambda = 0$, the first system to solve is simplified to

$$M\Delta\ddot{X} + \left(\frac{\partial C}{\partial \ddot{X}}\right)^T \Delta\Lambda = F_x \quad (29.73)$$

$$J\Delta\dot{\Omega} + \left(\frac{\partial C}{\partial \dot{\Omega}}\right)^T \Delta\Lambda = F_\theta. \quad (29.74)$$

This, i.e., [equations \(29.72\)-\(29.74\)](#), is a standard symmetric Lagrange system of equations, and is often sufficient to yield an accurate solution without iterating. However, due to nonlinearity C may grow and therefore a fully iterative scheme is performed according at a time step frequency given by an internal rule. For simplicity, the approximation $\Lambda \approx 0$ is maintained for the setup of the system matrix to avoid the second derivative of constraints and a more complex nonsymmetric system. For the right hand side we make no such approximation.

Some of the joint constraints of interest are

Spherical joint

$$p^i - p^j = 0 \quad (29.75)$$

Revolute joint

$$p^i - p^j = 0 \quad (29.76)$$

$$q_1^i \bullet q_2^j = q_3^i \bullet q_1^j = 0 \quad (29.77)$$

Cylindrical joint

$$q_2^i \bullet (p^i - p^j) = q_3^i \bullet (p^i - p^j) = 0 \quad (29.78)$$

$$q_1^i \bullet q_2^j = q_3^i \bullet q_1^j = 0 \quad (29.79)$$

Planar joint

$$q_1^i \bullet (p^i - p^j) = 0 \quad (29.80)$$

$$q_1^i \bullet q_2^j = q_3^i \bullet q_1^j = 0 \quad (29.81)$$

Translational joint

$$q_2^i \bullet (p^i - p^j) = q_3^i \bullet (p^i - p^j) = 0 \quad (29.82)$$

$$q_1^i \bullet q_2^j = q_2^i \bullet q_3^j = q_3^i \bullet q_1^j = 0 \quad (29.83)$$

To form the system of equations, the first order sensitivities needed are those of Q , x and d , and can be expressed as

$$\delta q_i = \Delta t_{n+1} \left(I - \frac{1}{2} \Delta t_{n+1} R(\omega) \right)^{-1} \delta \omega \times \left\{ \left(I - \frac{1}{2} \Delta t_{n+1} R(\omega) \right)^{-1} q_i \right\} \quad (29.84)$$

$$\approx \Delta t_{n+1} \Delta t_{n+1/2} \delta \dot{\omega} \times q_i$$

$$\delta d = \dots \approx \Delta t_{n+1} \Delta t_{n+1/2} \delta \dot{\omega} \times d \quad (29.85)$$

$$\delta x = \Delta t_{n+1} \Delta t_{n+1/2} \delta \ddot{x} \quad (29.86)$$

where we have differentiated (29.61)-(29.64) using the accelerations as independent variables.

29.3 Deformable to Rigid Material Switching

Occasionally in practice, long duration, large rigid body motions arise that are prohibitively expensive to simulate if the elements in the model are deformable. Such a case could occur possibly in automotive rollover where the time duration of the rollover would dominate the cost relative to the post impact response that occurs much later.

To permit such simulations to be efficiently handled a capability to switch a subset of materials from deformable to rigid and back to deformable is available. In practice the suspension system and tires would remain deformable. A flag is set in the input to let LS-DYNA know that all materials in the model have the potential to become rigid bodies sometime during the calculation. When this flag is set a cost penalty is incurred on vector machines since the blocking of materials in the element loops will be based on the part ID rather than the material type ID. Normally this cost is insignificant relative to the cost reduction due to this unique feature.

For rigid body switching to work properly the choice of the shell element formulation is critical. The Hughes-Liu elements cannot currently be used for two reasons. First, since these elements compute the strains from the rotations of the nodal fiber vectors from one cycle to the next, the nodal fiber vectors should be updated with the rigid body motions and this is not done. Secondly, the stresses are stored in the global system as opposed to the co-rotational system. Therefore, the stresses would also need to be transformed with the rigid body motions or zeroed out. The co-rotational elements of Belytschko and co-workers do not reference nodal fibers for the strain computations and the stresses are stored in the co-rotational coordinate system which eliminates the need for the transformations; consequently, these elements can be safely used. The membrane elements and airbag elements are closely related to the Belytschko shells and can be safely used with the switching options.

The beam elements have nodal triads that are used to track the nodal rotations and to calculate the deformation displacements from one cycle to the next. These nodal triads are updated every cycle with the rigid body rotations to prevent non-physical behavior

when the rigid body is switched back to deformable. This applies to all beam element formulations in LS-DYNA. The Belytschko beam formulations are preferred for the switching options for like the shell elements, the Hughes-Liu beams keep the stresses in the global system. Truss elements like the membrane elements are trivially treated and pose no difficulties.

The brick elements store the stresses in the global system and upon switching the rigid material to deformable the element stresses are zeroed to eliminate spurious behavior.

The implementation addresses many potential problems and has worked well in practice. The current restrictions can be eliminated if the need arises and anyway should pose no insurmountable problems. We will continue to improve this capability if we find that it is becoming a popular option.

29.4 Rigid Body Welds

The weld capability in LS-DYNA is based on rigid body dynamics. Each weld is defined by a set of nodal points which moves rigidly with six degrees of freedom until a failure criteria is satisfied. Five weld options are implemented including:

- Spot weld.
- Fillet weld
- Butt weld
- Cross fillet weld
- General weld

Welds can fail three ways: by ductile failure which is based on the effective plastic strain, by brittle failure which is based on the force resultants acting on the rigid body weld, and by a failure time which is specified in the input. When effective plastic strain is used the weld fails when the nodal plastic strain exceeds the input value. A least squares algorithm is used to generate the nodal values of plastic strains at the nodes from the element integration point values. The plastic strain is integrated through the element and the average value is projected to the nodes with a least square fit. In the resultant based brittle failure the resultant forces and moments on each node of the weld are computed. These resultants are checked against a failure criterion which is expressed in terms of these resultants. The forces may be averaged over a user specified number of time steps to eliminate breakage due to spurious noise. After all nodes of a weld are released the rigid body is removed from the calculation.

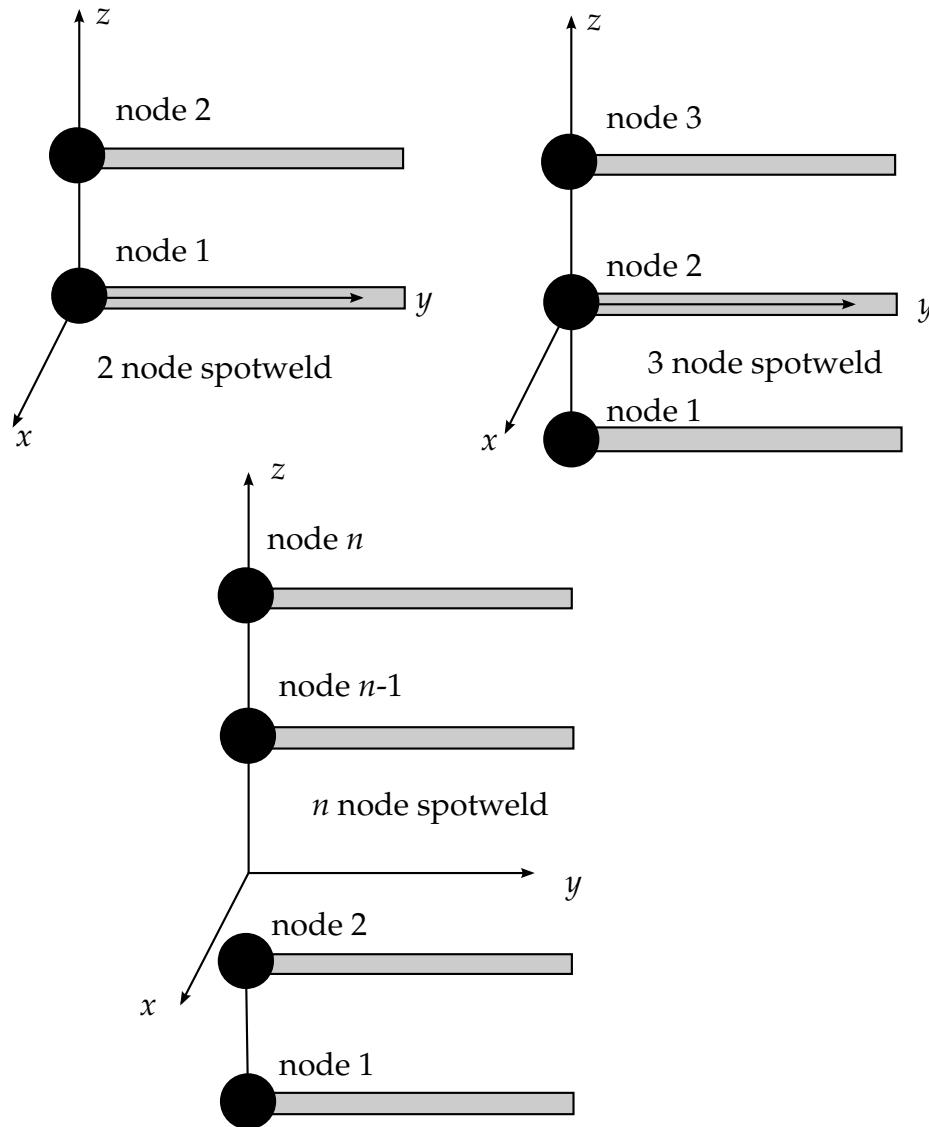


Figure 29.2. Nodal ordering and orientation of the local coordinate system is important for determining spotweld failure.

Spotwelds are shown in [Figure 29.2](#). Spotweld failure due to plastic straining occurs when the effective nodal plastic strain exceeds the input value, ϵ_{fail}^p . This option can model the tearing out of a spotweld from the sheet metal since the plasticity is in the material that surrounds the spotweld, not the spotweld itself. This option should only be used for the material models related to metallic plasticity and can result in slightly increased run times.

Brittle failure of the spotwelds occurs when:

$$\left(\frac{\max(f_n, 0)}{S_n} \right)^n + \left(\frac{|f_s|}{S_s} \right)^m \geq 1, \quad (29.87)$$

where f_n and f_s are the normal and shear interface force. Component f_n contributes for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes

inactive and the constrained nodes may move freely. In Figure 29.2 the ordering of the nodes is shown for the 2 and 3 noded spotwelds. This order is with respect to the local coordinate system where the local z axis determines the tensile direction. The nodes in the spotweld may coincide but if they are offset the local system is not needed since the z -axis is automatically oriented based on the locations of node 1, the origin, and node 2. The failure of the 3 noded spotweld may occur gradually with first one node failing and later the second node may fail. For n noded spotwelds the failure is progressive starting with the outer nodes (1 and n) and then moving inward to nodes 2 and $n - 1$. Progressive failure is necessary to preclude failures that would create new rigid bodies.

Ductile fillet weld failure, due to plastic straining, is treated identically to spotweld failure. Brittle failure of the fillet welds occurs when:

$$\beta \sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f, \quad (29.88)$$

where

σ_n = normal stress

τ_n = shear stress in direction of weld (local y)

τ_t = shear stress normal to weld (local x)

σ_f = failure stress

β = failure parameter

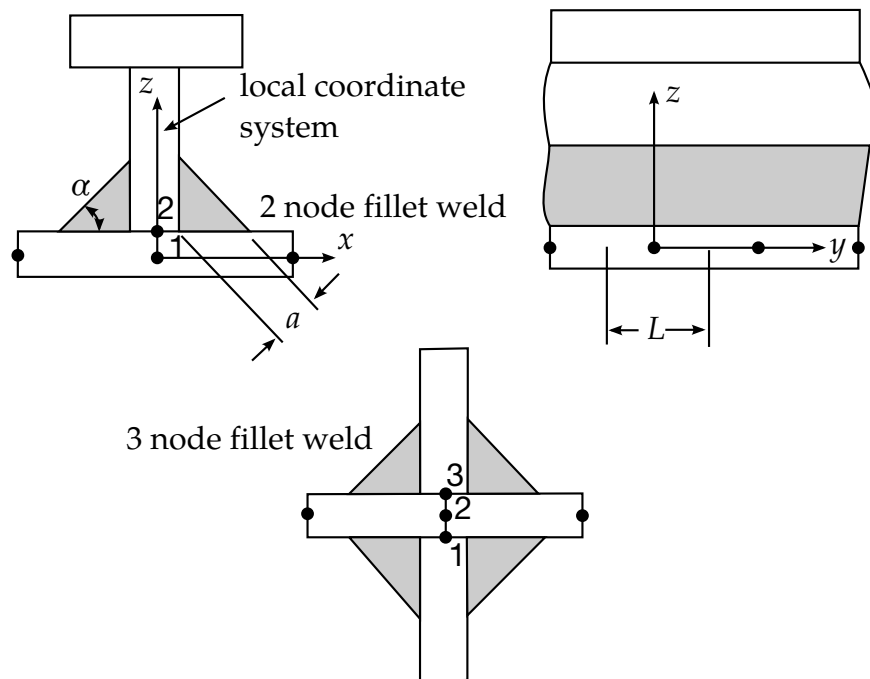


Figure 29.3. Nodal ordering and orientation of the local coordinate system is shown for fillet weld failure.

Component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. In Figure 29.3 the ordering of the nodes is shown for the 2 node and 3 node fillet welds. This order is with respect to the local coordinate system where the local z axis determines the tensile direction. The nodes in the fillet weld may coincide. The failure of the 3 node fillet weld may occur gradually with first one node failing and later the second node may fail.

In Figure 29.4 the butt weld is shown. Ductile butt weld failure, due to plastic straining, is treated identically to spotweld failure. Brittle failure of the butt welds occurs when:

$$\beta \sqrt{\sigma_n^2 + 3(\tau_n^2 + \tau_t^2)} \geq \sigma_f, \quad (29.89)$$

where

σ_n = normal stress

τ_n = shear stress in direction of weld (local y)

τ_t = shear stress normal to weld (local z)

σ_f = failure stress

β = failure parameter

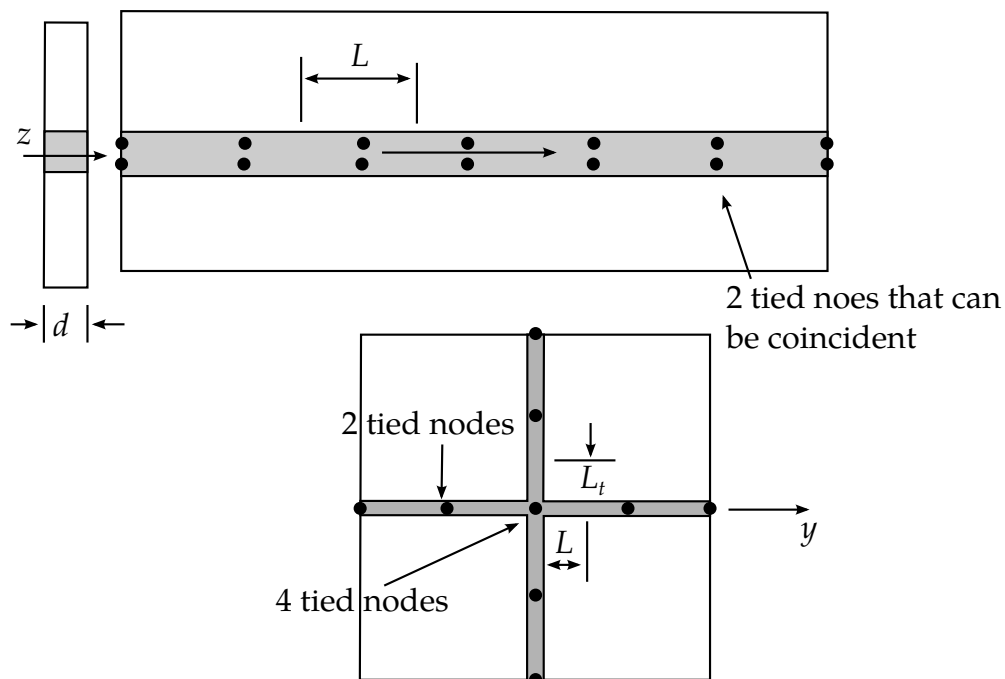


Figure 29.4. Orientation of the local coordinate system and nodal ordering is shown for butt weld failure.

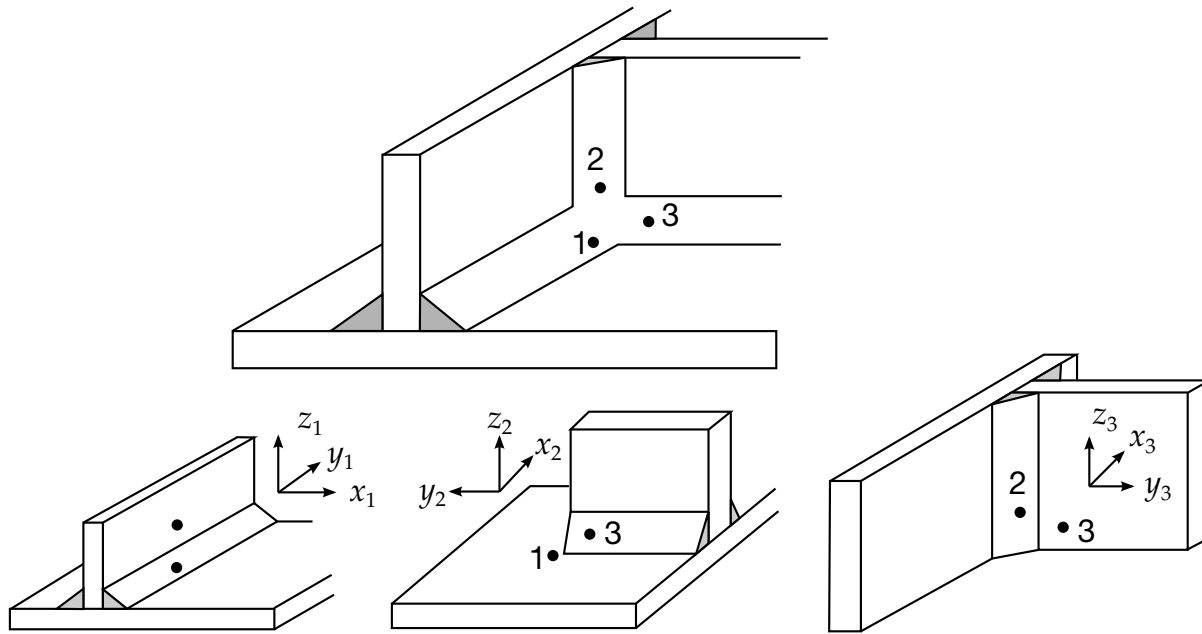


Figure 29.5. A simple cross fillet weld illustrates the required input. Here $NFW = 3$ with nodal pairs $(A = 2, B = 1)$, $(A = 3, B = 1)$, and $(A = 3, B = 2)$. The local coordinate axes are shown. These axes are fixed in the rigid body and are referenced to the local rigid body coordinate system which tracks the rigid body rotation.

Component σ_n is nonzero for tensile values only. When the failure time, t_f , is reached the nodal rigid body becomes inactive and the constrained nodes may move freely. The nodes in the butt weld may coincide.

The cross fillet weld and general weld are shown in [Figures 29.5](#) and [29.6](#), respectively. The treatment of failure for these welds is based on the formulation for the fillet and butt welds.

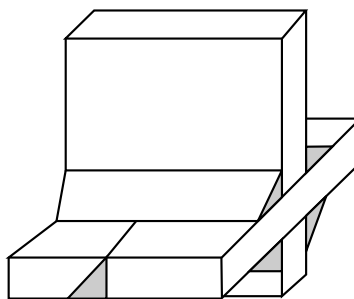


Figure 29.6. A general weld is a mixture of fillet and butt welds.

30

Contact-Impact Algorithm

30.1 Introduction

The treatment of sliding and impact along interfaces has always been an important capability in LS-DYNA and its predecessors. We have implemented three distinct methods for handling contact: the kinematic constraint, penalty, and distributed parameter. Of these, we use the first approach for tying interfaces. We discuss the relative merits of each approach in Sections [30.2](#), [30.3](#), and [30.4](#).

Interfaces can be defined in three dimensions by listing in arbitrary order all triangular and quadrilateral segments that comprise each side of the interface. We designate one side of the interface as the tracked side and the other as the reference side. We refer to nodes on those surfaces as tracked and reference nodes, respectively. In the symmetric penalty method, both sides of the interface act as the tracked and reference surfaces. But for the other methods, LS-DYNA constrains the tracked nodes to slide on the reference surface after impact and remain on the reference surface until a tensile force develops between the node and the surface. In the keyword input, surfaces A and B give the surfaces, where for nonsymmetric methods, A is the tracked surface and B is the reference surface.

Today, automatic contact definitions are commonly used. In this approach, LS-DYNA internally generates the tracked and reference surfaces from the part IDs given for each surface. For automotive crash models, including the entire vehicle in one single surface contact definition, where all the nodes and elements within the interface can interact, is quite common.

30.2 Kinematic Constraint Method

The kinematic constraint method uses the impact and release conditions of Hughes et al. [1976]. It was initially implemented in DYNA2D [Hallquist 1976b] and extended to three dimensions in DYNA3D. This method imposes constraints on the global equations by transforming the nodal displacement components of the tracked surface nodes along the

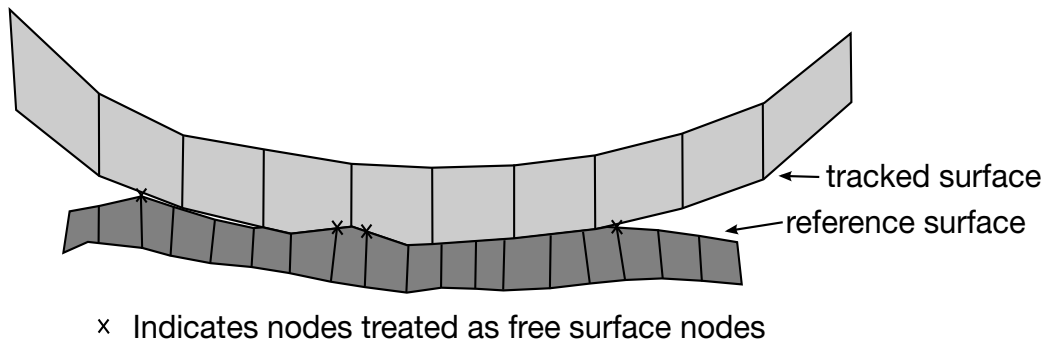


Figure 30.1. Nodes of the reference slide surface designated with an “x” are treated as free surface nodes in the nodal constraint method.

contact interface. This transformation has the effect of eliminating the normal degree of freedom of nodes. To preserve the efficiency of the explicit time integration, the mass is lumped to the extent that only the global degrees of freedom of each reference surface node are coupled. Impact and release conditions are imposed to ensure momentum conservation. The release conditions are only of academic interest and were quickly removed from the original implementations.

Problems arise with this method when the reference surface zoning is finer than the tracked surface zoning as shown in two dimensions in [Figure 30.1](#). Here, certain reference surface nodes can penetrate through the tracked surface without resistance and create a kink in the slide line. Such kinks are relatively common with this formulation, and, when interface pressures are high, these kinks occur whether one or more quadrature points are used in the element integration. It may be argued, of course, that better zoning would minimize such problems; but for many problems that are of interest, good zoning in the initial configuration may be very poor zoning later. Such is the case, for example, when gaseous products of a high-explosive gas expand against the surface of a structural member.

30.3 Penalty Method

The penalty method consists of placing normal interface springs between all penetrating nodes/segments and the reference segments. The algorithm chosen dictates whether LS-DYNA searches for segments penetrating segments or nodes penetrating segments. The penalty method is available for both explicit and implicit contact methods. Except for the spring stiffness matrix which must be assembled into the global stiffness matrix, the implicit and explicit treatments are similar. LS-DYNA calculates a penalty stiffness for each contact segment based on the element containing it. A scaling factor for the stiffness included in the keyword input controls the depth of penetration across the contact surface.

Quite in contrast to the nodal constraint method, the penalty method approach excites little, if any, mesh hourglassing. This lack of noise is undoubtedly attributable to the symmetry of the approach. Momentum is exactly conserved without the necessity of

imposing impact and release conditions. Furthermore, no special treatment of intersecting interfaces is required, greatly simplifying the implementation.

Currently, LS-DYNA has four implementations of the penalty algorithm:

- Standard Penalty Formulation
- Soft Constraint Penalty Formulation
- Segment-to-Segment Penalty Formulation
- Mortar Segment-to-Segment Penalty Formulation

The standard penalty formulation is the default penalty formulation for all penalty contacts other than Mortar contact. It is specified by setting $\text{SOFT} = 0$ on Optional Card A. Setting $\text{SOFT} = 1$ and 2 , respectively, in the input deck selects the soft constraint penalty formulation and segment-to-segment formulation. To enable Mortar contact, use any contact type with MORTAR in the name. SOFT is ignored in this case. $\text{SOFT} = 0$ and 1 search for nodes penetrating segments while $\text{SOFT} = 2$ and MORTAR look at segments penetrating segments.

In the standard penalty formulation, the interface stiffness is chosen to be approximately the same order of magnitude as the stiffness of the interface element normal to the interface. Consequently, the computed time step size is unaffected by the existence of the interfaces. However, if interface pressures become large, unacceptable penetration may occur. By scaling up the stiffness and scaling down the time step size, we may still solve such problems using the penalty approach. Since this increases the number of time steps and hence the cost, a sliding-only option has been implemented for treating explosive-structure interaction problems thereby avoiding use of the penalty approach. This latter option is based on a specialization of the third method described in Section 30.4.

We implemented the soft constraint penalty formulation to treat contact between bodies with dissimilar material properties (such as steel-foam). The stiffness calculation and its update during the simulation differ from the standard penalty formulation. The segment-to-segment penalty formulation is a powerful contact algorithm whose logic is a tracked segment-reference segment approach instead of a traditional tracked node-reference segment approach. With this contact, the algorithm chooses which segment is the tracked segment and which is the reference segment unless a one-way contact type is chosen. This contact has proven very useful for airbag self-contact during inflation and complex contact conditions. Mortar contact is a different kind of segment-to-segment contact suited for implicit analysis.

Section 30.6 provides a more detailed discussion of the different penalty methods other than Mortar contact. We discuss Mortar contact in Section 30.23.

30.4 Distributed Parameter Method

This method was used in DYNA2D, and a specialization of it was the sliding-only option in DYNA3D. In LS-DYNA, this method is used by 2D contacts with *OPTION1* set to SLIDING_ONLY or SLIDING_VOID. For 3D contacts, this method is enabled by setting

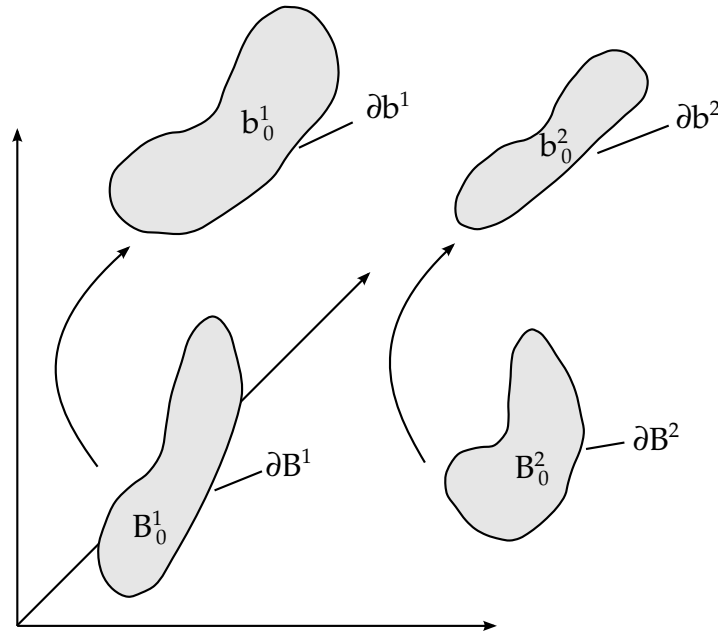


Figure 30.2. Reference and deformed configuration.

OPTION1 to *SLIDING_ONLY* or *SLIDING_ONLY_PENALTY*. Motivation for this approach came from the *TENSOR* [Burton et. al., 1982] and *HEMP* [Wilkins 1964] programs which displayed fewer mesh instabilities than *DYNA2D* with the nodal constraint algorithm. The first *DYNA2D* implementation of this last algorithm is described in detail by Hallquist [1978]. Since this early publication, the method has been moderately improved, but the major ideas remain the same.

In the distributed parameter formulation, half the mass of each tracked element in contact is distributed to the covered reference surface area. The internal stress in each tracked element determines a pressure distribution for the reference surface area that receives the mass. After completing this distribution of mass and pressure, we can update the acceleration of the reference surface. Constraints are then imposed on tracked nodes' accelerations and velocities to ensure their movement along the reference surface. Unlike the finite difference hydro programs, we do not allow tracked nodes to penetrate; therefore, we avoid “put back on” logic. In another simplification, our calculation of the tracked element's relative volume ignores any intrusion of the reference surfaces. The *HEMP* and *TENSOR* codes consider the reference surface in this calculation. For an outline of our sliding-only algorithm, see Section 30.15.

30.5 Preliminaries

Consider the time-dependent motion of two bodies occupying regions B^1 and B^2 in their undeformed configuration at time zero. Assume that the intersection

$$B^1 \cap B^2 = \emptyset \quad (30.1)$$

is satisfied. Let ∂B^1 and ∂B^2 denote the boundaries of B^1 and B^2 , respectively. At some later time, these bodies occupy regions b^1 and b^2 bounded by ∂b^1 and ∂b^2 as shown in [Figure 30.2](#). Because the deformed configurations cannot penetrate,

$$(b^1) \cap (b^2 - \partial b^2) = \emptyset. \quad (30.2)$$

As long as $(\partial b^1 \cap \partial b^2) = \emptyset$, the equations of motion remain uncoupled. In the foregoing and following equations, the right superscript α ($= 1, 2$) denotes the body to which the quantity refers.

Before a detailed description of the theory is given, some additional statements should be made concerning the terminology. The surfaces ∂b^1 and ∂b^2 of the discretized bodies b^1 and b^2 become the tracked and reference surfaces, respectively. For symmetric penalty treatment, both surfaces act as the tracked and reference surfaces, so when defining the surfaces, the input order of the surfaces is arbitrary. Otherwise, the more coarsely meshed surface should be chosen as the reference surface (second surface input) unless there is a large difference in mass densities in which case the side corresponding to the material with the highest density is recommended. Nodal points that define ∂b^1 are called tracked nodes and nodes that define ∂b^2 are called reference nodes. When $(\partial b^1 \cap \partial b^2) \neq \emptyset$, the constraints are imposed to prevent penetration. Right superscripts are implied whenever a variable refers to either the tracked surface ∂b^1 , or reference surface, ∂b^2 . Consequently, these superscripts are dropped in the development which follows.

30.6 Sliding with Closure and Separation

In this section, we will discuss all the penalty formulations specified with SOFT = 0, 1, and 2 on Optional Card A for 3D contact. [Section 30.23](#) discusses Mortar contact.

30.6.1 Standard Penalty Formulation

The standard penalty formulation is the default method for penalty contacts. For 3D contact, it can be specified with SOFT = 0 (the default value) on Optional Card A. In applying this penalty method, each tracked node is checked for penetration through the reference surface. If the tracked node does not penetrate, nothing is done. If it does penetrate, an interface force is applied between the tracked node and its contact point. The magnitude of this force is proportional to the amount of penetration. This may be thought of as the addition of an interface spring.

Penetration of tracked node n_t through the reference segment, containing its contact point, is indicated if

$$l = \mathbf{n}_i \times [\mathbf{t} - \mathbf{r}(\xi_c, \eta_c)] < 0, \quad (30.3)$$

where

$$\mathbf{n}_i = \mathbf{n}_i(\xi_c, \eta_c) \quad (30.4)$$

is normal to the reference segment at the contact point.

If tracked node n_t , has penetrated through reference segment, s_i , we add an interface force vector \mathbf{f}_t :

$$\mathbf{f}_t = -lk_i \mathbf{n}_i \quad \text{if } l < 0 \quad (30.5)$$

to the degrees of freedom corresponding to n_t and

$$f_r^j = \phi_j(\xi_c, \eta_c) f_r \quad \text{if } l < 0 \quad (30.6)$$

to the four nodes ($j = 1,2,3,4$) that comprise reference segment s_i . The stiffness factor k_i for reference segment s_i is given in terms of the bulk modulus K_i , the volume V_i , and the face area A_i of the element that contains s_i as

$$k_i = \frac{f_{si} K_i A_i^2}{V_i} \quad (30.7)$$

for hexahedral elements and

$$k_i = \frac{f_{si} K_i A_i}{\max(\text{shell diagonal})} \quad (30.8)$$

for shell elements. f_{si} is a scale factor for the interface stiffness and normally has a default value of .10. Larger values may cause instabilities unless the time step size is scaled back in the time step calculation.

In LS-DYNA, a number of options are available for setting the penalty stiffness value. This is often an issue since the materials in contact may have drastically different bulk moduli. The calculational choices are:

- Minimum of the reference segment and tracked node stiffness (default)
- Use reference segment stiffness
- Use tracked node value
- Use tracked node value, area or mass weighted
- Same as the previous one but inversely proportional to the shell thickness (generally not recommended)

See PENOPT on *CONTROL_CONTACT for setting this option.

The default may sometimes fail due to an excessively small stiffness. When this occurs, it is necessary to manually scale the interface stiffness. Care must be taken not to induce an instability when such scaling is performed. If the soft material also has a low density, it may be necessary to reduce the scale factor on the computed stable time step.

30.6.2 Soft Constraint Penalty Formulation

Very soft materials have an undesired effect on the contact stiffness, lowering its value and ultimately causing excessive penetration. The Soft Constraint Penalty Formulation (SOFT = 1 on Optional Card A for 3D contact) gives an alternative to applying a scale factor on the contact stiffness for SOFT = 0. The idea behind this option is to eliminate the excessive penetration by using a different formulation for the contact stiffness.

In addition to the reference and tracked contact stiffness, an additional stiffness is calculated, which is based on the stability (Courant's criterion) of the local system comprised of two masses (segments) connected by a spring. This is the stability contact stiffness k_{cs} and is calculated by:

$$k_{cs}(t) = 0.5 \times \text{SOFSCCL} \times m^* \times \left(\frac{1}{\Delta t_c(t)} \right)^2, \quad (30.9)$$

Here, SOFSCCL on Optional Card A of *CONTACT_OPTION is the scale factor for the soft constraint penalty formulation and m^* is a function of the mass of the tracked node and reference nodes. Δt_c is set to the initial solution time step. If the solution time step grows, Δt_c is reset to the current time step to prevent unstable behavior.

LS-DYNA performs a comparative check against the contact stiffness calculated with the traditional penalty formulation, $k_{\text{soft}=0}$. In general, LS-DYNA takes the maximum stiffness between the two:

$$k_{\text{soft}=1} = \max\{k_{cs}, k_{\text{soft}=0}\}. \quad (30.10)$$

30.6.3 Segment-to-Segment (SOFT = 2) Penalty Formulation

Segment-to-segment contact invoked with SOFT = 2 on Optional Card A of *CONTACT_OPTION is a general-purpose shell and solid element penalty type contact algorithm. This contact algorithm uses a contact stiffness similar to the SOFT = 1 stiffness option, but the details are quite different.

$$k_{cs}(t) = 0.5 \times \text{SLSFAC} \times \left\{ \begin{array}{l} \text{SFSA} \\ \text{or} \\ \text{SFSA} \end{array} \right\} \left(\frac{m_1 m_2}{m_1 + m_2} \right) \left(\frac{1}{\Delta t_c(t)} \right)^2. \quad (30.11)$$

Segment masses are used rather than nodal masses. Segment mass is equal to the element mass for shell segments and half the element mass for solid element segments. Like the soft constraint penalty formulation, Δt_c is set to the initial solution time step,

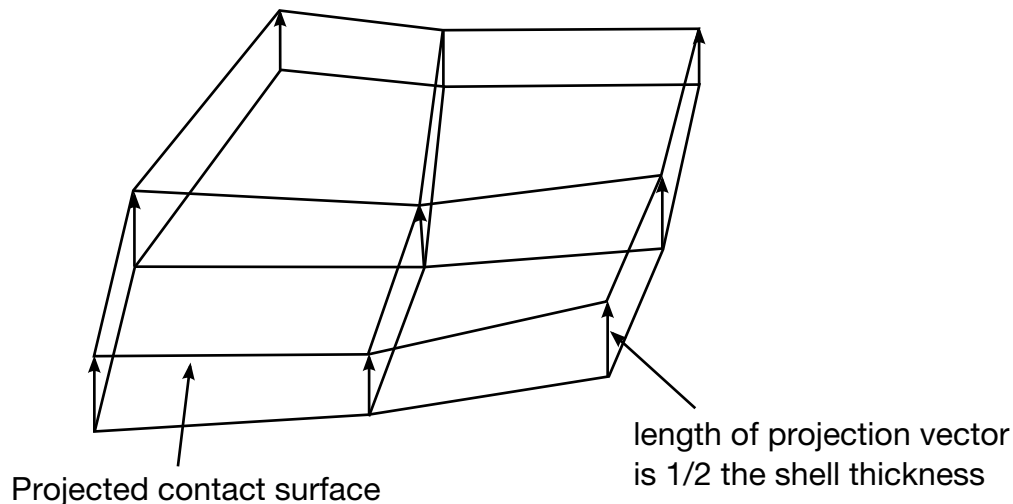


Figure 30.3. Contact surface is based on midsurface normal projection vectors.

which is updated if the solution time step grows larger to prevent unstable behavior.

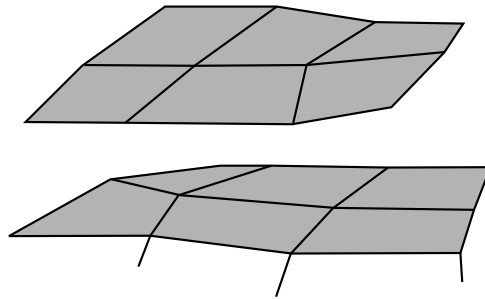


Figure 30.4. The tracked and reference surfaces must be offset in the input by half the total shell thickness. This also allows the segments to be oriented automatically.

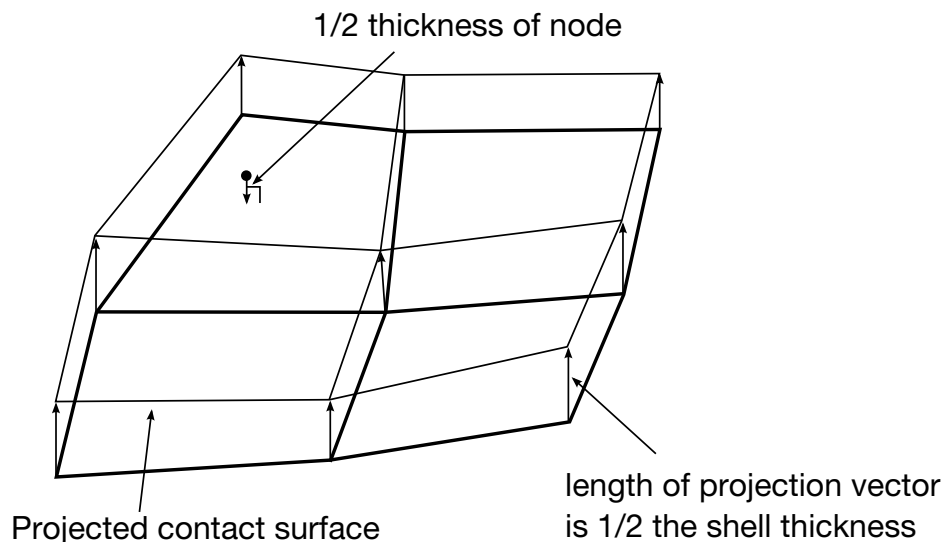


Figure 30.5. In a type 5 contact (nodes-to-surface contacts), thickness can also be taken into account.

However, it differs from $\text{SOFT} = 1$ in how Δt_c is updated. Δt_c is updated only if the solution time step grows by more than 5%. This allows Δt_c to remain constant in most cases, even if the solution time step slightly grows.

30.7 Accounting for Shell Thickness

Shell thickness effects are important when modeling sheet metal with shell elements. Unless thickness is considered in the contact, treating the effect of thinning on the frictional interface stresses due to membrane stretching is difficult.

In the treatment of thickness for surface-to-surface and single surface contacts, we project both the tracked and reference surfaces based on the midsurface normal projection vectors as shown in [Figure 30.3](#). The surfaces, therefore, must be offset by an amount equal to 1/2 their total thickness ([Figure 30.4](#)). This allows LS-DYNA to check the node numbering of the segments automatically to ensure that the shells are properly oriented.

Nodes-to-surface contacts treat the reference surface in the same way as the previously mentioned contacts. These contacts project the tracked node in a direction normal to the contact surface with half the nodal thickness as shown in [Figure 30.5](#). The nodal thickness is the maximum shell thickness over all the shells to which the node belongs.

Single surface contacts, automatic nodes-to-surface, automatic surface-to-surface contacts, and constraint-based contacts always consider shell thickness offsets. Non-automatic surface-to-surface (including one way) and non-automatic nodes-to-surface contacts do not by default account for shell thickness offsets. For these contacts, set SHLTHK to 1 or 2 on *CONTROL_CONTACT, or set THKOPT to 1 and SHLTHK to 1 or 2 on Optional Card B to activate consideration of shell thickness offsets. SHLTHK = 1 excludes rigid bodies while SHLTHK = 2 does not.

Thickness changes in the contact are accounted for *if and only if* the shell thickness change option is flagged in the input (see ISTD on *CONTROL_SHELL). Single surface contacts additionally require THKCHG be set to 1 on *CONTROL_CONTACT to consider thickness change. Each cycle, LS-DYNA stores the nodal thicknesses while processing the shell elements for use in the contact algorithms. The interface stiffness may change with thickness depending on the input options used.

30.8 Initial Interpenetrations

The need to offset contact surfaces to account for the thickness of the shell elements contributes to initial interpenetrations. These interpenetrations can lead to severe numerical problems when execution begins, so they should be corrected if LS-DYNA is to run successfully. Often an early growth of negative contact energy is one sign that initial interpenetrations exist. Currently, LS-DYNA prints warning messages to the terminal, the d3hsp file, and the messag file to report interpenetrations of nodes through contact segments and the modifications to the geometry made by LS-DYNA to eliminate the interpenetrations. Sometimes such corrections simply move the problem elsewhere since it is possible that the physical location of the shell midsurface and possibly the shell thickness are incorrect. The single surface contact algorithms remove any nodes still interpenetrating on the second time step from the contact and print a warning message.

For the automatic contacts, the IGNORE flag specifies the treatment of initial interpenetrations. The flag can be set on Card 4 of *CONTROL_CONTACT, MPP1 of *CONTACT_OPTION, and Optional Card C of *CONTACT_OPTION. By default, LS-DYNA moves the nodes to remove initial interpenetrations. Changing the value of IGNORE causes LS-DYNA to track the initial interpenetrations but not eliminate them. With tracking, LS-DYNA applies a contact force if the penetration becomes larger than the initial interpenetration with a penetration depth equal to the difference between the two penetrations. If the penetration becomes less than the initial interpenetration, the contact algorithms track this smaller penetration. Thus, LS-DYNA ignores the initial interpenetrations.

For certain geometries, LS-DYNA cannot detect interpenetrations since the tracked node interpenetrates completely through the surface at the beginning of the calculation.

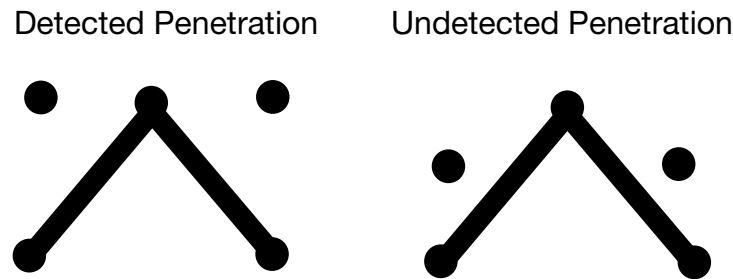


Figure 30.6. Undetected interpenetration. Such interpenetrations are frequently due to the use of coarse meshes.

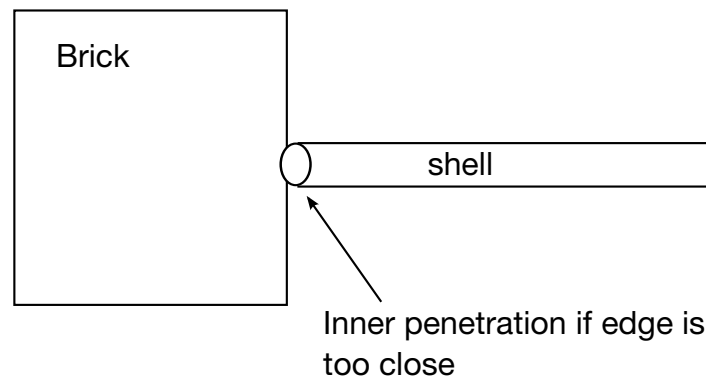


Figure 30.7. Undetected interpenetration due to rounding the edge of the shell element.

Figure 30.6 illustrates this type of undetectable penetration. Another case contributing to initial undetected interpenetrations occurs when the edge of a shell element is on the surface of a solid material as seen in Figure 30.7. Currently, shell edges are rounded with a radius equal to one-half the shell thickness. To avoid problems with initial interpenetrations, consider the following recommendations:

- Adequately offset adjacent surfaces to account for part thickness during the mesh generation phase.
- Use consistently refined meshes on adjacent parts which have significant curvatures.
- Be very careful when defining thickness on shell and beam section definitions -- especially for rigid bodies.
- Scale back part thickness if necessary. Scaling a 1.5 mm thickness to .75 mm should not cause problems but scaling to .075 mm might. Alternatively, define a smaller contact thickness by part ID. Warning: if the part is too thin, contact failure will probably occur.
- Use spot welds instead of merged nodes to allow the shell midsurfaces to be offset.

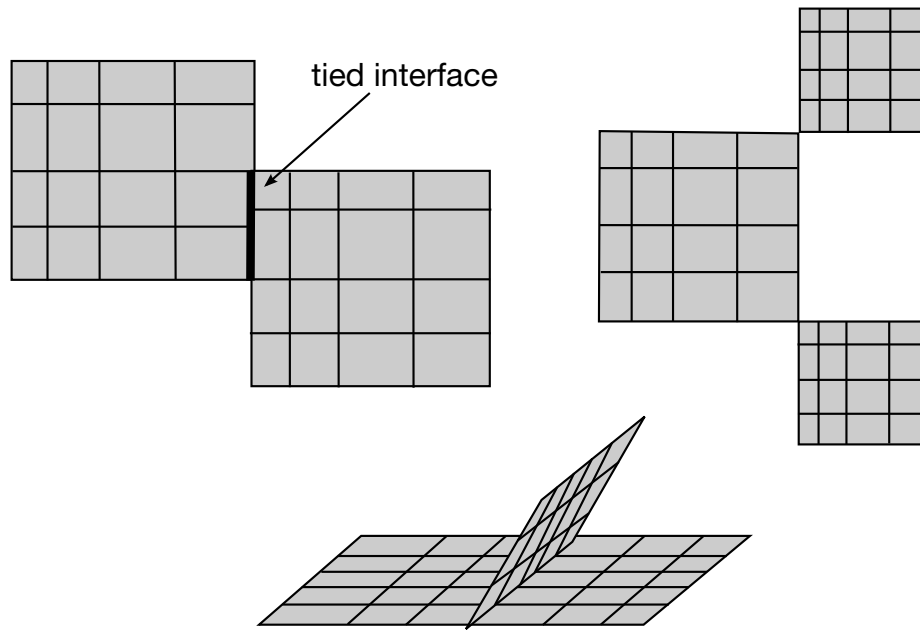


Figure 30.8. Incremental searching may fail on surfaces that are not simply connected. More recent contact algorithms in LS-DYNA avoid incremental searching for nodal points that are not in contact, so all these cases are considered.

30.9 Finding Penetration by Tracked Nodes and Segments

The original contact algorithms in LS-DYNA only performed an incremental tracked node search. This search finds for each tracked node its nearest point on the reference surface. However this type of search fails when the tracked or reference surfaces are disjoint. [Figure 30.8](#) illustrates such cases. Many models extensively use tied interfaces creating what appears to the contact algorithms to be topologically disjoint regions. All of our algorithms other than a few older ones use bucket sorting algorithms to identify the contact pairs and incremental searching to track the position of the pairs between bucket sorts. Note that non-automatic surface-to-surface including one way (*CONTACT_SURFACE_TO_SURFACE / *CONTACT_ONE_WAY_SURFACE_TO_SURFACE) and non-automatic nodes-to-surface (*CONTACT_NODES_TO_SURFACE) do not by default use bucket sorts if not accounting for shell thickness. They only use incremental searching. Enabling accounting for shell thickness with SHLTHK = 1 or 2 causes these contact types to use bucket sorts with incremental searching like the other contacts.

30.9.1 Overview of Bucket Sorting

In a direct search of a set of N nodes to determine the nearest node, the number of distance comparisons required is $N - 1$. Since this comparison needs to be made for each node, the total number of comparisons is $N(N - 1)$, with each of these comparisons requiring a distance calculation

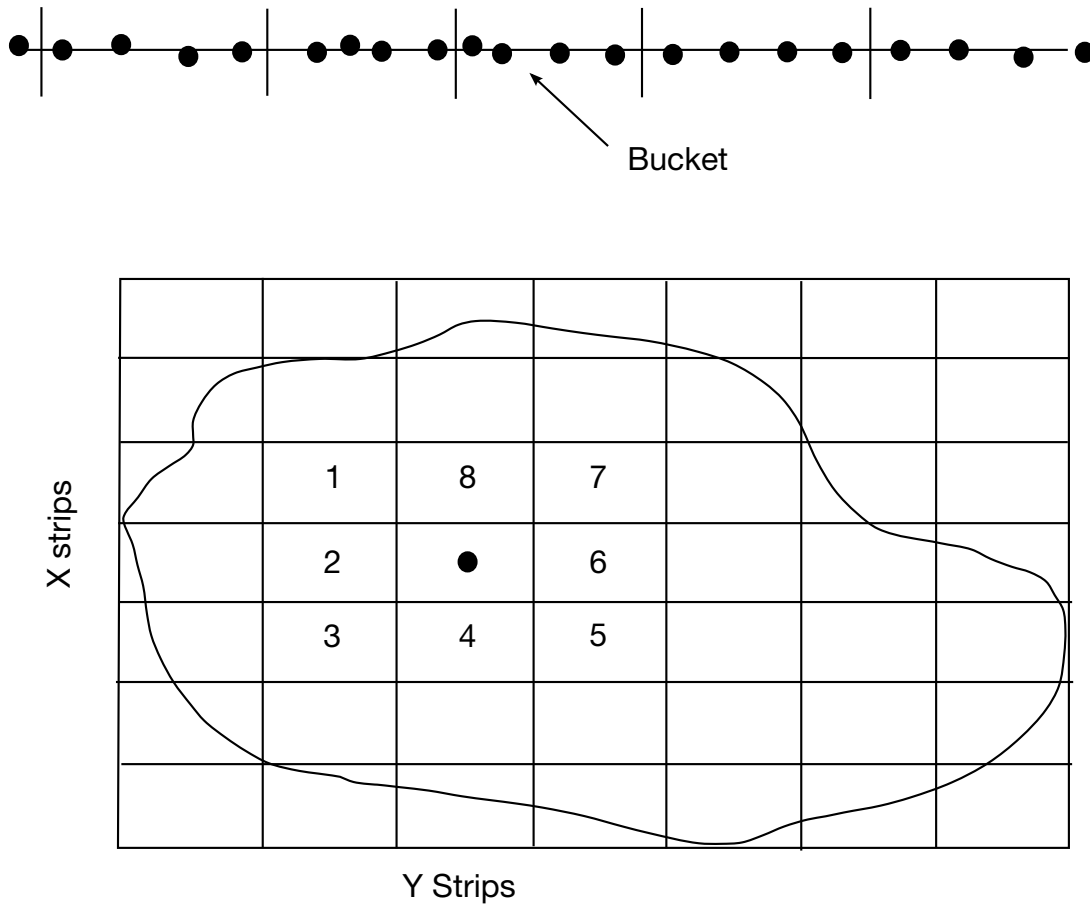


Figure 30.9. One- and two-dimensional bucket sorting.

$$l^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2, \quad (30.12)$$

that uses eight mathematical operations. The cumulative effect of these mathematical operations for $N(N - 1)$ compares can dominate the solution cost at less than 100 elements.

The idea behind a bucket sort is to perform some grouping of the nodes so that the sort operation need only calculate the distance of the nodes in the nearest groups. Consider the partitioning of the one-dimensional domain shown in Figure 30.9. With this partitioning, the nearest node will either reside in the same bucket or in one of the two adjoining buckets. The number of distance calculations is now given by

$$\frac{3N}{a} - 1, \quad (30.13)$$

where a is the number of buckets. The total number of distance comparisons for the entire one-dimensional surface is

$$N \left(\frac{3N}{a} - 1 \right). \quad (30.14)$$

Thus, if the number of buckets is greater than 3, the bucket sort requires fewer distance comparisons than a direct sort. It is easy to show that the corresponding number

of distance comparisons for two-dimensional and three-dimensional bucket sorts are given by

$$N \left(\frac{9N}{ab} - 1 \right) \text{ for 2D} \quad (30.15)$$

$$N \left(\frac{27N}{abc} - 1 \right) \text{ for 3D} \quad (30.16)$$

where b and c are the number of partitions along the additional dimension.

The cost of the grouping operations, needed to form the buckets, is nearly linear with the number of nodes, N . For typical LS-DYNA applications, the bucket sort is 100 to 1000 times faster than the corresponding direct sort. However, the sort is still an expensive part of the contact algorithms. To further minimize this cost, the contact algorithms perform the sort periodically and store the nearest three nodes. Typically, bucket sorting absorbs three to five percent of the calculational costs when the contact definition includes most surface segments.

For every contact in SMP and all segment-to-segment contacts in MPP and SMP, BSORT on Optional Card A sets the sorting frequency. Depending on the contact type, the default bucket sort interval with BSORT is between 10 and 100 cycles. BCKT on MPP1 sets the sorting frequency in MPP for all contacts except segment-to-segment contacts. BCKT has a default value of 200. Except for high speed impact, the default interval is almost always adequate. If nodes move from one disconnected surface to another in short time intervals or if the surface is folding onto itself, increase the contact bucket searching frequency, meaning reduce BSORT or increase BCKT. If two relatively smooth simply-connected surfaces are moving across each other without folds, the bucket sorting can be performed at larger intervals. Note that if the surfaces are more than several segment widths away from each other, no information is stored related to future contact, and later bucket searching is required to pick up future contacts. Once a tracked node is in contact, incremental searching tracks the motion, and bucket sorting for the nodes, which are in contact, is not necessary.

30.9.2 Bucket Sorting in Non-Automatic Single Surface Contact (*CONTACT_SINGLE_SURFACE)

We set the number of buckets in the x , y , and z coordinate directions to NX , NY , and NZ , respectively. Letting $LMAX$ represent the longest characteristic length (found by checking the length of the segment diagonals and taking a fraction thereof) over all segments in the contact definition, the following equations give the number of buckets in each direction

$$NX = \frac{x_{\max} - x_{\min}}{LMAX}, \quad (30.17)$$

$$NY = \frac{y_{\max} - y_{\min}}{LMAX}, \quad (30.18)$$

$$NZ = \frac{z_{\max} - z_{\min}}{LMAX}, \quad (30.19)$$

where the coordinate pairs (x_{\min}, x_{\max}) , (y_{\min}, y_{\max}) , and (z_{\min}, z_{\max}) define the extent of the contact surface and are updated each time bucket searching is performed. In order to dynamically allocate memory effectively with Fortran, we further restrict the number of buckets such that the total number of buckets does not exceed the number of nodes in the contact surface, NTN, or an internally determined maximum size, MAXSIZE:

$$NX \times NY \times NZ \leq \min(\text{NTN}, \text{MAXSIZE}) . \quad (30.20)$$

If the characteristic length, LMAX, is large due to an oversized contact segment or an instability leading to a node flying off into space, bucket sorting can slow down considerably because of the reduced number of buckets.

We use the formulas given by Belytschko and Lin [1985] to find the bucket containing a node with coordinates (x, y, z) . The bucket pointers are given by

$$PX = NX \times \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} + 1, \quad (30.21)$$

$$PY = NY \times \frac{(y - y_{\min})}{(y_{\max} - y_{\min})} + 1, \quad (30.22)$$

$$PZ = NZ \times \frac{(z - z_{\min})}{(z_{\max} - z_{\min})} + 1, \quad (30.23)$$

They are used to compute the bucket number given by

$$NB = PX + (PY - 1) \times PX + (PZ - 1) \times PX \times PY. \quad (30.24)$$

For each nodal point, k , in the contact surface we locate the three nearest neighboring nodes by searching all nodes in buckets from

$$\text{MAX}(1, PX1), \text{MIN}(NX, PX + 1), \quad (30.25)$$

$$\text{MAX}(1, PY1), \text{MIN}(NY, PY + 1), \quad (30.26)$$

$$\text{MAX}(1, PZ1), \text{MIN}(NZ, PZ + 1). \quad (30.27)$$

The algorithm searches a maximum of twenty-seven buckets. This nodal search does not consider nodes that share a contact segment with k . By storing the three nearest nodes and rechecking these stored nodes every cycle to see if the nearest node changes, we avoid performing bucket sorting every cycle. Typically, sorting every five to fifteen cycles is adequate. Implicit in this approach is the assumption that a node contacts just one surface. For this reason the non-automatic single surface contact (*CONTACT_SINGLE_SURFACE_... in LS-DYNA) does not apply to all problems. For example, in metal forming applications contact often occurs on both surfaces of the workpiece.

We define the nearest contact segment to a given node, k , as the first segment encountered when moving in a direction normal to the surface away from k . [Figure 30.10](#) depicts a major deficiency with the nearest node search in which the nearest nodes are not even members of the nearest contact segment. Obviously, a more uniform mesh does

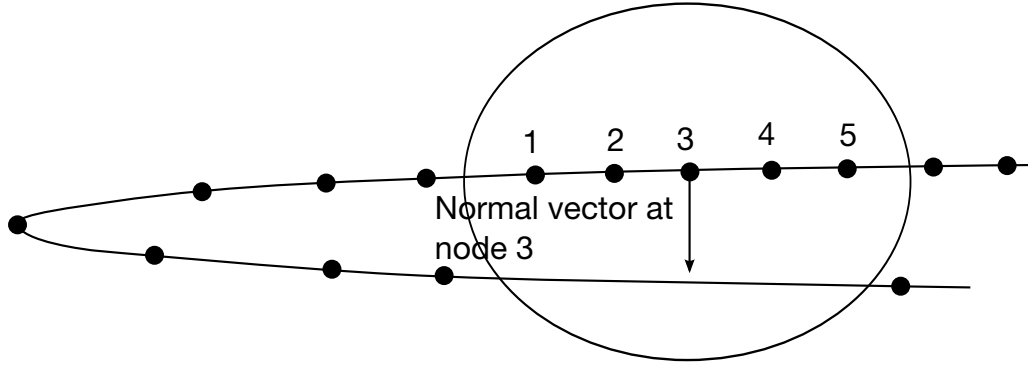


Figure 30.10. Nodes 2 and 4 share segments with node 3 and therefore the two nearest nodes are 1 and 5. The nearest contact segment is not considered since its nodes are not members of the nearest node set.

not have this problem. To overcome this problem, we have adopted segment-based searching in both surface-to-surface and automatic single surface contact as discussed in the next section.

30.9.3 Bucket Sorting in Surface-to-Surface and Automatic Single Surface Contact (*CONTACT_AUTOMATIC_SINGLE_SURFACE)

The procedure is roughly the same as discussed in the previous section, except we no longer base the bucket size on LMAX, which can result in as few as one bucket being generated. Rather, the product of the number of buckets in each direction always approaches NTN or MAXSIZE whichever is smaller,

$$NX \times NY \times NZ \leq \min(\text{NTN}, \text{MAXSIZE}), \quad (30.28)$$

where the coordinate pairs (x_{\min}, x_{\max}) , (y_{\min}, y_{\max}) , and (z_{\min}, z_{\max}) span the entire contact surface. In this newer procedure we loop over the segments rather than the nodal points. For each segment we use a nested DO LOOP to loop through a subset of buckets from IMIN to IMAX, JMIN to JMAX, and KMIN to KMAX where

$$\text{IMIN} = \text{MIN}(\text{PX1}, \text{PX2}, \text{PX3}, \text{PX4}), \quad (30.29)$$

$$\text{IMAX} = \text{MAX}(\text{PX1}, \text{PX2}, \text{PX3}, \text{PX4}), \quad (30.30)$$

$$\text{JMIN} = \text{MIN}(\text{PY1}, \text{PY2}, \text{PY3}, \text{PY4}), \quad (30.31)$$

$$\text{JMAX} = \text{MAX}(\text{PY1}, \text{PY2}, \text{PY3}, \text{PY4}), \quad (30.32)$$

$$\text{KMIN} = \text{MIN}(\text{PZ1}, \text{PZ2}, \text{PZ3}, \text{PZ4}), \quad (30.33)$$

$$\text{KMAX} = \text{MAX}(\text{PZ1}, \text{PZ2}, \text{PZ3}, \text{PZ4}), \quad (30.34)$$

and $\text{PX}k$, $\text{PY}k$, $\text{PZ}k$ are the bucket pointers for the k^{th} node. Figure 30.11 shows a segment passing through a volume that has been partitioned into buckets.

We check the orthogonal distance of all nodes in the bucket subset from the segment. During processing of each segment, the algorithm determines the minimum distance to a segment for every node in the surface and stores the two nearest segments.

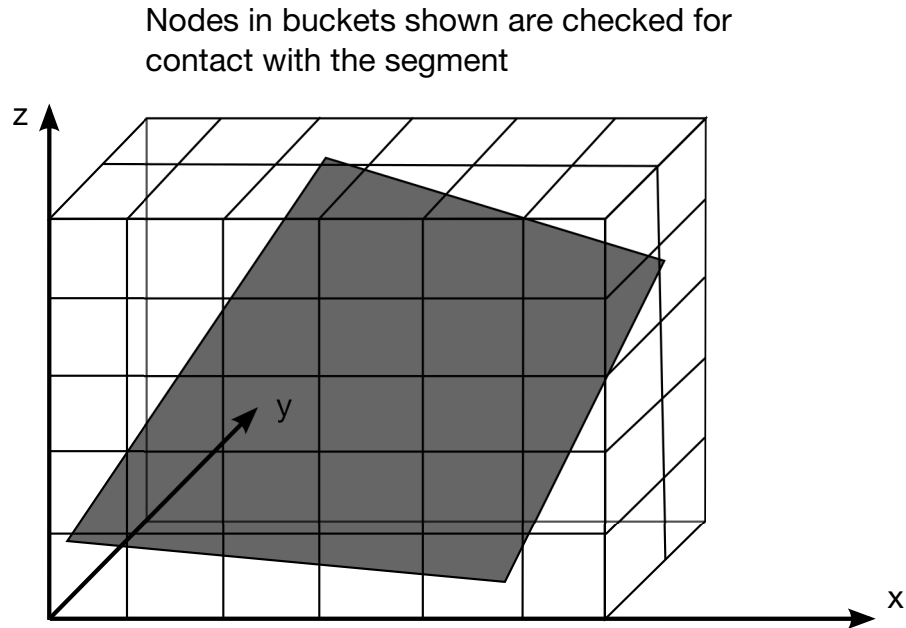


Figure 30.11. The algorithm determines the orthogonal distance of each tracked node contained in the box from the segment. This box is subdivided into sixty buckets.

Therefore the required storage allocation is still deterministic. This would not be the case if we stored for each segment a list of nodes that could possibly contact the segment.

We have now determined for each node, k , in the contact surface the two nearest segments for contact. Having located these segments we permanently store the node on these segments which is nearest to node k . When checking for interpenetrating nodes we check the segments surrounding the node including the nearest segment since during the steps between bucket searches it is likely that the nearest segment may change. It is possible to bypass nodes that are already in contact and save some compute time. However, if multiple contacts per node are admissible, bypassing the search may lead to unacceptable errors.

30.9.4 Incremental Tracked Node Search

All contact algorithms other than segment-to-segment type contacts in LS-DYNA use the incremental tracked node search. For most contacts this search occurs between bucket sorts as discussed previously. This search finds for each tracked node its nearest point on the reference surface. Lines drawn from a tracked node to its nearest point are perpendicular to the reference surface, unless the point lies along the intersection of two reference segments, where a segment is defined to be a 3- or 4-node element of a surface.

Consider a tracked node, n_t , sliding on a piecewise smooth reference surface and assume that a search of the reference surface has located the reference node, r_t , lying nearest to n_t . Figure 30.12 depicts a portion of a reference surface with nodes r_t and n_t labeled. If r_t and n_t do not coincide, n_t can usually be shown to lie in a segment s_1 via the following tests:

(30.35)

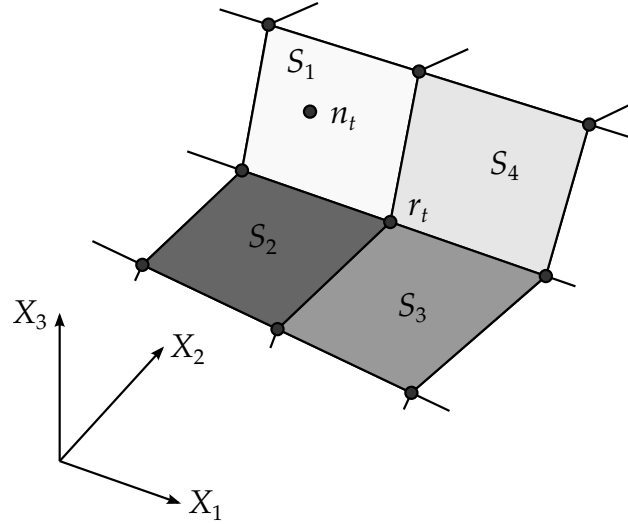


Figure 30.12. In this figure, four reference segments can harbor tracked node n_t given that r_t is the nearest reference node.

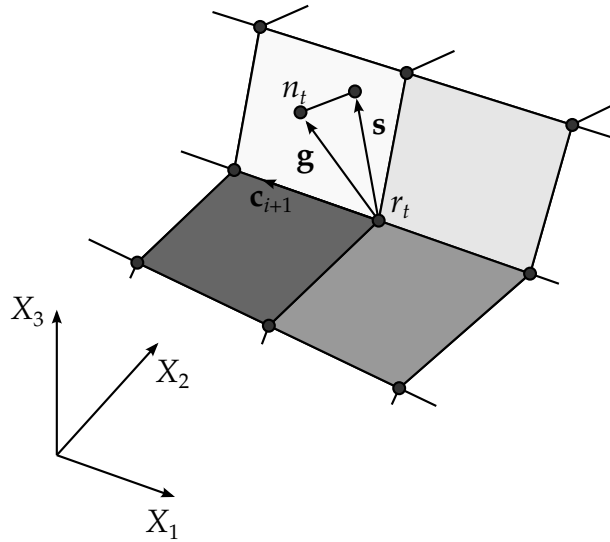


Figure 30.13. Projection of \mathbf{g} onto reference segment s_1

$$(\mathbf{c}_i \times \mathbf{s}) \cdot (\mathbf{s} \times \mathbf{c}_{i+1}) > 0,$$

where vector \mathbf{c}_i and \mathbf{c}_{i+1} are along edges of s_1 and point outward from r_t . Vector \mathbf{s} is the projection of the vector beginning at r_t and ending at n_t (denoted by \mathbf{g}) onto the plane being examined (see [Figure 30.13](#)).

$$\mathbf{s} = \mathbf{g} - (\mathbf{g} \cdot \mathbf{m})\mathbf{m}, \quad (30.36)$$

where for segment s_1

$$\mathbf{m} = \frac{\mathbf{c}_i \times \mathbf{c}_{i+1}}{|\mathbf{c}_i \times \mathbf{c}_{i+1}|}. \quad (30.37)$$

Since the sliding constraints keep n_t close but not necessarily on the reference surface and since n_t may lie near or even on the intersection of two reference segments, the

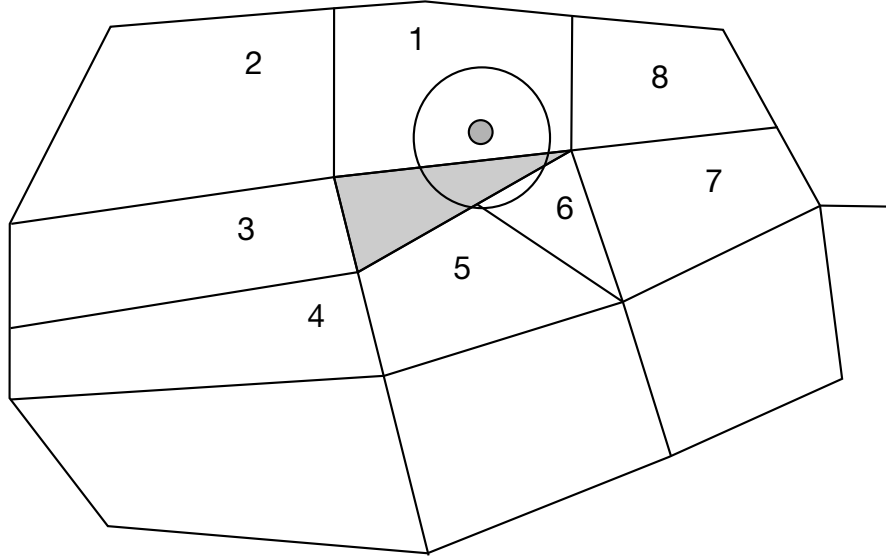


Figure 30.14. When the nearest node fails to contain the segment that harbors the tracked node, segments numbered 1-8 are searched in the order shown.

inequalities of Equation (30.35) may be inconclusive, meaning they may fail to be satisfied or more than one may give positive results. When this occurs n_t is assumed to lie along the intersection which yields the maximum value for the quantity

$$\frac{\mathbf{g} \cdot \mathbf{c}_i}{|\mathbf{c}_i|} \quad i = 1, 2, 3, 4, \dots \quad (30.38)$$

Furthermore, when the contact surface is made up of badly shaped elements, the segment apparently identified as containing the tracked node actually may not, as shown in Figure 30.14.

Assume that a reference segment has been located for tracked node n_t and that n_t is not identified as lying on the intersection of two reference segments. Then, the identification of the contact point, defined as the point on the reference segment which is nearest to n_t , becomes nontrivial. For each reference surface segment, s_1 is given the parametric representation of Equation (27.2), repeated here for clarity:

$$\mathbf{r} = f_1(\xi, \eta)\mathbf{i}_1 + f_2(\xi, \eta)\mathbf{i}_2 + f_3(\xi, \eta)\mathbf{i}_3, \quad (30.39)$$

where

$$f_i(\xi, \eta) = \sum_{j=1}^4 \phi_j x_i^j. \quad (30.40)$$

Note that \mathbf{r} is at least once continuously differentiable and that

$$\frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta} \neq 0, \quad (30.41)$$

Thus, \mathbf{r} represents a reference segment that has a unique normal whose direction depends continuously on the points of s_1 .

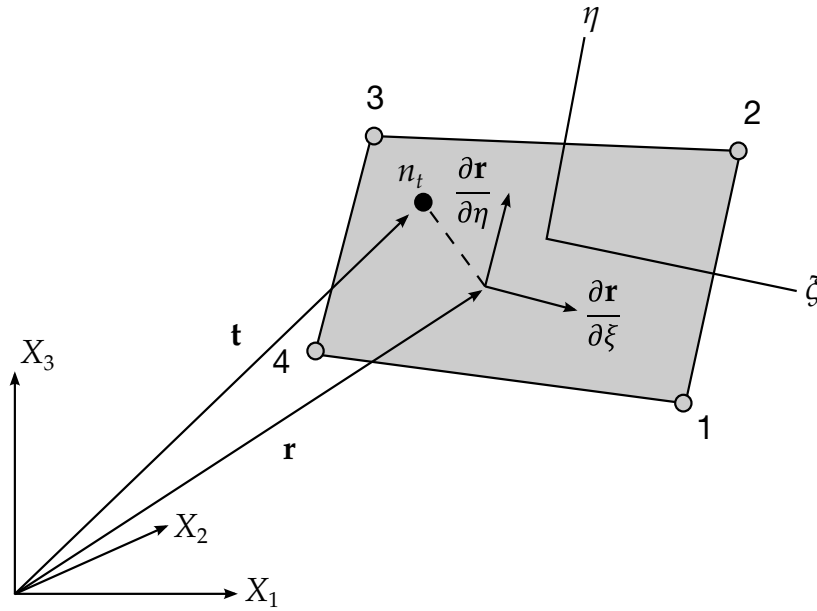


Figure 30.15. Location of contact point when n_t lies above the reference segment.

Let \mathbf{t} be a position vector drawn to tracked node n_t and assume that the reference surface segment s_1 has been identified with n_t . The contact point coordinates (ξ_c, η_c) on s_1 must satisfy

$$\frac{\partial \mathbf{r}}{\partial \xi}(\xi_c, \eta_c) \cdot [\mathbf{t} - \mathbf{r}(\xi_c, \eta_c)] = 0, \quad (30.42)$$

$$\frac{\partial \mathbf{r}}{\partial \eta}(\xi_c, \eta_c) \cdot [\mathbf{t} - \mathbf{r}(\xi_c, \eta_c)] = 0. \quad (30.43)$$

The physical problem is illustrated in [Figure 30.15](#), which shows n_t lying above the reference surface. [Equations \(30.42\) and \(30.43\)](#) are readily solved for ξ_c and η_c . One way to accomplish this is to solve [Equation \(30.42\)](#) for ξ_c in terms of η_c , and substitute the results into [Equation \(30.43\)](#). This yields a cubic equation in η_c .

LS-DYNA solves [Equations \(30.42\) and \(30.43\)](#) numerically. When two nodes of a bilinear quadrilateral are collapsed into a single node for a triangle, the Jacobian of the minimization problem is singular at the collapsed node. Fortunately, there is an analytical solution for triangular segments since three points define a plane. Newton-Raphson iteration is a natural choice for solving these simple nonlinear equations. The method diverges with distorted elements unless the initial guess is accurate. An exact contact point calculation is critical in post-buckling calculations to prevent the solution from wandering away from the desired buckling mode.

Three iterations with a least-squares projection are used to generate an initial guess:

$$(30.44)$$

$$\tilde{\zeta}_{i+1} = \tilde{\zeta}_i + \Delta\tilde{\zeta}, \quad \eta_{i+1} = \eta_i + \Delta\eta,$$

followed by the Newton-Raphson iterations which are limited to ten iterations, but which usually converges in four or less.

$$\begin{aligned} [H] \begin{Bmatrix} \Delta\tilde{\zeta} \\ \Delta\eta \end{Bmatrix} &= - \begin{Bmatrix} \mathbf{r}_{,\tilde{\zeta}} \\ \mathbf{r}_{,\eta} \end{Bmatrix} \{ \mathbf{r}(\tilde{\zeta}_i, \eta_i) - \mathbf{t} \}, \\ [H] &= \begin{Bmatrix} \mathbf{r}_{,\tilde{\zeta}} \\ \mathbf{r}_{,\eta} \end{Bmatrix} [\mathbf{r}_{,\tilde{\zeta}} \quad \mathbf{r}_{,\eta}] + \begin{bmatrix} 0 & \mathbf{r} \cdot \mathbf{r}_{,\tilde{\zeta}\eta} \\ \mathbf{r} \cdot \mathbf{r}_{,\tilde{\zeta}\eta} & 0 \end{bmatrix}, \\ \tilde{\zeta}_{i+1} &= \tilde{\zeta}_i + \Delta\tilde{\zeta}, \quad \eta_{i+1} = \eta_i + \Delta\eta, \end{aligned} \quad (30.45)$$

In concave regions, a tracked node may have isoparametric coordinates that lie outside of the $[-1, +1]$ range for all reference segments, yet still have penetrated the surface. A simple strategy is used for handling this case, but it can fail. The contact segment for each node is saved every time step. If the tracked node contact point defined in terms of the isoparametric coordinates of the segment, is just outside of the segment, and the node penetrated the isoparametric surface, and no other segment associated with the nearest neighbor satisfies the inequality test, then the contact point is assumed to occur on the edge of the segment. In effect, the definition of the reference segments is extended so that they overlap by a small amount. In the hydrocode literature, this approach is similar to the slide line extensions used in two dimensions. This simple procedure works well for most cases, but it can fail in situations involving sharp concave corners.

30.10 Contact Energy Calculation

Contact energy, E_{contact} , is incrementally updated from time n to time $n + 1$ for each contact interface as:

$$E_{\text{contact}}^{n+1} = E_{\text{contact}}^n + \left[\sum_{i=1}^{\text{ntn}} \Delta F_i^{\text{tracked}} \times \Delta \text{dist}_i^{\text{tracked}} + \sum_{i=1}^{\text{nrn}} \Delta F_i^{\text{reference}} \times \Delta \text{dist}_i^{\text{reference}} \right]^{n+\frac{1}{2}}, \quad (30.46)$$

where ntn is the number of tracked nodes, nrn is the number of reference nodes, $\Delta F_i^{\text{tracked}}$ is the interface force between the i^{th} tracked node and the contact segment, $\Delta F_i^{\text{reference}}$ is the interface force between the i^{th} reference node and the contact segment, $\Delta \text{dist}_i^{\text{tracked}}$ is the incremental distance the i^{th} tracked node has moved during the current time step, and $\Delta \text{dist}_i^{\text{reference}}$ is the incremental distance the i^{th} reference node has moved during the current time step. In the absence of friction, the tracked and reference side energies should be close in magnitude but opposite in sign. The sum, E_{contact} , should equal the stored energy. Large negative contact energy is usually caused by undetected penetrations. LS-DYNA reports contact energies in the SLEOUT file. In the presence of friction and damping discussed below the interface energy can take on a substantial positive value, especially when substantial sliding occurs in the presence of friction.

30.11 Contact Damping

Viscous contact damping has been added to all contacts. We originally intended it to damp out oscillations normal to the contact surfaces during metal forming operations. However, we later found it worked effectively in removing high frequency noise in problems that involve impact. The input requires a damping value as a percentage of critical, $2m$, where m is the mass and ω is the natural frequency. Letting k denote the interface stiffness, we compute the natural frequency for an interface tracked node from [Equation \(1.47\)](#).

$$\omega = \sqrt{\frac{k(m_{\text{tracked}} + m_{\text{reference}})}{m_{\text{tracked}}m_{\text{reference}}}} \quad m = \min\{m_{\text{tracked}}, m_{\text{reference}}\}. \quad (30.47)$$

The reference mass, $m_{\text{reference}}$, is interpolated from the reference nodes of the segment containing the reference node using the basis functions evaluated at the contact point of the tracked node.

Force oscillations often occur as curved surfaces undergo relative motion. In these cases contact damping eliminates the high frequency content in the contact reaction forces but is unable to damp the lower frequency oscillations caused by nodes moving from segment to segment when a large angle change between the segments exists. The hemispherical punch deep drawing shown in [Figure 30.16](#) gives an example of this phenomenon. The reaction forces with and without contact damping in [Figure 30.17](#) show only minor differences since the oscillations do not come from the dynamic effects of explicit integration. However, refining the mesh as shown in [Figure 30.18](#) to include more ele-

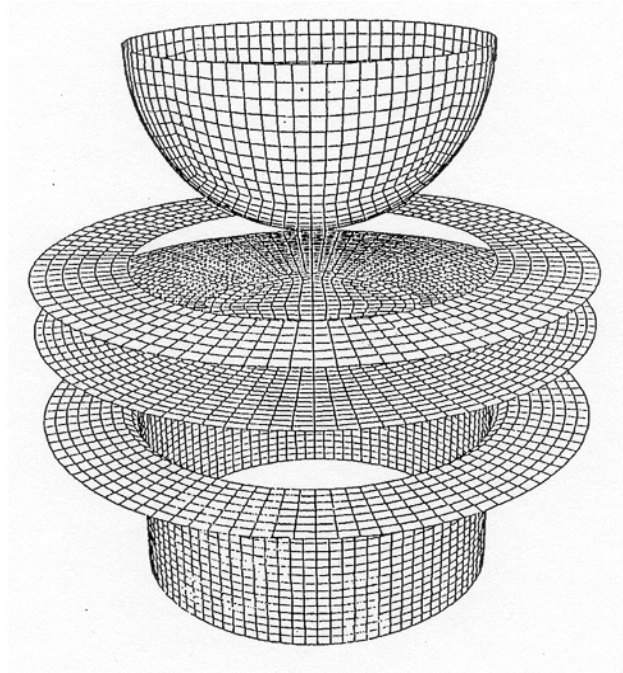


Figure 30.16. Hemispherical deep drawing problem.

ments around the die corner as in [Figure 30.18](#) greatly reduces the oscillations as shown

in Figure 30.19. This example indicates the importance of using an adequate mesh density in applications with significant relative motion around sharp corners.

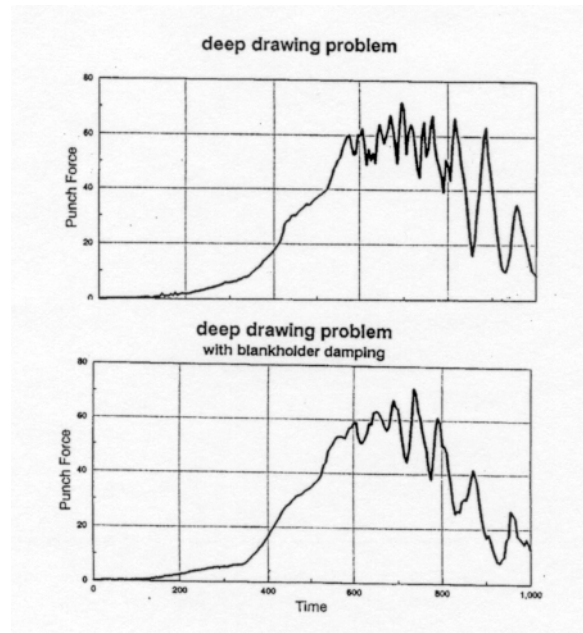


Figure 30.17. Reaction forces with and without contact damping.

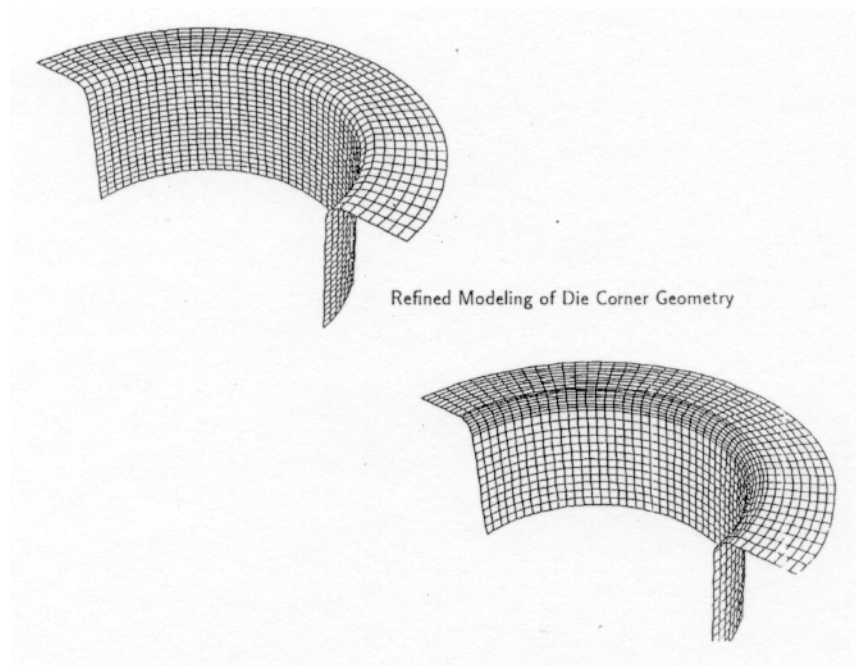


Figure 30.18. Refinement of die radius.

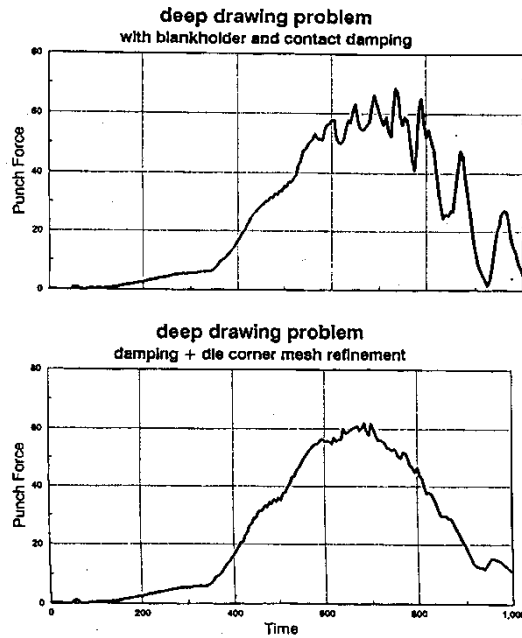


Figure 30.19. The oscillations are effectively eliminated by the mesh refinement.

30.12 Friction

We based friction modeling in LS-DYNA on a Coulomb formulation. Let \mathbf{f}^* be the trial force, \mathbf{f}_n the normal force, k the interface stiffness, μ the coefficient of friction, and \mathbf{f}^n the frictional force at time n . The frictional algorithm, outlined below, uses the equivalent of an elastic-plastic spring. The steps are as follows:

1. Compute the yield force, F_y :

$$F_y = \mu |\mathbf{f}_n| \quad (30.48)$$

2. Compute the incremental movement of the tracked node

$$\Delta \mathbf{e} = \mathbf{r}^{n+1}(\zeta_c^{n+1}, \eta_c^{n+1}) - \mathbf{r}^{n+1}(\zeta_c^n, \eta_c^n) \quad (30.49)$$

3. Update the interface force to a trial value:

$$\mathbf{f}^* = \mathbf{f}^n - k \Delta \mathbf{e} \quad (30.50)$$

4. Check the yield condition:

$$\mathbf{f}^{n+1} = \mathbf{f}^* \quad \text{if } |\mathbf{f}^*| \leq F_y \quad (30.51)$$

5. Scale the trial force if it is too large:

$$\mathbf{f}^{n+1} = \frac{F_y \mathbf{f}^*}{|\mathbf{f}^*|} \quad \text{if } |\mathbf{f}^*| > F_y \quad (30.52)$$

An exponential interpolation function smooths the transition between the static, μ_s , and dynamic, μ_d , coefficients of friction:

$$\mu = \mu_d + (\mu_s - \mu_d) e^{-c|\mathbf{v}|}, \quad (30.53)$$

Here, \mathbf{v} is the relative velocity between the tracked node and the reference segment:

$$\mathbf{v} = \frac{\Delta \mathbf{e}}{\Delta t}, \quad (30.54)$$

Δt is the time step size, and c is a decay constant.

The interface shear stress that develops as a result of Coulomb friction can be very large and in some cases may exceed the ability of the material to carry such a stress. We, therefore, allow another limit to be placed on the value of the tangential force:

$$f^{n+1} = \min(f_{\text{Coulomb}}^{n+1}, \kappa A_{\text{reference}}), \quad (30.55)$$

where $A_{\text{reference}}$ is the area of the reference segment and κ is the viscous coefficient. Since more than one node may contribute to the shear stress of a segment, we recognize that the stress may still in some cases exceed the limit κ .

Marks Engineering Handbook contains typical values of friction. See [Table 1.1](#).

Table 30.1. Typical values of Coulomb friction [Marks]

MATERIALS	STATIC	SLIDING
Hard steel on hard steel	0.78 (dry)	08 (greasy), .42 (dry)
Mild steel on mild steel	0.74 (dry)	10 (greasy), .57 (dry)
Aluminum on mild steel	0.61 (dry)	47 (dry)
Aluminum on aluminum	05 (dry)	1.4 (dry)
Tires on pavement (40psi)	0.90 (dry)	69(wet), .85(dry)

30.13 Tied Interfaces

Tied interfaces permit sudden transitions in zoning. For instance, [Figure 30.20](#) shows the joining of two meshes of solid elements. This feature can often decrease the amount of effort required to generate meshes since it reduces the need to match nodes across interfaces of merged parts. With this feature, LS-DYNA ties tracked nodes within a certain tolerance of the reference surface to the reference segments. For most tied interfaces, tying only occurs at the beginning of the simulation. With offsets enabled, the tied tracked node can be offset from the reference segment. Otherwise, the tied contact algorithms move the node to the surface during initialization.

Since the contact algorithms impose the constraints only on the tracks nodes, we recommend making the more coarsely meshed side of the interface the reference surface.

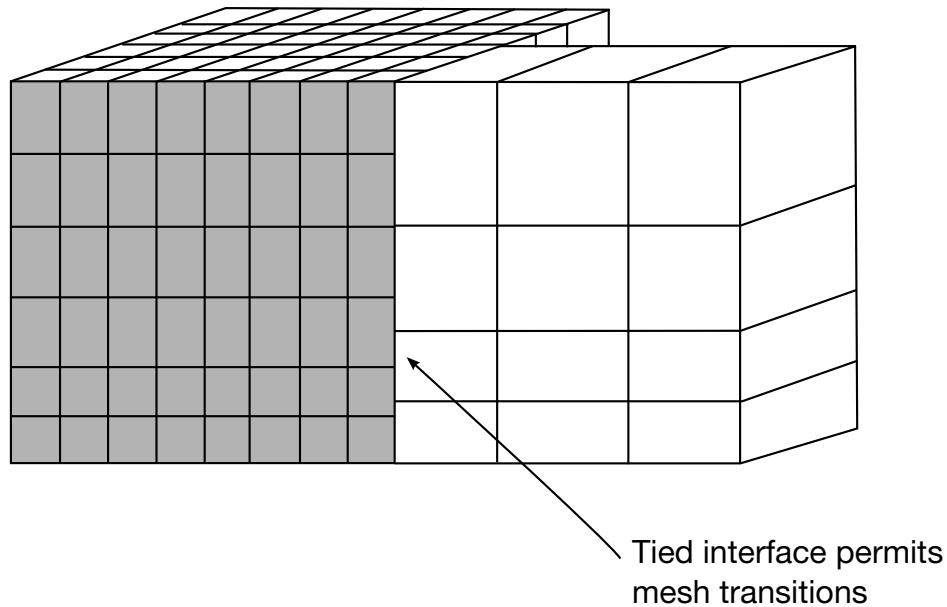


Figure 30.20. Tied interface used for a mesh transition.

Ideally, each reference node should coincide with a tracked node to ensure complete displacement compatibility along the interface, but in practice this is often difficult if not impossible to achieve. In other words, reference nodes that do not coincide with a tracked node can interpenetrate through the tracked surface.

A key distinction among the different types of tied contacts is whether they tie only translational degrees of freedom or tie both translational and rotational degrees of freedom. Tying only translational degrees of freedom applies to elements that only have translational degrees of freedom. Tying translational and rotational degrees of freedom applies to elements with rotational degrees of freedom like shells and beams.

Most of our tied contacts enforce the tie with the kinematic constraint method (see Section 30.2). With some of the contacts, we enabled offsets which switch the formulation from constraint-based to penalty-based (see Standard Penalty Formulation and Soft Constraint Penalty Formulation in Section 30.3). The penalty approach removes some major limitations of the constraint formulation:

- Multiple tied interfaces can share common nodes.
- Rigid body nodes can be constrained.
- Tied interface nodes can have other constraints applied and can be subjected to prescribed motions.

However, the penalty method does not enforce the constraint exactly.

The keyword options `OFFSET`, `BEAM_OFFSET`, and `CONSTRAINED_OFFSET` enable offsets. The `OFFSET` and `BEAM_OFFSET` options change the tied contact from a kinematic constraint formulation to a penalty formulation. Because they use penalty formulations, they can be used for tying to rigid bodies. We developed `OFFSET` prior to `BEAM_OFFSET`, so it suffers some drawbacks. `OFFSET` uses discrete springs to transmit the force and moments while `BEAM_OFFSET` uses beam-like springs. It does not couple

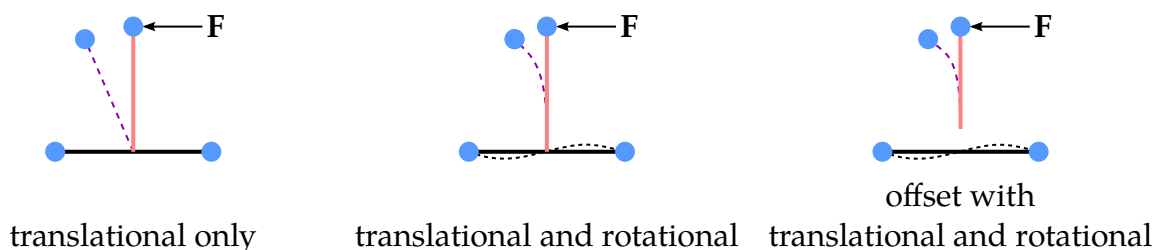


Figure 30.21. Broad classifications of tied contacts. The left schematic illustrates coupling only translational degrees of freedom. The middle schematic shows coupling both the translational and rotational degrees of freedom. The right schematic indicates the behavior for an offset tied contact with coupling both the translational and rotational degrees of freedom.

moments and forces and thus does not enforce equilibrium. Because of the type of spring model, OFFSET does not account for the moment associated with the offset distance and can lead to nonphysical rotational constraints. As an example, if we tie two offset shells (with no boundary conditions applied) and apply an in-plane force to one of the shells, we would expect the system to rotate. With OFFSET, no rotation occurs. With BEAM-OFFSET or CONSTRAINED_OFFSET, the system rotates as expected.

Some of the tied contact support failure. Most of these include TIEBREAK in the name. The TIEBREAK contacts specifically turn into sliding contacts after failure. They are also penalty-based, so they can be used with rigid bodies.

Table 1.2 below indicates the tied contact types and which of these features they support. This table does not include TIEBREAK contacts as they will be discussed in Section 30.13.3. The * in the FAILURE column of the table indicates that these contacts can include failure if using *MAT_SPOTWELD as the material. *MAT_SPOTWELD sets the failure criteria.

Table 30.2. Table indicating the different types of tied contacts. The * indicates contacts that support failure through *MAT_SPOTWELD.

Tied Contact Name	Translational	Rotational	Offsets	Failure	Penalty-Based
AUTOMATIC_SINGLE_SURFACE_TIED	Y				Y
AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_TIED_WELD	Y				Y
AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD	Y				Y
SPOTWELD	Y	Y		Y*	
SPOTWELD_BEAM_OFFSET	Y	Y	Y	Y*	Y
SPOTWELD_CONSTRAINED_OFFSET	Y	Y	Y	Y*	
SPOTWELD_WITH_TORSION	Y	Y		Y*	

Tied Contact Name	Translational	Rotational	Offsets	Failure	Penalty-Based
SPOTWELD_WITH_TORSION_PENALTY	Y	Y		Y*	Y
TIED_NODES_TO_SURFACE	Y				
TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET	Y		Y		
TIED_NODES_TO_SURFACE_OFFSET	Y		Y		Y
TIED_SHELL_EDGE_TO_SOLID	Y	Y			
TIED_SHELL_EDGE_TO_SURFACE	Y	Y		Y*	
TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET	Y	Y	Y	Y*	Y
TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET	Y	Y	Y	Y*	
TIED_SHELL_EDGE_TO_SURFACE_OFFSET	Y	Y	Y	Y*	Y
TIED_SURFACE_TO_SURFACE	Y				
TIED_SURFACE_TO_SURFACE_CONSTRAINED_OFFSET	Y		Y		
TIED_SURFACE_TO_SURFACE_FAILURE	Y			Y	
TIED_SURFACE_TO_SURFACE_OFFSET	Y		Y		Y

30.13.1 Outline of Available Tied Contacts

Here we will briefly outline the tied contacts listed in [Table 30.2](#).

*CONTACT_TIED_NODES_TO_SURFACE and *CONTACT_TIED_SURFACE_TO_SURFACE should only be used for tying solid elements to solid elements because they do not constrain rotational degrees of freedom. They may cause unrealistically soft behavior if used with shell elements. These two contacts only differ in input format for the tracked and reference surfaces. The numerical treatment is the same. They also can have the OFFSET keyword option. These contacts work best if the surfaces are close since this offset algorithm does not account for the moments associated with the offset distance between the two surfaces as previously discussed. *CONTACT_TIED_SURFACE_TO_SURFACE_FAILURE supports failure with failure stresses. Segment orientation is important to distinguish tension from compression for this contact.

For shell-to-shell tying and beam-to-shell tying, use *CONTACT_TIED_SHELL_EDGE_TO_SURFACE as it includes tying of rotational degrees of freedom. The tied interface allows tying of edges of shells to shell surfaces (see [Figure 30.22](#)). In such transitions the contact algorithm does not consider shell thickness. Note that for implicit, setting IACC = 1 enables using the offset versions of either *CONTACT_TIED_NODES_TO_SURFACE or *CONTACT_TIED_SHELL_EDGE_TO_SURFACE with any type of element. The contact algorithms apply the rotational constraints automatically as needed (see [Section 30.14](#)).

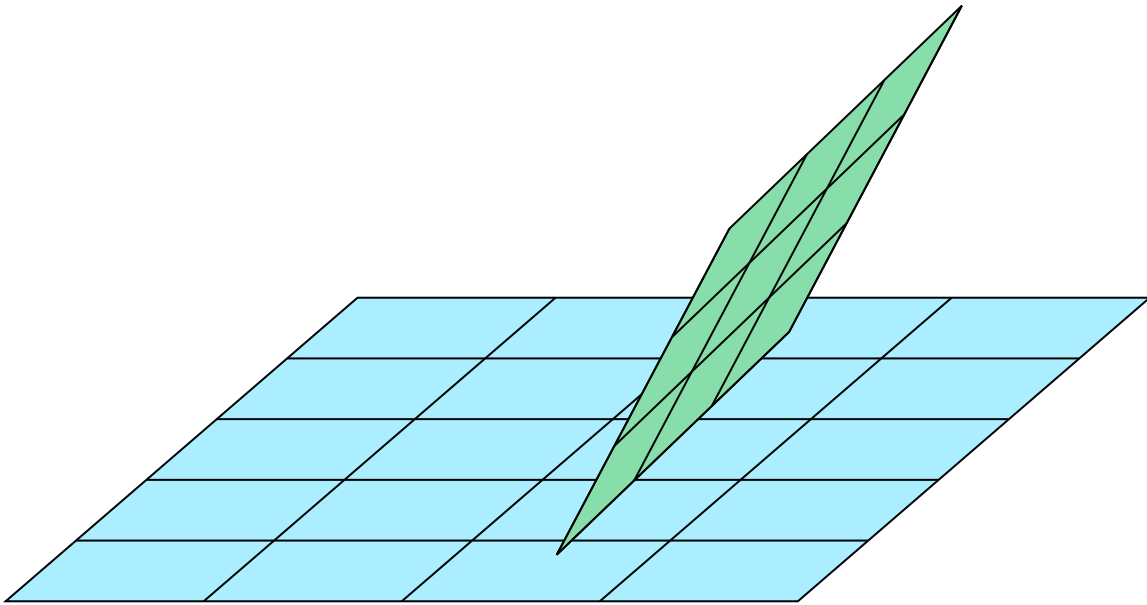


Figure 30.22. Shell-edge-to-surface tied contact

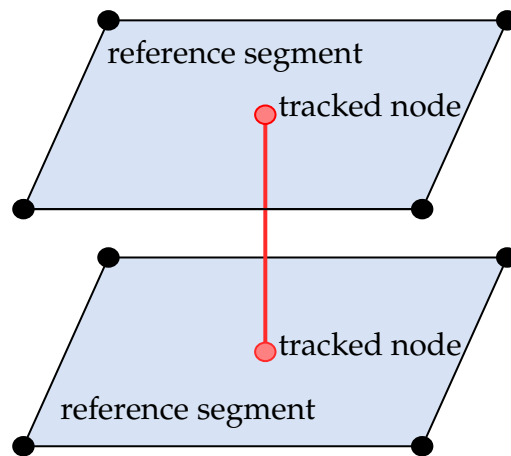


Figure 30.23. Illustration of modeling spot welds using beams

For modeling beam spot welds, use `*CONTACT_SPOTWELD` or `*CONTACT_SPOTWELD_WITH_TORSION` (see [Figure 30.23](#)). The beams are the tracked surface while the shell surfaces are the reference. With the `TORSION` option, the contact algorithm transmits the torsional forces in the beam as equivalent forces to the surrounding nodes of the reference surface. The rotational constraint about the axis of the beam is then enforced. The nonlinear shell elements in LS-DYNA have a zero stiffness drilling degree of freedom at each node, so it is necessary to carry the torsional forces through the membrane behavior of the shell.

We developed `*CONTACT_TIED_SHELL_EDGE_TO_SOLID` for tying shells or beams to solids. This tied contact transfers moments from shells to solids through forces.

`*CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED`, `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD`, and `*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR_TIED_WELD` are penalty-based tied contacts for

particular situations. We implemented *CONTACT_AUTOMATIC_SINGLE_SURFACE_TIED for calculating eigenvalues and eigenvectors when modeling the contacts with *CONTACT_AUTOMATIC_SINGLE_SURFACE. The TIED_WELD contacts simulate welding. Unlike the other tied contacts, tying occurs throughout the simulation when regions come into contact and have reached an appropriate temperature. Before reaching the weld temperature LS-DYNA treats the surfaces coming into contact using normal AUTOMATIC_SURFACE_TO_SURFACE(_MORTAR).

30.13.2 Implementation of Constraint-Based Tied Contacts

The implementation of tied interface constraints is straightforward. Each time step we loop through the tied interfaces and update each one independently. First, we distribute the nodal forces and nodal mass of each tracked node to the reference nodes, which define the segment containing the contact point. In other words, we add the increments in mass and forces

$$\Delta f_r^i = \phi_i(\xi_c, \eta_c) f_t^i \quad (30.56)$$

to the mass and force vector of the reference surface. After the summation over all tracked nodes is complete, we can compute the acceleration of the reference surface. The acceleration of each tracked node a_{i_t} is then interpolated from the reference segment containing its contact points:

$$a_{i_t} = \sum_{j=1}^4 \phi_j(\xi_c, \eta_c) a_r^j. \quad (30.57)$$

Velocities and displacements are now updated normally.

The interpolated contact point, (ξ_c, η_c) , for each tracked node is computed once since its relative position on the reference segment is constant for the duration of the calculation. If the closest point projection of the tracked node to the reference surface is non-orthogonal, values of (ξ_c, η_c) greater than unity will be computed. To allow for slight errors in the mesh definition, the tracked node is left unconstrained if the magnitude of the contact point exceeds 1.02. Great care should be exercised in setting up tied interfaces to ensure that the tracked nodes are covered by reference segments.

Conflicting constraints must be avoided. Care should be taken not to include nodes that are involved in a tied interfaces in another tied interface, in constraint sets such as nodal constraint sets, in linear constraint equations, and in spot welds. Furthermore, constraint-based tied interfaces between rigid and deformable bodies are not permitted. LS-DYNA checks for conflicting constraints on nodal points and if such conflicts are found, the calculation terminates with an error message identifying the conflict. Nodes in tied interfaces should not be included as tracked nodes in rigid wall definitions since interactions with stonewalls cause the constraints that were applied in the tied interface logic to be violated. We do not currently check for this latter condition in LS-DYNA.

Tied interfaces require coincident surfaces. For shell elements, this requirement means that the midsurfaces must be coincident. Consider [Figure 30.24](#) where identical

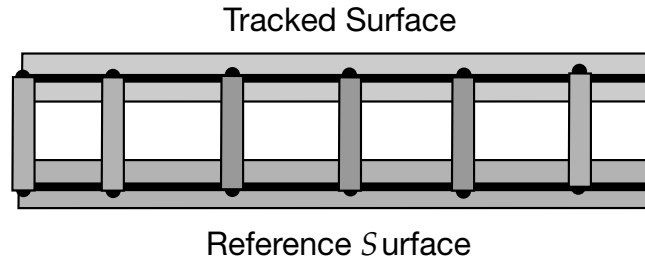


Figure 30.24. Offset tied interface.

tracked and reference surfaces are offset. In this case the tied constraints require that translational velocities of tied nodes be identical, meaning

$$\mathbf{v}^t = \mathbf{v}^r. \quad (30.58)$$

Consequently, if the nodes are offset, *rotations are not possible*. The velocity of a tied tracked node in Figure 30.24 should account for the segment rotation:

$$\mathbf{v}^t = \mathbf{v}^r - \hat{\mathbf{z}} \mathbf{e}_3 \times \boldsymbol{\omega}, \quad (30.59)$$

where $\hat{\mathbf{z}}$ is the distance to the tracked node, \mathbf{e}_3 is the normal vector to the reference surface at the contact point, and $\boldsymbol{\omega}$ is the angular velocity. Since this is not the case in the tied interfaces logic, $\hat{\mathbf{z}}$ must be of zero length.

LS-DYNA projects tied tracked nodes back to the reference surface if possible and prints warning messages for all projected offset nodes or nodes too far away to tie. This projection eliminates the problems with rotational constraints but creates other difficulties:

- Geometry is modified.
- Tied interfaces must be excluded from automatic generation since tied surfaces cannot be mixed with automatic contact with thickness offsets.

30.13.3 Tiebreak Contacts

Tiebreak contacts model connections that transmit both compressive and tensile forces with optional failure. For instance, they can model glue. These contacts are all penalty-based. Thus, tracked and reference nodes in a tiebreak contact can belong to other constraints and can belong to rigid bodies. No sliding is allowed between the elements used in the tiebreak definitions until failure. Upon failure, most of these contacts turn into regular sliding contacts. The failure criteria depends on the contact type and can be based on stress or force.

The older tiebreak contacts are non-automatic. These contacts are TIEBREAK_NODES_ONLY, TIEBREAK_NODES_TO_SURFACE, TIEBREAK_SURFACE_TO_SURFACE, and TIEBREAK_SURFACE_TO_SURFACE_ONLY. Because they are non-automatic, segment orientation is important for determining the tensile and compressive directions. They all only tie nodes at the beginning of the simulation. The TIEBREAK_NODES_ONLY and TIEBREAK_NODES_TO_SURFACE are the same contact until failure. They fail with following force-based criterion:

$$\left(\frac{|f_n|}{\text{NFLF}}\right)^{\text{NEN}} + \left(\frac{|f_s|}{\text{SFLF}}\right)^{\text{MES}} \geq 1 \quad (30.60)$$

where f_n and f_s are the normal (tensile) and shear forces. NFLF, SFLF, NEN, and MES are input parameters. After failure TIEBREAK_NODES_ONLY no longer acts as a contact, and TIEBREAK_NODES_TO_SURFACE acts a normal NODES_TO_SURFACE contact with no thickness offsets. Similarly, TIEBREAK_SURFACE_TO_SURFACE and TIEBREAK_SURFACE_TO_SURFACE_ONLY only behave differently upon failure. They fail with the following stress-based criterion:

$$\left(\frac{|\sigma_n|}{\text{NFLS}}\right)^2 + \left(\frac{|\sigma_s|}{\text{SFLS}}\right)^2 \geq 1 \quad (30.61)$$

where σ_n is the tensile, normal stress and σ_s is the shear stress. The compressive normal stress does not contribute to the failure. Again TIEBREAK_SURFACE_TO_SURFACE_ONLY no longer acts as a contact after failure. TIEBREAK_SURFACE_TO_SURFACE acts as a normal surface-to-surface contact with no thickness offsets by default. To enable thickness offsets, set THKOFF = 1 on Card 4 of the contact definition.

The options for automatic tiebreak are:

AUTOMATIC_GENERAL_TIEBREAK

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK_DAMPING

AUTOMATIC_SINGLE_SURFACE_TIEBREAK

AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK

AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_MORTAR

AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK_USER

AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_USER

AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK_USER_MORTAR

After failure, all of these contacts act as normal sliding contacts with thickness offsets. Since they are automatic, LS-DYNA automatically orients the shell/segment normals. They all fail with a stress-based failure model. The failure model for the non-USER versions depends on OPTION on Card 4. OPTION also determines when LS-DYNA ties tracked nodes to the reference surface. With some choices of OPTION, tying can occur throughout the simulation instead of at the beginning of the simulation. OPTION also determines the kinds of elements that can be tied and whether the contact transfers moments. Some OPTION choices also enable damage modeling. With the USER versions, you provide the failure model and damage model in a user-defined subroutine. For these models tying only occurs at the beginning of the simulation. Note that we generally

recommend the AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE versions over the AUTOMATIC_SURFACE_TO_SURFACE versions.

30.13.4 Alternatives to Tied Contacts

The *CONSTRAINED keyword family provides some alternatives to tied and tiebreak contacts for tying nodes to nodes or to surfaces. Some of them also include failure criteria.

*CONSTRAINED_SPOTWELD, *CONSTRAINED_GENERALIZED_WELD [*SPOT, FILLET, BUTT, CROSS_FILLET, COMBINED*], and *CONSTRAINED_BUTT_WELD provide methods of modeling welds. They all include failure models. *CONSTRAINED_SPOTWELD models the spot weld with massless, rigid beams.

*CONSTRAINED_NODAL_RIGID_BODY creates a rigid body from a set of nodes. *CONSTRAINED_EXTRA_NODES adds extra nodes to an existing rigid body. *CONSTRAINED_RIGID_BODIES merges two rigid bodies together.

*CONSTRAINED_TIE-BREAK ties shell edges to each other. It creates a tie using two sets of nodes to define the sides of the edges and includes a ductile, strain-based failure model. We intend it for modeling seams or structural joints. In certain cases it may be easier to use than *CONTACT_TIED_SHELL_EDGE_TO_SURFACE. *CONSTRAINED_TIED_NODES_FAILURE ties all the nodes in a set together. It has failure based on plastic strain. This model requires the nodes to be coincident. *CONSTRAINED_SHELL_TO_SOLID ties shell edges to solid elements. However, *CONTACT_TIED_SHELL_EDGE_TO_SOLID is easier to use than *CONSTRAINED_SHELL_TO_SOLID.

30.14 Strongly Objective Tied Contacts

In implicit calculations, nonphysical results have been observed when some tied contact formulations are combined with automatic single point constraints on solid element rotational degrees of freedom (AUTOSPC on *CONTROL_IMPLICIT_SOLVER). This issue has motivated us to add an optional strongly objective extension to some tied contacts. Strongly objective means the forces and moments transform correctly under superposed rigid body motions in a single implicit time step. The chosen contact types for this extension provide a universal tied contact formulation in implicit that works well in most situations. Thus, with the available types, you should not have to figure out which interface is best suited for the application. We selected the following contacts:

*CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

*CONTACT_TIED_NODES_TO_SURFACE_OFFSET

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

The first two contact types do not consider rotational degrees of freedom, whereas the other two do. Furthermore, the first and third are constraint-based and the other two are penalty-based. In all, these four contact types cover much of what is expected from a tied interface. By setting IACC to 1 on *CONTROL_ACCURACY, LS-DYNA treats any of the tied contact options mentioned above (and the non-offset counterparts as a side effect, namely, *CONTACT_TIED_NODES_TO_SURFACE and *CONTACT_TIED_SHELL_EDGE_TO_SURFACE) with this strongly objective formulation.

In addition to being strongly objective, this formulation applies rotational constraints consistently *when and only when necessary*. Thus, not only are tracked nodes without rotational degrees of freedom *not* rotationally constrained, but also bending and torsional rotations are constrained to the reference segment's rotational motion in a way that is physically justified. To be more specific, the tracked node's bending rotations (meaning rotations in the plane of the reference segment) are constrained to the reference segment rotational degrees of freedom if this happens to stem from a shell element. Otherwise, the rotations are constrained to the reference segment rotation as determined from its individual nodal translations. The tracked node's torsional rotations (that is, rotations with respect to the normal of the reference segment) are always constrained according to this latter philosophy, thus avoiding a torsional constraint to the relatively weak drilling mode of shells. Thus, this tied contact formulation properly treats both bending *and* torsional rotations. Here the tracked node's rotational degrees of freedom typically come from shell or beam elements. As a result of how rotational constraints are applied, it is sufficient to only consider:

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET

for most situations (choosing between a constraint or penalty formulation). The other two “non-rotational” formulations, namely,

*CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET

*CONTACT_TIED_NODES_TO_SURFACE_OFFSET

are included in the event of not wanting to constrain rotations whatsoever.

30.14.1 Kinematics

Referring to [Figure 30.25](#), we will discuss the mathematical formulation of this tied contact formulation. We consider a tracked node t tied to a reference segment R with an offset λ . The tracked node projection onto the reference segment is denoted p . Let \mathbf{x}_t denote the tracked node's coordinate and

$$\mathbf{x}_p = \sum_{i=1}^4 h_i \mathbf{x}_r^i \quad (30.62)$$

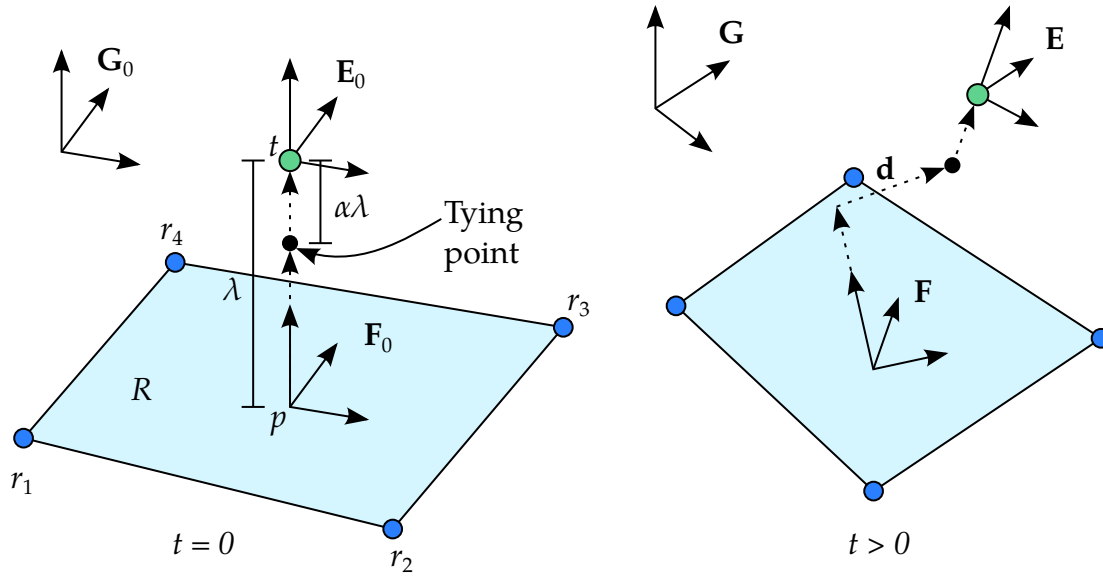


Figure 30.25. Kinematics of the implicit strong objective tied contact formulation

be the tracked node's projection onto the reference segment, where h_i are the constant isoparametric weights. To each of t , p , and R , we associate orthonormal bases (coordinate systems) represented by

$$\mathbf{E} = \{\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3\} \quad (30.63)$$

$$\mathbf{F} = \{\mathbf{f}_1 \quad \mathbf{f}_2 \quad \mathbf{f}_3\} \quad (30.64)$$

$$\mathbf{G} = \{\mathbf{g}_1 \quad \mathbf{g}_2 \quad \mathbf{g}_3\} \quad (30.65)$$

respectively. The orthogonal matrix \mathbf{G} is the reference segment coordinate system expressed as a function of the reference coordinates \mathbf{x}_r^i with normal \mathbf{g}_3 . At $t = 0$, we initialize $\mathbf{E}_0 = \mathbf{F}_0 = \mathbf{G}_0$, but \mathbf{E} and \mathbf{F} then evolve independently based on the nodal rotational velocities ω_t and

$$\omega_p = \sum_{i=1}^4 h_i \omega_r^i. \quad (30.66)$$

In the numerical implementation, if $\Delta\theta_t = \omega_t \Delta t$ and $\Delta\theta_p = \omega_p \Delta t$ are the incremental rotations of t and p at a given time step, then the coordinate systems are updated

$$\mathbf{E}_{n+1} = \mathbf{R}(\Delta\theta_t) \mathbf{E}_n \quad (30.67)$$

$$\mathbf{F}_{n+1} = \mathbf{R}(\Delta\theta_p) \mathbf{F}_n \quad (30.68)$$

where $\mathbf{R}(\Delta\theta)$ denotes the finite rotation matrix corresponding to the rotational increment $\Delta\theta$. Of course, these expressions only make sense if the tracked node and/or reference segment have rotational degrees of freedom. If the tracked node lacks rotational degrees of freedom, then neither bending nor torsion is constrained, and \mathbf{E} is of no interest. If the reference segment lacks rotational degrees of freedom, we use $\mathbf{F} = \mathbf{G}$, which is still of

interest to deal with the offset. \mathbf{G} is always well-defined. LS-DYNA automatically detects rotational degrees of freedom and takes the proper measures.

30.14.2 Translational Constraint

Given the notation in the previous section and referring to [Figure 30.25](#),

$$\mathbf{d} = \{\mathbf{x}_t - \alpha\lambda\mathbf{e}_3\} - \{\mathbf{x}_p + (1 - \alpha)\lambda\mathbf{f}_3\} \quad (30.69)$$

represents the vector we want to be zero in the translational part of the contact. Here λ is the offset distance between p and t . It is constant throughout the simulation. Furthermore, α is a constant between 0 and 1 that determines the actual tying point according to the following simple rules. First, if the tracked node is connected to a solid or beam element *or* if the contact definition does not take rotational degrees of freedom into account, then $\alpha = 0$. Otherwise, if the tracked node is connected to a shell element and the reference segment is connected to a solid element, then $\alpha = 1$. If neither of those situations apply, then both the tracked and reference sides must connect to shell elements. In this case, $\alpha = 0.5$. Note that the value of α is not accounting for the relative difference in shell thicknesses but assumes equal shell thickness on both the reference and tracked sides. For `*CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET` and `*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET`, this condition is imposed as a constraint:

$$\mathbf{d} = \mathbf{0} \quad (30.70)$$

Whereas for `*CONTACT_TIED_NODES_TO_SURFACE_OFFSET` and `*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET`, a penalty formulation is used. To this end, we use a constitutive relation between force and displacement

$$\mathbf{f} = K_f \mathbf{d} \quad (30.71)$$

with K_f is the penalty stiffness. Then an energy principle is employed to identify the nodal forces and moments,

$$\mathbf{f}^T \delta \mathbf{d} = \mathbf{P}_f^T \delta \mathbf{X}, \quad (30.72)$$

where \mathbf{X} is a nodal coordinate array of the tracked-reference pair, and δ is the variation operator. Identifying \mathbf{B}_f by

$$\delta \mathbf{d} = \mathbf{B}_f \delta \mathbf{X} \quad (30.73)$$

the nodal force array is given by

$$\mathbf{P}_f = \mathbf{B}_f^T \mathbf{f}. \quad (30.74)$$

It is worth noticing that \mathbf{B}_f is the constraint matrix corresponding to the constraint variant of the contact.

30.14.3 Bending and Torsion Constraint

If the tracked node has rotational degrees of freedom and the contact formulation is said to treat those, bending and torsion is constrained. The bending strain is calculated as

$$\varphi_j = \mathbf{g}_j^T (\mathbf{e}_3 \times \mathbf{f}_3), \quad j = 1, 2 \quad (30.75)$$

which essentially measures the amount \mathbf{e}_3 and \mathbf{f}_3 deviate from being parallel. The torsional strain is a scalar given by

$$\varphi_3 = \frac{1}{2} (\mathbf{e}_1^T \mathbf{g}_2 - \mathbf{e}_2^T \mathbf{g}_1) \quad (30.76)$$

and is a measure of the relative rotation between \mathbf{E} and \mathbf{G} with respect to the normal, \mathbf{g}_3 . The constraint to enforce it is:

$$\boldsymbol{\varphi} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{pmatrix} = \mathbf{0} \quad (30.77)$$

which for a penalty formulation leads to a constitutive relation between the moment and rotation vector

$$\mathbf{m} = K_m \boldsymbol{\varphi} \quad (30.78)$$

with K_m being a stiffness parameter. Following the approach for the translational treatment, we obtain the nodal force contribution:

$$\mathbf{P}_m = \mathbf{B}_m^T \mathbf{m} \quad (30.79)$$

where

$$\delta \boldsymbol{\varphi} = \mathbf{B}_m \delta \mathbf{X}. \quad (30.80)$$

Also, here \mathbf{B}_m is the constraint matrix in case a constraint formulation is used.

The formulae for K_f and K_m are found in earlier sections on tied interfaces, while the expressions for the matrices \mathbf{B}_f and \mathbf{B}_m are quite involved and omitted for the sake of clarity. The nodal forces and moments are implemented by using (manual) algorithmic differentiation.

30.15 Sliding-Only Interfaces

This option is seldom useful in structural calculations. Its chief usefulness is for treating interfaces where the gaseous detonation products of a high explosive act on a solid material. The present algorithm, though simple, has performed satisfactorily on a number of problems of this latter type. We briefly outline the approach here.

The method consists of five steps. In the first step, the mass per unit area (mass/area) and pressure are found at each node on the tracked surface. Next, the contact point for each reference node is found, and the tracked mass/area and tracked pressure at each reference node is interpolated from the tracked surface. In the third step, this pressure distribution is applied to the reference surface to update its acceleration. In the fourth step, the normal component of the acceleration at each node on the reference surface is scaled by its z-factor defined as the mass/area of the reference surface at the reference node divided by the sum of the mass/area of the tracked surface at the reference

node. The last step consists of resetting the normal acceleration and velocity components of all tracked nodes to ensure compatibility.

30.16 Single Surface Contact Algorithms in LS-DYNA

The single surface contact algorithms evolved from the surface-to-surface contact algorithms and the post-contact searching follows the procedures employed for the surface-to-surface contact. Type 4 contact (*OPTION1* = SINGLE_SURFACE for *CONTACT) uses the following steps where NSEG is the number of contact segments and NTN is the number of nodes in the interface:

- Loop through the contact segments from 1 to NSEG
 - Compute the normal segment vectors and accumulate an area-weighted average at the nodal points to determine the normal vectors at the nodal points.
- Loop through the tracked nodes from 1 to NTN
 - Check all nearest nodes, stored from the bucket sort, and locate the node which is nearest.
 - Check to see if nearest node is within a penetration tolerance determined during the bucket sort, if not, proceed to the end of the loop.
 - For shell elements, determine if the nearest node is approaching the segment from the positive or negative side based on the right-hand rule. Project both the node and the contact segment along the nodal normal vectors to account for the shell thickness.
 - Check for interpenetrating nodes and if a node has penetrated apply a nodal point force that is proportional to the penetration depth.

End of Loop

Of course, several obvious limitations of the above procedure exist. The normal vectors that are used to project the contact surface are meaningless for nodes along an intersection of two or more shell surfaces (see [Figure 30.26](#)). In this case the normal vector will be arbitrarily skewed depending on the choice of the numbering of the connectivities of the shells in the intersecting surfaces. Secondly, by considering the possibility of just one contact segment per node, metal forming problems cannot be handled within one contact definition. For example, if a workpiece is constrained between a die and a blankholder then at least some nodal points in the workpiece must necessarily be in contact with two segments—one in the die and the other in the blankholder. These two important limitations motivated the development of the bucket sorting algorithm described in [Section 30.9.3](#) and the modified single surface contact procedure, type 13 (*OPTION1* = AUTOMATIC_SINGLE_SURFACE).

A major change in type 13 contact from type 4 is the elimination of the normal nodal vector projection by using the segment normal vector as shown in [Figure 30.26](#). Segment numbering within the contact surface is arbitrary when the segment normal is used greatly simplifying the model input generation. However, additional complexity is introduced since special handling of the nodal points is required at segment intersections where nodes may approach undetected as depicted in [Figure 30.27a](#). To overcome this limitation, additional logic puts cylindrical caps at segment intersections in contact type 13 (and a3). See [Figure 30.27b](#).

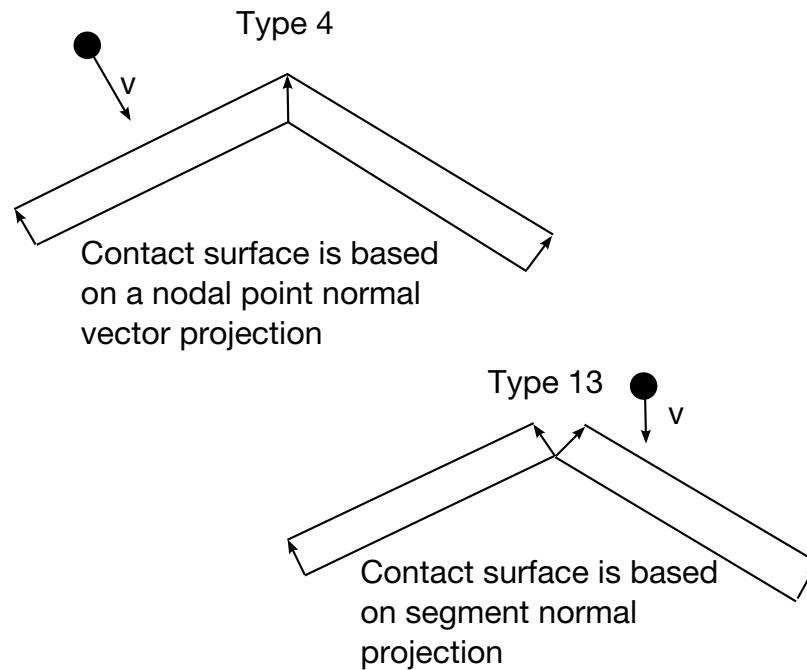


Figure 30.26. Projection of the contact surface for a node approaching from above is shown for types 4 and 13 contact.

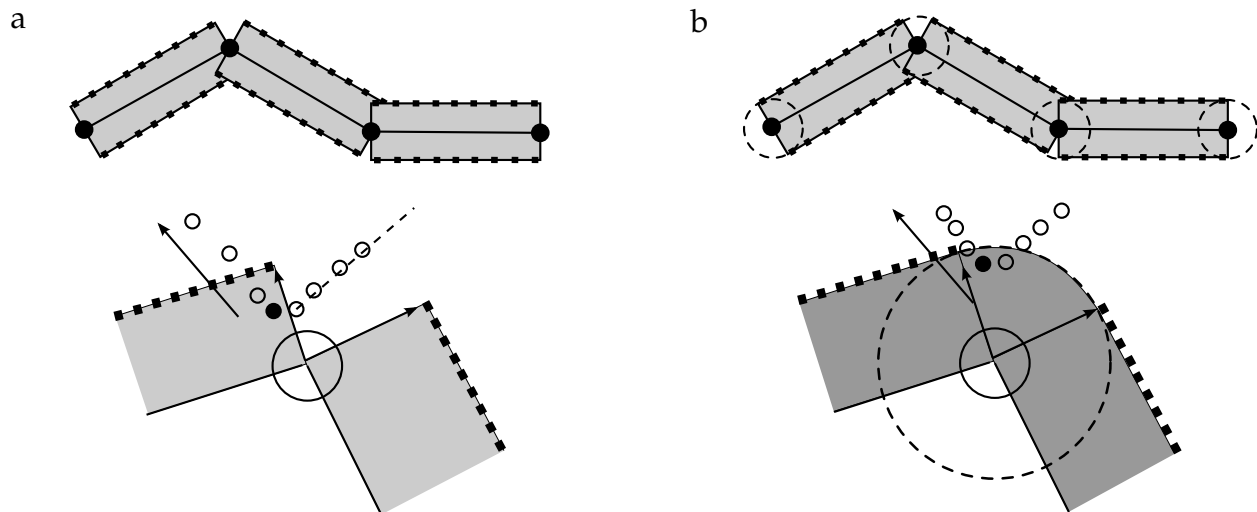


Figure 30.27. Contact treatment at segment intersections. The left figure shows that without adding cylindrical caps to the segment intersections, contact could be missed with segment normal projection. Cylindrical caps were added for AUTOMATIC_SINGLE_SURFACE contact to treat this limitation.

Assuming that segment-based bucket sort has been completed and closest segments are known for all tracked nodes, the procedure for processing the type 13 contact simplifies to:

- Loop through the tracked nodes from 1 to NTN

- If node is in contact, check to see if the contact segment has changed, and if so, then update the closest segment information and the orientation flag which remembers the side in contact. Since no segment orientation information is stored this flag may change as the node moves from segment to segment.
- Check the closest segment to see if the node is in contact. If not, then proceed to the end of the loop. If the tracked node or contact segment connectivity is a member of a shell element, project both the node and the contact segment along the segment normal vector to account for the shell thickness. A nodal thickness is stored for each node. and a segment thickness is stored for each segment. A zero thickness is stored for solid elements. The thickness can be optionally updated to account for membrane thinning.
- Check for interpenetrating nodes, and if a node has penetrated, apply a nodal point force that is proportional to the penetration depth.

End of Loop

Note that type 13 contact does not require the calculation of nodal normal vectors.

In addition to SINGLE_SURFACE and AUTOMATIC_SINGLE_SURFACE, LS-DYNA single surface contacts also include AIRBAG_SINGLE_SURFACE, AUTOMATIC_GENERAL, and SINGLE_EDGE. AIRBAG_SINGLE_SURFACE is based on AUTOMATIC_SINGLE_SURFACE with modifications specifically for airbag deployment. AUTOMATIC_GENERAL is like AUTOMATIC_SINGLE_SURFACE, but it handles beam contact and shell edge contact differently. *CONTACT_AUTOMATIC_GENERAL includes treatment of beam-to-beam contact in which it checks the entire length of the beam for penetration, not just the nodes of the beam. For shells, it checks for penetration along the entire length of the exterior (unshared) shell edges. *CONTACT_AUTOMATIC_GENERAL essentially adds null beams to the exterior edges of shell parts so that edge-to-edge treatment of the shell parts is handled by virtue of contact through the automatically generated null beams. *CONTACT_SINGLE_EDGE is based on single surface contact but only treats edge-to-edge contact. It should be used with contact types that do not already have edge treatment. For details about how this contact works, see Section 30.21.

30.17 Surface-to-Surface Constraint Algorithm

The constraint algorithm is based on the algorithm developed by Taylor and Flanagan [1989]. This algorithm is enabled with *OPTION1* for *CONTACT set to CONSTRAINT_SURFACE_TO_SURFACE. This algorithm involves a two-pass symmetric approach with a partitioning parameter, β , that is set between negative and positive unity. $\beta = 1$ and $\beta = -1$ correspond to one-way treatments with SURFB accumulating the mass and forces from SURFA (for $\beta = 1$) and visa versa (for $\beta = -1$). In other words, for $\beta = 1$, SURFA acts as the tracked surface and SURFB acts as the reference surface, while for $\beta = -1$ the

treatment is vice versa. The searching algorithms are those used in the other contact algorithms for surface-to-surface contact.

In this constraint approach, the accelerations, velocities, and displacements are first updated to a trial configuration without accounting for interface interactions. After the update, a penetration force is computed for the tracked node as a function of the penetration distance, ΔL :

$$\mathbf{f}_p = \frac{m_{tr}\Delta L}{\Delta t^2} \mathbf{n}, \quad (30.81)$$

where \mathbf{n} is the normal vector to the reference surface.

We desire that the response of the normal component of the tracked node acceleration vector, \mathbf{a}_{tr} , of a tracked node residing on reference segment k be consistent with the motion of the reference segment at its contact segment (s_c, t_c) :

$$\mathbf{a}_{tr} = \phi_1(s_c, t_c) \mathbf{a}_{nk}^1 + \phi_2(s_c, t_c) \mathbf{a}_{nk}^2 + \phi_3(s_c, t_c) \mathbf{a}_{nk}^3 + \phi_4(s_c, t_c) \mathbf{a}_{nk}^4. \quad (30.82)$$

For each tracked node in contact with and penetrating through the reference surface in its trial configuration, its nodal mass and its penetration force given by Equation (30.81) are accumulated to a global reference surface mass and force vector:

$$\left(m_k + \sum_{tr} m_{k_{tr}} \right) \mathbf{a}_{nk} = \sum_{tr} \mathbf{f}_{k_{tr}}, \quad (30.83)$$

where

$$m_{k_{tr}} = \phi_k m_{tr}, \quad (30.84)$$

$$\mathbf{f}_{k_{tr}} = \phi_k \mathbf{f}_{tr}. \quad (30.85)$$

After solving Equation (30.83) for the acceleration vector, \mathbf{a}_{nk} , we can obtain the acceleration for the tracked node, \mathbf{a}_{tr} , through the interpolation given in Equation (30.82). Thus, the acceleration correction for the tracked node is

$$\mathbf{a}_{ntr} = \mathbf{a}_{tr} - \frac{\mathbf{f}_p}{m_{tr}}. \quad (30.86)$$

The above process is repeated after reversing SURFA and SURFB from tracked and reference to reference and tracked. In the final step the averaged final correction to the acceleration vector is found

$$\mathbf{a}_n^{\text{final}} = \frac{1}{2} (1 + \beta) \mathbf{a}_n^{\text{1st pass}} + \frac{1}{2} (1 - \beta) \mathbf{a}_n^{\text{2nd pass}}, \quad (30.87)$$

and used to compute the final acceleration at time $n + 1$

$$\mathbf{a}^{n+1} = \mathbf{a}^{\text{trial}} + \mathbf{a}_n^{\text{final}}, \quad (30.88)$$

Friction, as described by Taylor and Flanagan [1989], is included in our implementation. Friction resists the relative tangential velocity of the tracked node with respect to the reference surface. This relative velocity is determined by:

$$\mathbf{v}_r = \mathbf{v}_{tr} - (\phi_1 \mathbf{v}_k^1 + \phi_2 \mathbf{v}_k^2 + \phi_3 \mathbf{v}_k^3 + \phi_4 \mathbf{v}_k^4), \quad (30.89)$$

with the tangential component given by:

$$\mathbf{v}_t = \mathbf{v}_r - (\mathbf{n} \cdot \mathbf{v}_r)\mathbf{n}. \quad (30.90)$$

A trial tangential force is computed that cancels the tangential velocity

$$f^* = -\frac{m_{tr}v_t}{\Delta t}, \quad (30.91)$$

where v_t is the magnitude of the tangential velocity vector

$$v_t = \sqrt{\mathbf{v}_t \cdot \mathbf{v}_t}. \quad (30.92)$$

The magnitude of the tangential force is limited by the magnitude of the product of the Coulomb friction constant with the normal force defined as

$$f_n = m_{tr}\mathbf{a}_{ntr} \cdot \mathbf{n}, \quad (30.93)$$

The limiting force is, therefore,

$$F_y = \mu|\mathbf{f}_n|, \quad (30.94)$$

and

$$\mathbf{f}^{n+1} = \mathbf{f}^* \quad \text{if } |\mathbf{f}^*| \leq F_y, \quad (30.95)$$

$$\mathbf{f}^{n+1} = \frac{F_y \mathbf{f}^*}{|\mathbf{f}^*|} \quad \text{if } |\mathbf{f}^*| > F_y. \quad (30.96)$$

Therefore, using the above equations, the modification to the tangential acceleration component of the tracked node is given by

$$a_t = \min \left(\mu \mathbf{a}_{nt} \cdot \mathbf{n}, \frac{|\mathbf{v}_t|}{\Delta t} \right), \quad (30.97)$$

which must act in the direction of the tangential vector defined as

$$\mathbf{n}_t = \frac{\mathbf{v}_t}{v_t}. \quad (30.98)$$

The corrections to both the tracked and reference node acceleration components are:

$$\mathbf{a}_{ttr} = a_t \mathbf{n}_t, \quad (30.99)$$

$$\mathbf{a}_{tk} = -\phi_k \frac{a_t m_{tr}}{m_k} \mathbf{n}_t, \quad (30.100)$$

The above process is again repeated after reversing SURFA and SURFB from tracked and reference to reference and tracked. In the final step the averaged final correction to the acceleration vector is found

$$\mathbf{a}_t^{\text{final}} = \frac{1}{2}(1 + \beta)\mathbf{a}_t^{\text{1st pass}} + \frac{1}{2}(1 - \beta)\mathbf{a}_t^{\text{2nd pass}}, \quad (30.101)$$

and is used to compute the final acceleration at time $n + 1$

$$\mathbf{a}^{n+1} = \mathbf{a}^{\text{trial}} + \mathbf{a}_n^{\text{final}} + \mathbf{a}_t^{\text{final}}. \quad (30.102)$$

A significant disadvantage of the constraint method relative to the penalty method appears if an interface node is subjected to additional constraints such as spot welds, constraint equations, tied interfaces, and rigid bodies. Rigid bodies can often be used with this contact algorithm if their motions are prescribed as is the case in metal forming. For the more general cases involving rigid bodies, the above equations are not directly applicable since the local nodal masses of rigid body nodes are usually meaningless. Subjecting the two sides of a shell surface to this constraint algorithm also leads to erroneous results since an interface node cannot be constrained to move simultaneously on two mutually independent surfaces. In the latter case, the constraint technique could be used on one side and the penalty method on the other.

The biggest advantage of the constraint algorithm is that interface nodes remain on or very close to the surfaces with which they are in contact. Furthermore, elastic vibrations that can occur in penalty formulations are insignificant with the constraint technique. The problem related to finding good penalty constants for the contact are totally avoided by the latter approach. Having both methods available is possibly the best option of all.

30.18 Rigid Boundaries

The rigid boundary represents the simplest contact problem. We, therefore, treat it separately. In this section, we discuss both planar and geometric boundaries. To invoke these boundaries, use *RIGIDWALL_PLANAR and *RIGIDWALL_GEOMETRIC.

30.18.1 Planar Rigid Boundaries

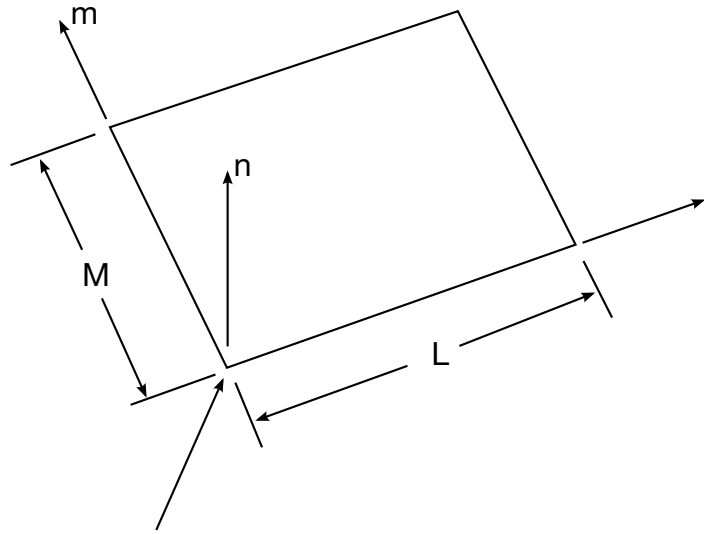
LS-DYNA supports flat, planar boundaries of finite or infinite extent. [Figure 30.28](#) illustrates a finite planar boundary. Defining a planar boundary requires specifying an outward normal unit vector \mathbf{n} with the origin of \mathbf{n} at a corner point on the wall if the wall is finite or at an arbitrary point on the wall if the wall extends to infinity. The finite wall is rectangular with edges of length L and M . Unit vectors \mathbf{l} and \mathbf{m} lie along these edges.

The input for *RIGIDWALL_PLANAR includes selecting a subset of nodes, usually boundary nodes of the calculational model, that are not allowed to penetrate. These are the tracked nodes. Let k represent one such tracked node and let \mathbf{r}_k^{n+1} be the position vector from the origin of \mathbf{n} to k after locally updating the coordinates. At each time step prior to globally updating the velocities and accelerations, we check each node for penetration. For finite walls we first determine whether the node lies within the wall by checking if the node's position satisfies the following inequalities :

$$\begin{aligned}\mathbf{r}_k^{n+1} \cdot \mathbf{l} &\leq L, \\ \mathbf{r}_k^{n+1} \cdot \mathbf{m} &\leq M.\end{aligned}\tag{30.103}$$

This test is skipped for the infinite rigid wall. Assuming that the inequality is satisfied, we then check the penetration condition to see if k is penetrating through the wall,

$$\mathbf{r}_k^{n+1} \cdot \mathbf{n} < 0,\tag{30.104}$$



Origin, if extent of stonewall is finite

Figure 30.28. Vector \mathbf{n} is normal to the stonewall. An optional vector \mathbf{l} can be defined such that $\mathbf{m} = \mathbf{n} \times \mathbf{l}$. The extent of the stonewall is limited by defining \mathbf{L} and \mathbf{M} . A zero value for either of these lengths indicates that the stonewall is infinite in that direction.

and if so, the velocity and acceleration components normal to the wall are set to zero:

$$\begin{aligned} \mathbf{a}_{k_{\text{new}}}^n &= \mathbf{a}_{k_{\text{old}}}^n - (\mathbf{a}_{k_{\text{old}}}^n \cdot \mathbf{n})\mathbf{n}, \\ \mathbf{v}_{k_{\text{new}}}^n &= \mathbf{v}_{k_{\text{old}}}^n - (\mathbf{v}_{k_{\text{old}}}^n \cdot \mathbf{n})\mathbf{n}. \end{aligned} \quad (30.105)$$

Here \mathbf{a}_k and \mathbf{v}_k are the nodal acceleration and velocity of node k , respectively. This procedure for stopping nodes represents a perfectly plastic impact resulting in irreversible energy loss. The total energy dissipated is found by taking the difference between the total kinetic energy of all the tracked nodal points of the rigid wall before and after impact with the wall. This energy is computed and accumulated in LS-DYNA and is printed in the `glstat` (global statistics) file.

The tangential motion of the tracked node may be unconstrained, fully constrained, or subjected to Coulomb friction while it is in contact with the rigid boundary. Coulomb friction acts along a vector defined as:

$$\mathbf{n}_t = \frac{\mathbf{v}_{k_{\text{new}}}^n}{\sqrt{\mathbf{v}_{k_{\text{new}}}^n \cdot \mathbf{v}_{k_{\text{new}}}^n}} \quad (30.106)$$

The magnitude of the tangential force which is applied to oppose the motion is given as

$$f_t = \min \left(\frac{m_t \sqrt{\mathbf{v}_{k_{\text{new}}}^n \cdot \mathbf{v}_{k_{\text{new}}}^n}}{\Delta t}, \mu |\mathbf{f}_n| \right), \quad (30.107)$$

meaning the maximum value required to hold the node in the same relative position on the stonewall or the product of the coefficient of friction and the magnitude of the normal

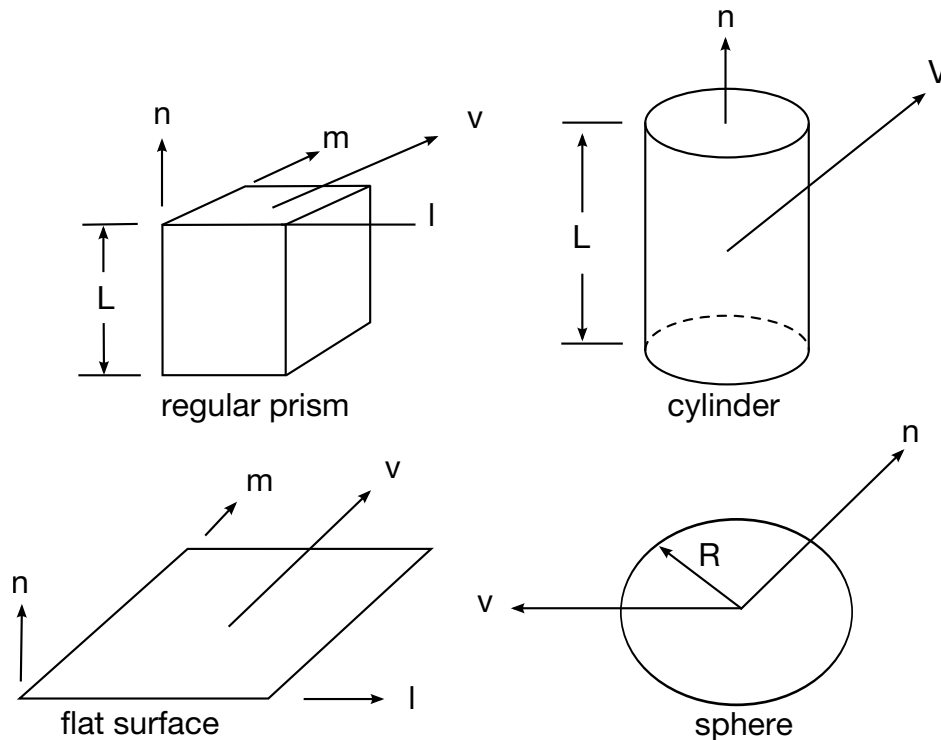


Figure 30.29. Vector \mathbf{n} determines the orientation of the generalized stonewalls. For the prescribed motion options the wall can be moved in the direction \mathbf{V} as shown.

force, whichever is less. In Equation (30.107), m_t is the mass of the tracked node and \mathbf{f}_n is the normal force.

30.18.2 Geometric Rigid Boundaries

The numerical treatment of geometric rigid walls is somewhat similar to that for the finite planar rigid walls. The geometric rigid walls can be subjected to a prescribed translational motion along an arbitrarily oriented vector. However, rotational motion is not permitted. As the geometric surface moves and contacts the structure, external work is generated which is integrated and added to the overall energy balance. In addition to the external work, plastic work also is generated as nodes contact the wall and assume the walls normal velocity at the point of contact. Contact can occur with any of the surfaces that enclose the volume. Currently, four geometric shapes are available including the rectangular prism, the cylinder, flat surface, and sphere. These are shown in Figure 30.29.

30.19 VDA/IGES Contact

This capability enables reading VDA/IGES surfaces directly into LS-DYNA for analysis as contact surfaces. No mesh generation is required. LS-DYNA performs the contact against the analytic surface. LS-DYNA supports the VDA standard and an important subset of the IGES entities including:

- #100 Circle arc
- #102 Composite Curve
- #106 Copious data
- #110 Lines
- #112 Parametric polynomial curve
- #114 Parametric polynomial surface
- #116 Points
- #126 NURBS Curves
- #128 NURBS Surfaces
- #142 Curve on Parametric Surface
- #144 Trimmed Parametric Surfaces
- #402 form 7-group
- #406 form 15-associate name

To use the VDA/IGES contact capability, first specify which VDA/IGES surfaces, faces, and groups should be attached to each material. This is done primarily through a special input file. Use groups to combine faces, surfaces, and groups from several different VDA/IGES input files that can be referred to by an alias. For example, for a simple sheet metal forming problem, an input file might look like this:

```
file punch.vda punch.bin {
  alias punch { grp001 }
}
file die.vda die.bin {
  alias part1 { fce001 sur002 }
  alias part2 { fce003 }
}
file die2.vda die2.bin {
  alias part3 { fce004 }
}
file holder.vda holder.bin {
  alias holder { sur001 sur002 }
}
alias die { part1 part2 part3 }
end
```

In the above example, the punch is made up of the group "grp001" from the file "punch.vda". The VDA file is converted to a binary file "punch.bin". If this simulation is ever rerun, LS-DYNA can read the VDA input directly from the binary file, thereby significantly reducing startup time. The die in this example is made up of several surfaces and faces from two different VDA files. This format of input enables combining any

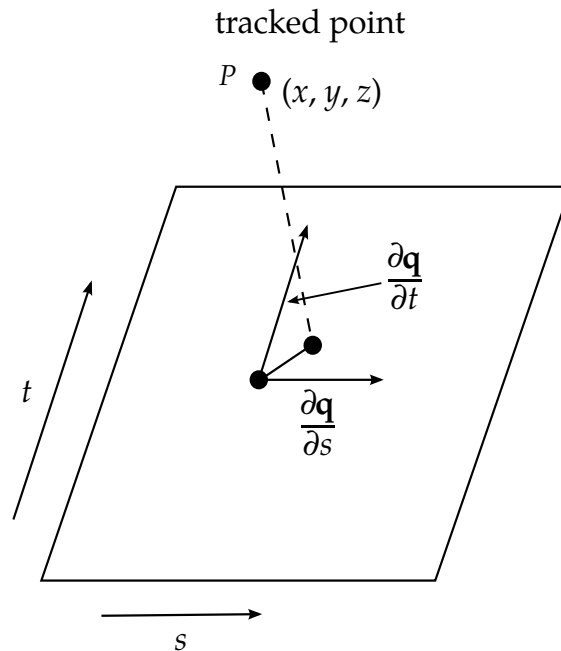


Figure 30.30. The geometry of the patch is a function of the parametric coordinates s and t .

number of faces, surfaces, and groups from any number of VDA files to define a single part. The LS-DYNA input file then references this single part name.

The contact algorithm works as follows. For the sake of simplicity, we will refer to one point as being tracked by a single part. Again, this part will in general be made up of several VDA surfaces and faces. First, the distance from the tracked point to each VDA surface is computed and stored. For the surface nearest to the tracked point, several other parameters are stored, such as the surface coordinates of the nearest point on the surface. This information is updated during each time step of the calculation. For the nearest surface, the new nearest point on the surface is calculated. For all other surfaces, the distance the tracked point moves is subtracted from the distance to the surface. These updates continually give a lower bound on the actual distance to each VDA surface. When this lower bound drops below the thickness of the point being tracked, the actual distance to the surface is recalculated. Actually, if the nearest surface is further away from the tracked point than some distance, the nearest point on the surface is not tracked at all until the tracked point comes close to some surface. These precautions result in the distance from the tracked point to a surface having to be totally recomputed every few hundred timesteps, in exchange for not having to continually track the point on each surface.

To track the point on the nearest surface, a 2D form of Newton's method is used. The vector function to be solved specifies that the displacement vector from the surface to the tracked point should be parallel to the surface normal vector. The surface tangent vectors are computed with respect to each of the two surface patch parameters, and the dot product taken with the displacement vector. See [Figure 30.30](#) and [Equation \(1.108\)](#).

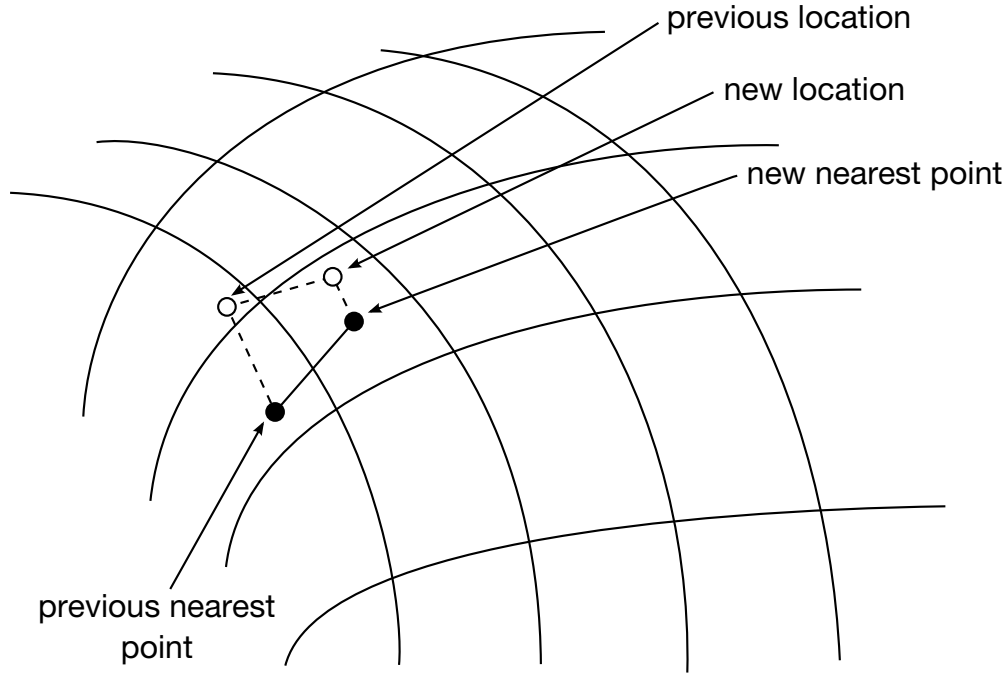


Figure 30.31. Newton iteration solves for the nearest point on the analytical surface.

$$(\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{q}}{\partial s} = 0 \quad \text{and} \quad (\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{q}}{\partial t} = 0. \quad (30.108)$$

This vector equation is then solved using Newton's method as in [Equation \(1.109\)](#).

$$\mathbf{q}_{i+1} = \mathbf{q}_i - (\mathbf{F}')^{-1} \mathbf{F}, \quad (30.109)$$

where

$$\mathbf{F}(s, t) = \begin{pmatrix} (\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{q}}{\partial s} \\ (\mathbf{p} - \mathbf{q}) \cdot \frac{\partial \mathbf{q}}{\partial t} \end{pmatrix}. \quad (30.110)$$

The convergence is damped in the sense that the surface point is not allowed to jump completely outside of a surface patch in one iteration. If the iteration point tries to leave a patch, it is placed in the neighboring patch, but on the adjoining boundary. This prevents the point from moving merely continuously (i.e., when the surface has a crease in it). Iteration continues until the maximum number of allowed iterations is reached, or a convergence tolerance is met. The convergence tolerance (as measured in the surface patch parameters) varies from patch to patch, and is based on the size and shape of the patch. The convergence criterion is set for a patch to ensure that the actual surface point has converged (in the spatial parameters x , y , and z) to some tolerance.

30.20 Simulated Draw Beads

The implementation of draw beads is based on elastic-plastic interface springs and nodes-to-surface contact. The area of the blank under the draw bead is taken as the reference surface. The draw bead is defined by a *consecutive* list of nodes that lie along the draw bead. For straight draw beads only two nodes need to be defined, but for curved beads sufficient nodes must be used to define the curvature. The draw bead line is discretized into points that become the tracked nodes to the reference surface. The spacing of the points is determined by LS-DYNA such that several points lie within each reference segment. This is illustrated in Figure 30.32. The dense distribution of point leads to a smooth draw bead force distribution which helps avoid exciting the zero energy (hourglass) modes within the shell elements in the workpiece. A three-dimensional bucket search is used for the contact searching to locate each point within a segment of the reference surface.

The nodes defining the draw beads can be attached to rigid bodies by using the extra nodes for rigid body input option. When defining draw beads, care should be taken to limit the number of elements that are used in the reference surface definition. If the entire blank is specified, the CPU cost increases significantly, and the memory requirements can become enormous. An automated draw bead box, which is defined by specifying the part ID for the workpiece and the node set ID for the draw bead, is available. The automated box option allows LS-DYNA to determine the box dimensions. The size of this box is based on the extent of the blank and the largest element in the workpiece as shown in Figure 30.33.

The input for the draw beads requires a load curve giving the force due to the

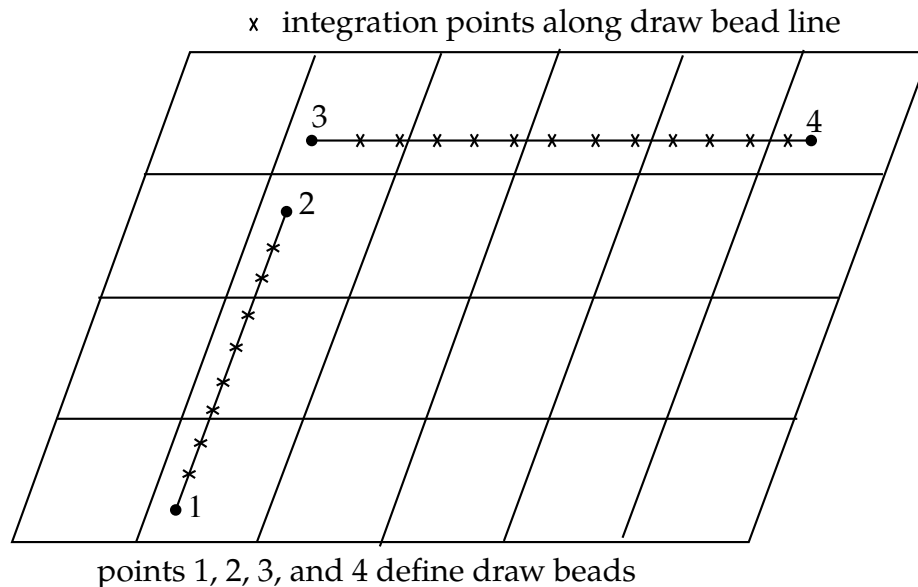


Figure 30.32. Draw bead contact provides a simple way of including draw bead behavior without the necessity of defining a finite element mesh for the draw beads. Since the draw bead is straight, each bead is defined by only two nodes.

bending and unbending of the blank as it moves through the draw bead. The load curve

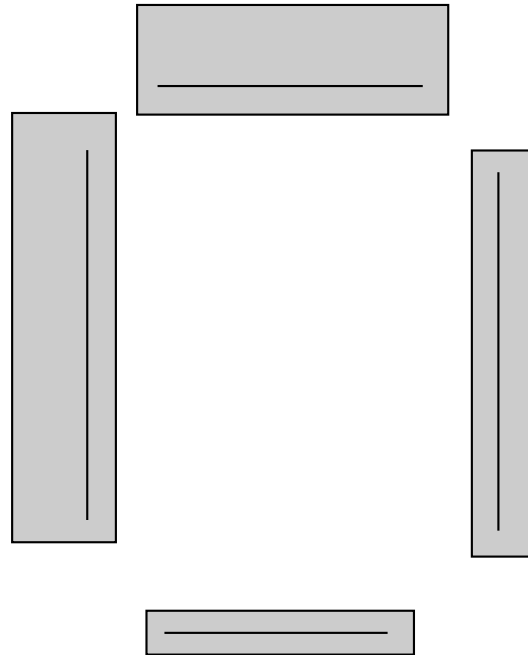


Figure 30.33. The draw bead box option automatically sizes the box around the draw bead. Any segments within the box are included as reference segments in the contact definition.

may also include the effect of friction. However, the Coulomb friction coefficients must be set to zero if the frictional component is included in the load curve. If the sign of the load curve ID is positive, the load curve gives the retaining force per unit draw bead length as a function of displacement, δ . If the sign is negative the load curve defines the maximum retaining force as a function of the normalized position along the draw bead. This position varies from 0 (at the origin) to 1 (at the end) along the draw bead. See [Figures 30.34](#) and [30.35](#).

When friction is active, the frictional force component normal to the bead in the plane of the work piece is computed. Frictional forces tangent to the bead are not allowed. The second load curve gives the normal force per unit draw bead length as a function of displacement, δ . This force is due to bending the blank into the draw bead as the binder closes on the die and represents a limiting value. The normal force begins to develop when the distance between the die and binder is less than the draw bead depth. As the binder and die close on the blank, this force should diminish or reach a plateau. This load curve was originally added to stabilize the calculation.

As the elements of the blank move under the draw bead, a plastic strain distribution develops through the shell thickness due to membrane stretching and bending. To account for this strain profile an optional load curve can be defined that gives the plastic strain as a function of the parametric coordinate through the shell thickness where the parametric coordinate is defined in the interval from -1 to 1 . The value of the plastic strain at each through thickness integration point is interpolated from this curve. If the plastic strain at an integration point exceeds the value of the load curve at the time initialization occurs, the plastic strain at the point remains unchanged. A scale factor that

multiplies the shell thickness as the shell element moves under the draw bead can also be defined as a way of accounting for any thinning that may occur.

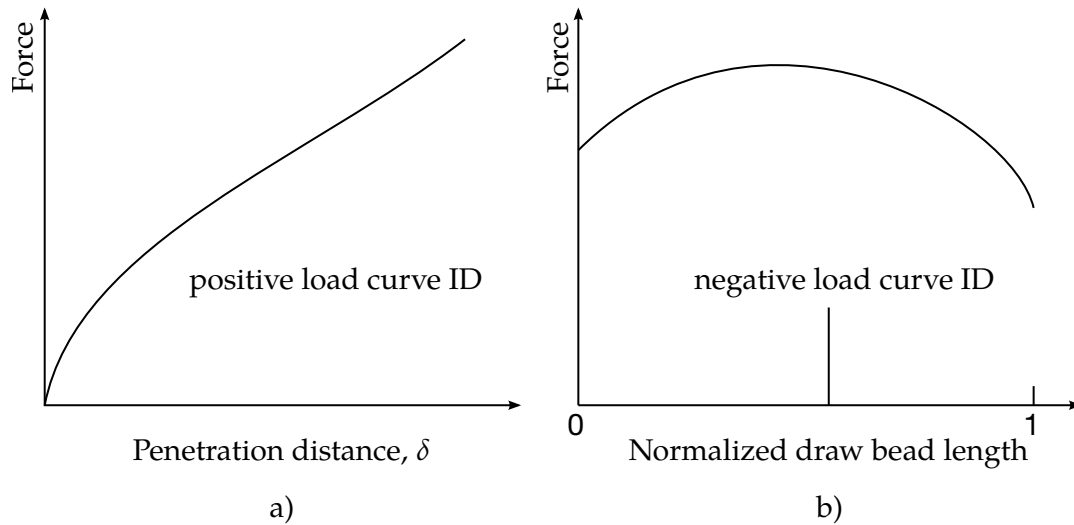


Figure 30.34. Draw bead contact model defines a resisting force as a function of draw bead displacement.

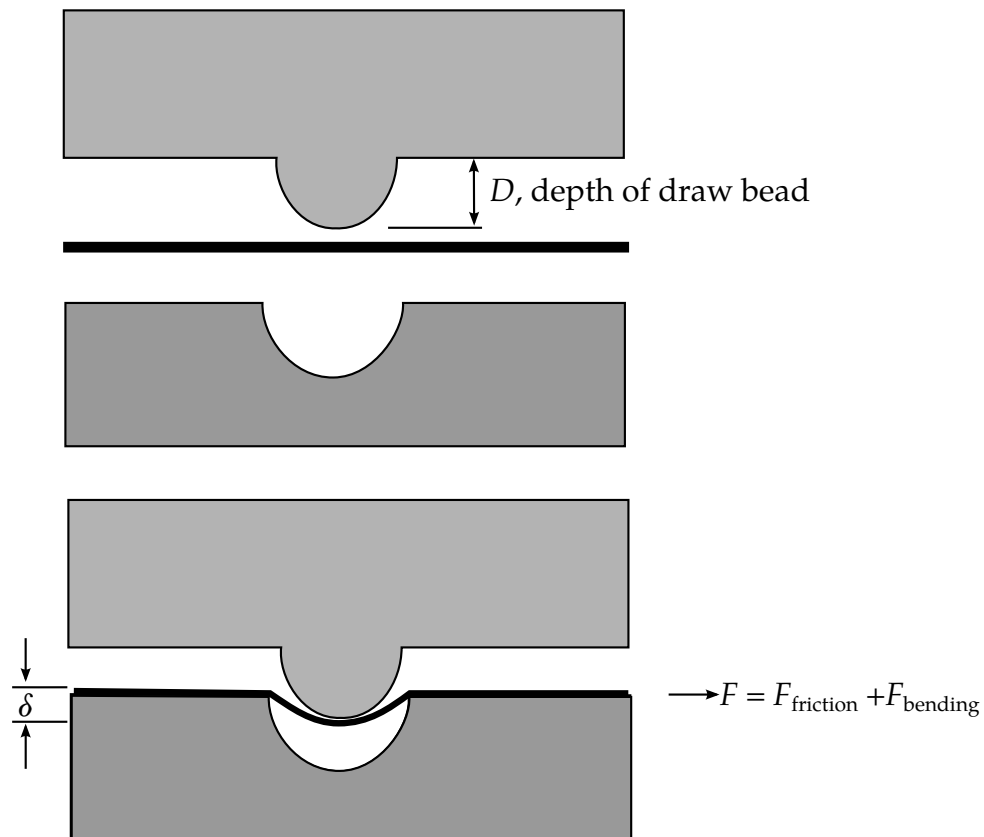


Figure 30.35. Draw bead contact model defines a resisting force as a function of draw bead displacement.

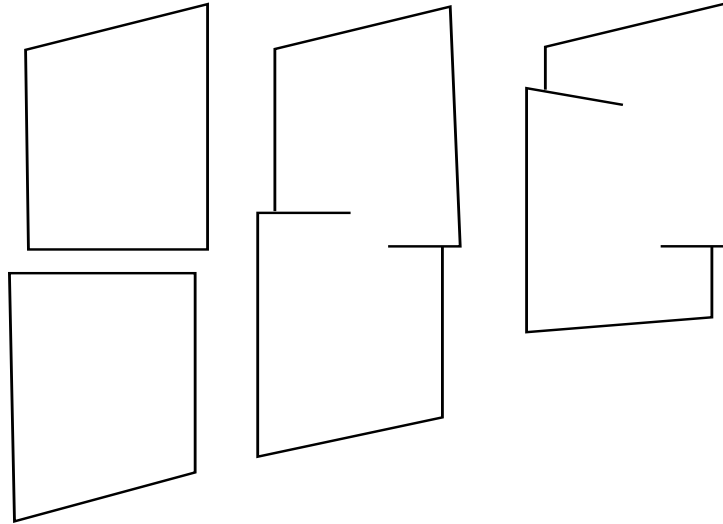


Figure 30.36. Contact between edges requires a special treatment since the nodes do not make contact.

30.21 Edge-to-Edge Contact

Edge-to-edge contact can be important in some simulations. For example, if a fan blade breaks away from the hub in a jet turbine, contact with the trailing blade will likely be along the edges of the blades. Edge-to-edge contact requires special treatment since the nodal points do not make contact with the reference segment, which is the basis of conventional contact treatments. Currently, only `*CONTACT_AUTOMATIC_GENERAL`, `*CONTACT_SINGLE_EDGE`, all contacts using the segment-to-segment ($\text{SOFT} = 2$) formulation with certain values of DEPTH , and all automatic Mortar contacts possess edge-to-edge capabilities.

Other than `*CONTACT_AUTOMATIC_GENERAL_INTERIOR` and segment-to-segment ($\text{SOFT} = 2$) contacts, the previously mentioned contacts only treat contact between exterior edges. `*CONTACT_AUTOMATIC_GENERAL_INTERIOR` and the segment-to-segment ($\text{SOFT} = 2$) contacts additionally support edge contact for interior (meshline) edges. Additionally, all the previously mentioned contacts other than segment-to-segment ($\text{SOFT} = 2$) contact only support edge contact for shell elements. The segment-to-segment ($\text{SOFT} = 2$) contact supports edge contact for shells, thick shells, and solids.

`*CONTACT_SINGLE_EDGE` is a older contact based on the single surface formulation. It has an identical input to the single surface formulation. However, it only considers contact between edges and does not treat nodes to surface contact. When the surface is defined by part ID, edge determination is automatic. It is also possible to use a manual definition by listing line segments. This contact only considers edge-to-edge contact of the type illustrated in [Figure 30.37](#). Here, the tangent vectors to the plane of the shell and normal to the edge must point to each other for contact to be considered.

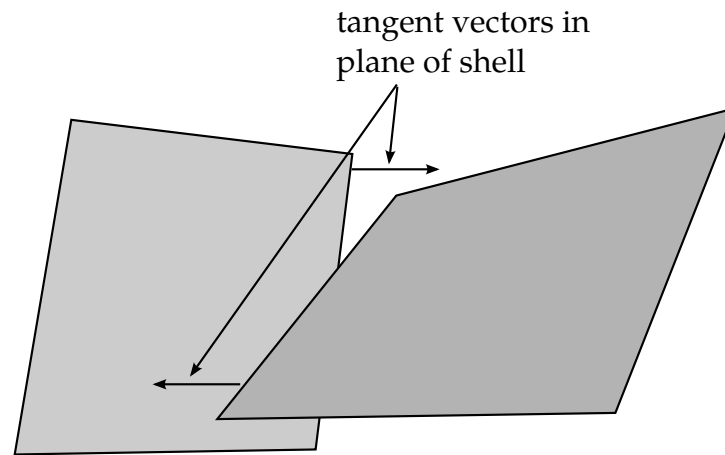


Figure 30.37. *CONTACT_SINGLE_EDGE considers contact between two edges whose normals point towards each other.

30.22 Beam-to-Beam Contact

In beam-to-beam contact, the contact surface is assumed to be the surface of a cylinder as shown in [Figure 30.38](#). The diameter of the contact cylinder is set equal to the square root of the area of the smallest rectangle that contains the cross section to avoid tracking the orientation of the beam within the contact algorithm. Contact is found by finding the intersection point between nearby beam elements and checking to see if their outer surfaces overlap as seen in [Figure 30.39](#). If the surfaces overlap, the contact force is computed and applied to the nodal points of the interacting beam elements.

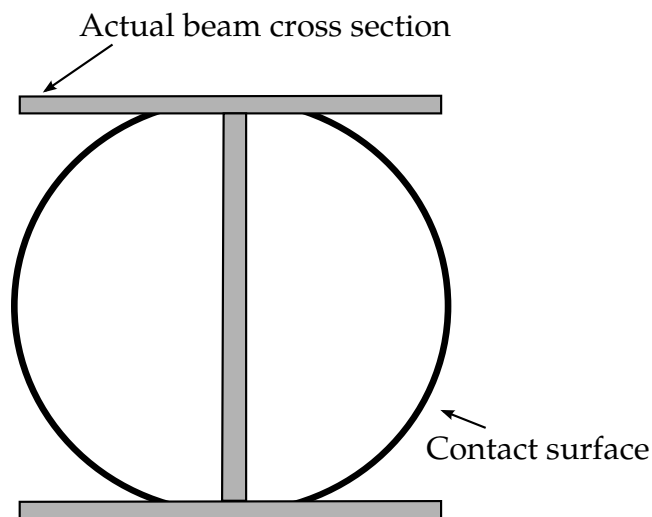


Figure 30.38. Beam contact surface approximation.

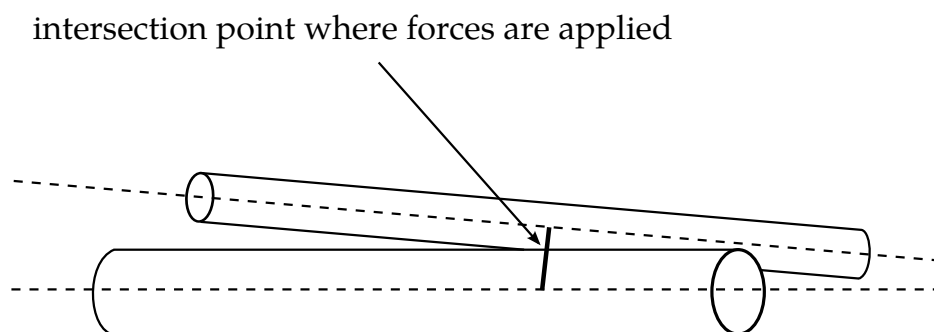


Figure 30.39. The forces are applied at the intersection point.

30.23 Mortar Contact

Mortar contact was initially implemented as a forming contact intended for stamping analysis. It has since become a general-purpose contact algorithm for implicit time integration. Mortar contact is available for automatic single and surface-to-surface contacts. It features proper edge treatment, beam contact, and optional features, including tied, tiebreak, and interference. Contact is often the one feature that hinders implicit performance. Thus, the LS-DYNA message files contain substantial information on penetrations to facilitate debugging of Mortar contacts.

Mortar contact is a penalty-based segment-to-segment contact with finite-element-consistent coupling between the non-matching discretizations of the two sliding surfaces. The implementation is based on Puso and Laursen [2004a,b]. This consistency, together with a differentiable penalty function for penetrating and sliding segments, enforces the continuity and (relative) smoothness in contact forces that is appealing when running implicit analyses. The algorithm primarily focuses on accuracy and robustness. The calculations associated with this aim make it expensive enough to be recommended for implicit analysis first and foremost.

The implementation includes numerous details that cannot be explained without making the presentation incomprehensible. The intention here is to summarize the general concepts of the theory behind the implementation and draw upon this to make general recommendations on usage.

30.23.1 Kinematics

Mortar contact is theoretically treated as a generalized finite element where each element in this context consists of a pair of contact segments. Mortar contact models friction with a standard Coulomb friction law. Each of the two segments inherits its isoparametric

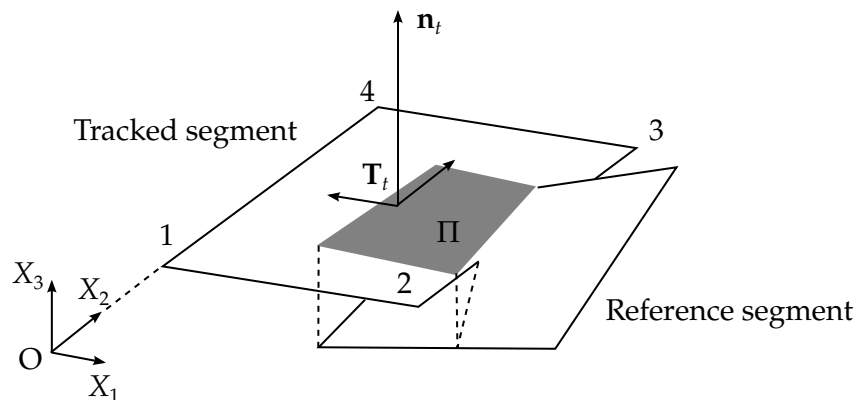


Figure 30.40. Illustration of Mortar segment-to-segment contact

representation from the underlying finite element formulation, so the coordinates for the tracked and reference segments can be written as:

$$\begin{aligned}\mathbf{x}_t &= N_t^i(\xi, \eta) \left\{ \mathbf{x}_i + \frac{1}{2} t_t \mathbf{n}_t \right\} \\ \mathbf{x}_r &= N_r^j(\xi, \eta) \left\{ \mathbf{x}_j + \frac{1}{2} t_r \mathbf{n}_r \right\},\end{aligned}\quad (30.111)$$

where summation over repeated indices is implicitly understood, meaning over the nodes. The terms $\frac{1}{2} t_t \mathbf{n}_t$ and $\frac{1}{2} t_r \mathbf{n}_r$ account for the possibility of the contact surface being offset from the midsurface of the elements, such as for shells. t_s and t_r are the thicknesses, and \mathbf{n}_t and \mathbf{n}_r are the normal vectors of the segments.

The kinematics for the contact element can be written as the penetration

$$d = \mathbf{n}_t^T (\mathbf{x}_t - \bar{\mathbf{x}}_r), \quad (30.112)$$

where \mathbf{n}_t is the tracked segment's normal vector, and $\bar{\mathbf{x}}_r$ is the projected point on the reference segment along the tracked segment's normal vector. The contact element is only defined for the intersection between the tracked and reference segments and for points where $d > 0$. This domain is denoted Π and illustrated in gray in [Figure 30.40](#). The sliding rate, \dot{s} , is similarly defined as

$$\dot{s} = \mathbf{T}_t^T (\dot{\mathbf{x}}_t - \dot{\bar{\mathbf{x}}}_r), \quad (30.113)$$

where \mathbf{T}_t is two co-rotational basis vectors pertaining to the tracked segment.

30.23.2 Constitutive Relation

The following constitutive law gives the contact pressure:

$$\sigma_n = \alpha \beta_t \beta_r \varepsilon K_t f \left(\frac{d}{\varepsilon d_c^t} \right) \quad (30.114)$$

Here,

α = stiffness scaling factor (SFSA \times SLSFAC)

K_t = stiffness modulus of the tracked segment

$\varepsilon = 0.03$

d = penetration given by [Equation \(30.112\)](#)

d_c^t = characteristic length of the tracked segment

β_t = stiffness scale factor of the tracked segment (=1 unless specifically stated)

β_r = stiffness scale factor of the reference segment (=1 unless specifically stated)

and

$$f(x) = \begin{cases} \frac{1}{4} x^2 & x < \frac{d_{\max}}{2\varepsilon d_c^t} \\ \text{cubic function that depends on IGAP} & x \geq \frac{d_{\max}}{2\varepsilon d_c^t} \end{cases} \quad (30.115)$$

where d_{\max} is the maximum penetration given in [Equation \(1.120\)](#).

The Coulomb friction law is expressed in terms of the tangential contact stress

$$\sigma_t = \mu \sigma_n \frac{\mathbf{s}}{|\mathbf{s}|} g \left(\frac{|\mathbf{s}|}{\mu d} \right), \quad (30.116)$$

where μ is the friction coefficient and

$$g(x) = \begin{cases} x & x \leq 1 - \varepsilon \\ 1 - \frac{\varepsilon}{4} \left(\frac{1 + \varepsilon - x}{\varepsilon} \right)^2 & 1 - \varepsilon < x \leq 1 + \varepsilon \\ 1 & 1 + \varepsilon < x \end{cases} \quad (30.117)$$

\mathbf{s} is updated incrementally and modified at the end of the step so that

$$|\mathbf{s}| \leq \mu d(1 + \varepsilon) \quad (30.118)$$

after the contact update.

30.23.3 Contact Nodal Forces

From the contact stress, the contact nodal forces are determined by the principle of virtual work:

$$\begin{aligned} \mathbf{f}_t^i &= \delta_i^k \left\{ \mathbf{I} + \frac{1}{2} t_t \frac{\partial \mathbf{n}_t}{\partial \mathbf{x}_k} \right\} \left\{ \mathbf{n}_t \int_{\Pi} N_t^k \sigma_n d\Pi + \mathbf{T}_t \int_{\Pi} N_t^k \sigma_t d\Pi \right\} \\ \mathbf{f}_r^j &= \delta_j^k \left\{ \mathbf{I} + \frac{1}{2} t_r \frac{\partial \mathbf{n}_r}{\partial \mathbf{x}_k} \right\} \left\{ -\mathbf{n}_r \int_{\Pi} N_r^k \sigma_n d\Pi - \mathbf{T}_r \int_{\Pi} N_r^k \sigma_t d\Pi \right\} \end{aligned} \quad (30.119)$$

where the subscripts t and r represent the tracked and reference nodal forces, respectively, and δ denotes the Kronecker delta. It is worth noting here that accounting for offsets in the kinematic description leads to a term that induces a torque due to frictional tractions. To this end, we must remark that the shell edges' kinematics do not fall into the framework presented here. The actual map between the nodal and segment coordinates is not accounted for, and contact traction, in those cases only, induces translational forces on the nodes along the edge. For beams, nodal rotations are involved in the kinematics, and frictional torques are accounted for but in a different way.

30.23.4 Treatment of beams, sharp solid edges, and shell edges

Automatic Mortar contacts support contact with the lateral surface and end tips of beams as well as edges of shell elements and sharp edges of solids. The concept of sharp, solid edges will be defined below. In this case, the abovementioned theory is applied to dummy segments corresponding to a faceted representation of the beam lateral surface and the edges of the solid and shell elements, as indicated in [Figure 30.41](#). For a beam element, the contact surface is represented by 14 faceted segments encapsulating a cylinder with the same length and volume as the beam element. This implies that all beam elements are assumed to have a circular cross-section for the contact. The edges of the shell element surface are identified assuming the contact definition (tracked or reference) consists of the entire physical component (metal sheet). We, therefore, recommend defining the contacts using part or part sets. Otherwise, false edges may be created in the interior of the component.

An edge contact element is created for shells by extruding the shell edge by a distance corresponding to the shell thickness with appropriate adjustments for irregular geometries. As mentioned above, the kinematics that create these edge segments do not account for their numerical representation. Hence, Mortar contact does not assemble the

corresponding forces in a finite-element-consistent manner. The shell edge treatment is to be seen as a simplified treatment that incorporates contact resistance for these geometries, which is probably okay in most practical situations. For beam elements, however, rotational degrees of freedom *are* included consistently, thus causing beam elements to rotate with respect to their axes when tangential friction forces are applied to the lateral surface.

A *sharp*, solid edge is detected when the angle θ between the normals of two adjacent contact segments is larger than $\pi/3$. In this case, the edge is smoothed by adding four segments between the two nodes common to the two contact segments while at the same time adjusting the size of the two main contact segments. The effect is a rounded representation of the edge and a smoother contact response. See [Figure 30.41](#). The size of the smoothing is at most 5% of the size of the smallest of the two contact segments, so the effect is reduced with mesh refinement. Because the contact area of these edge segments may be small, the stiffness of the contact for those segments is scaled by the factor $\beta = \sqrt{3} \cot\left(\frac{\pi-\theta}{2}\right)$, where θ is the initial angle between the main segments' normal vectors. This scale factor is applied to both the tracked and reference sides, meaning that if two right-angled edges come into contact, the stiffness is scaled by $\beta_t \beta_r = \sqrt{3} \cot\left(\frac{\pi-\frac{\pi}{2}}{2}\right) \sqrt{3} \cot\left(\frac{\pi-\frac{\pi}{2}}{2}\right) = 9$. Whether this is sufficient to handle most common situations is currently unknown. Hence, this design decision is subject to change in the future. If a contact segment has two sharp edges with a common node, a segment is created at the location of that node to account for contact with the corner of the solid element geometry. The stiffness scale factor for a corner node is the average of the scale factors for the connected edges.

The motivation behind solid edge smoothing is twofold. First, it adds penetration resistance between a solid edge/corner and other geometries. Second, it aids in establishing a physical contact state when solid elements slide off sharp geometrical objects. More specifically, it eliminates the ambiguity of which contact segments are in contact and presumably prevents sudden spikes in the contact force. This is also illustrated in [Figure 30.41](#).

It is important to stress that *automatic* Mortar contact surfaces are always on the *outer* geometry for both the tracked and reference sides. In other words, contact does *not* occur on the midsurface of shells. [Figure 30.42](#) illustrates a typical modeling problem that results in extremely large contact stress unless the IGNORE flag is appropriately used.

Forming contact is treated differently than automatic contact. Shell edge and beam contact are not supported. Additionally, any *shell* reference surface *must* be rigid with its segment normal vectors oriented towards the tracked side of the contact. Furthermore, the contact occurs at the midsurface on the reference side at the midsurface and on the outer geometry for the tracked side. The segment orientation of the (deformable) reference side is arbitrary, but, if it consists of shell elements, contact can only occur on one side for a given contact definition. For instance, the contact between the tool and blank and the contact between the die and blank must be defined using two different contact interface because these contacts typically occur on different sides of the blank.

Solid reference surfaces may be rigid or deformable, thus allowing for effects of tool deformation and/or cooling effects.

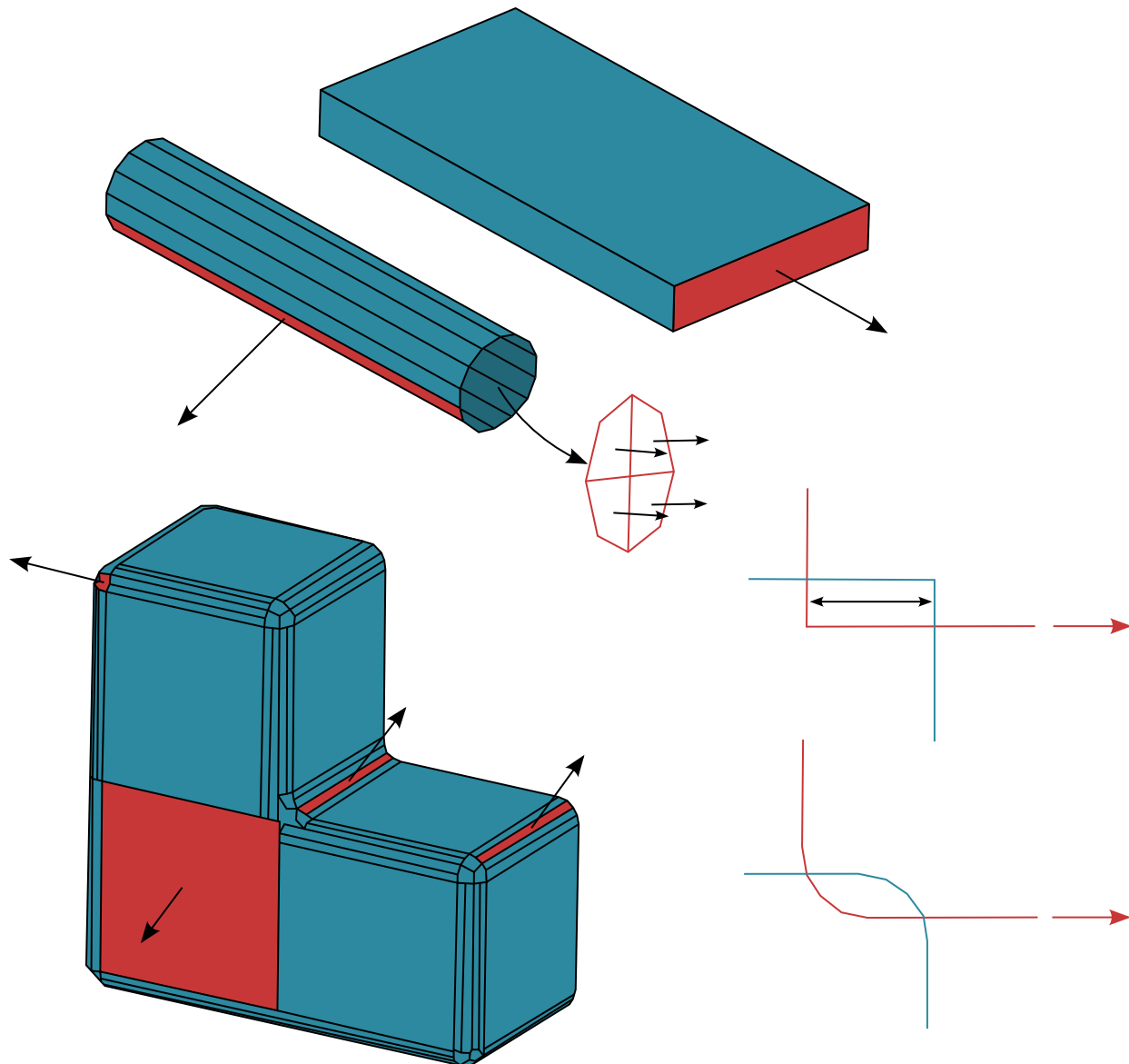


Figure 30.41. Faceted representation of beams (including four quads representing a tip at the end), shell edges, and sharp, solid edges, respectively. The segment geometry is indicated by red. The contact surface representation of three cubic solid elements is illustrated with somewhat exaggerated smoothing for illustration purposes. The effect of edge smoothing in contact is illustrated by the bottom right section cut. If the red object slides to the right and the blue object remains fixed, the sudden detection of a parasitic contact indicated by the double arrow results in a jump in the contact force. This is alleviated by smoothing of the edges below.

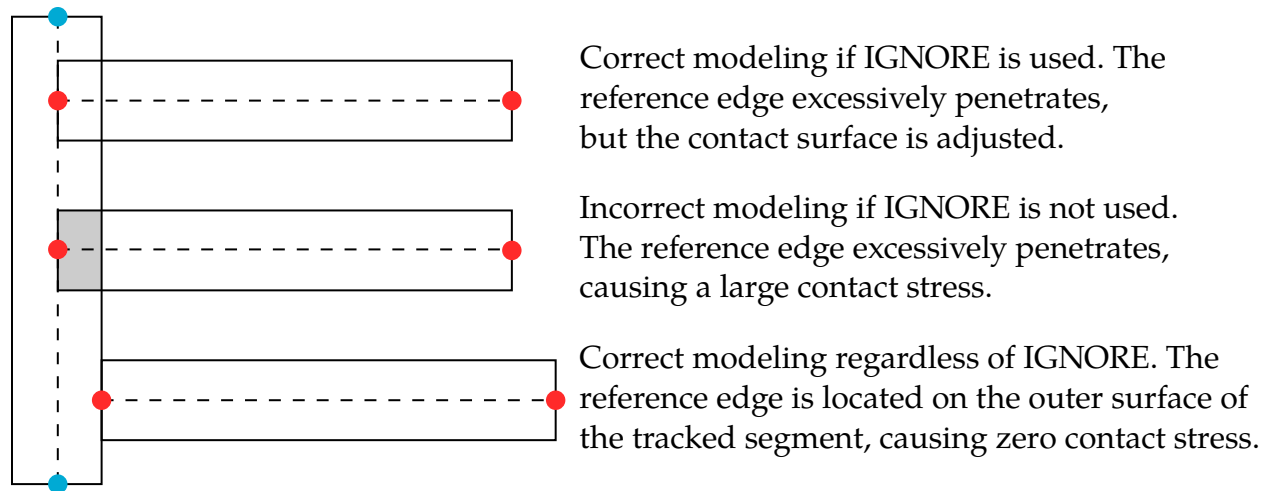


Figure 30.42. Intersected view of shells in edge-to-surface contact for automatic Mortar contact. Shell midsurfaces are indicated by dashed lines and outer surfaces by solid lines. Nodes of tracked shell elements are blue, and nodes of reference shell elements are red. The initial penetration volume is shaded in grey.

30.23.5 Characteristic Length and Contact Release

The characteristic lengths of the tracked and reference sides are denoted by d_c^t and d_c^r , respectively. They are important in Mortar contact because they affect the contact stiffness. They also determine the maximum allowed penetration, d_{\max} , between two arbitrary segments. To be more specific, two segments cannot penetrate more than 95% of the average characteristic lengths. Penetrations larger than that are not detected. In mathematical terms this can be stated as:

$$d_{\max} = 0.95 \frac{d_c^t + d_c^r}{2} \quad (30.120)$$

Figure 30.43 illustrates the effect of maximum penetration by giving the contact stress as a function of relative penetration. Increasing the stiffness parameter, SFSA, can minimize the risk of releasing contacts but may lead to worse convergence in implicit analysis. To stiffen the contact for large penetrations without affecting moderate penetrations, use the parameter IGAP. Figure 30.43 illustrates the affect of this parameter. IGAP can be used if the contact pressure is locally very high. This discussion highlights that Mortar contact has no “stick” option for improving implicit convergence. IGAP may lead to worse convergence characteristics but rules out sticky behavior and underreporting of contact forces.

The characteristic length for shells is the shell thickness. For solids it is the smallest element size in the part to which the segment belongs. The latter may lead to unrealistically high or low contact stiffness, or it may result in a too small maximum penetration depth, depending on the mesh. Setting PENMAX to the characteristic length may alleviate these issues. PENMAX should correspond to some physical member size in the model and/or adjust the contact stiffness, depending on the issue.

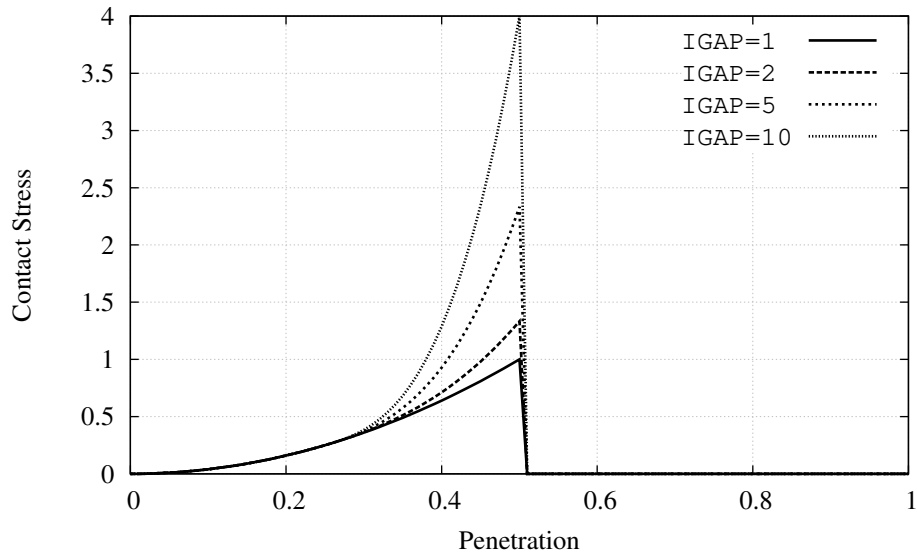


Figure 30.43. Mortar contact stress as function of penetration relative $2d_{\max}$.

30.23.6 Output for Debugging Implicit Models

In implicit analysis it is almost inevitable to run into convergence problems, especially when contacts are involved. To obtain detailed information on penetration distance and potential contact release (meaning penetration becomes too large for the contact to be detected in subsequent steps), set `MINFO = 1` on `*CONTROL_OUTPUT`. This option causes the LS-DYNA to provide information on the largest penetration, both absolute and relative, in the message files after each converged step, including a warning if penetration is close to being released. It also reports the elements with largest penetrations, making it easy to locate critical areas of the model in LS-PrePost. Contact release should be avoided. Otherwise, results may be useless.

Additionally, poor convergence can be due to a too stiff contact. Because the stiffness for a Mortar contact segment pair only depends on the tracked segment, we recommend putting weak parts on the tracked side. If a steel part is in contact with rubber, for instance, the rubber part should be on the tracked side in the contact definition. Turning on `D3ITCTL` on `*CONTROL_IMPLICIT_SOLUTION` and `RESPLT` on `*DATABASE_EXTENT_BINARY` aid in finding isolated areas in the model with poor convergence. Not rarely this is due to contacts.

30.23.7 Initial Penetrations

Initial penetrations are always reported in the message files, including the maximum penetration and how initial penetrations are to be handled. The `IGNORE` flag governs the latter and the options are:

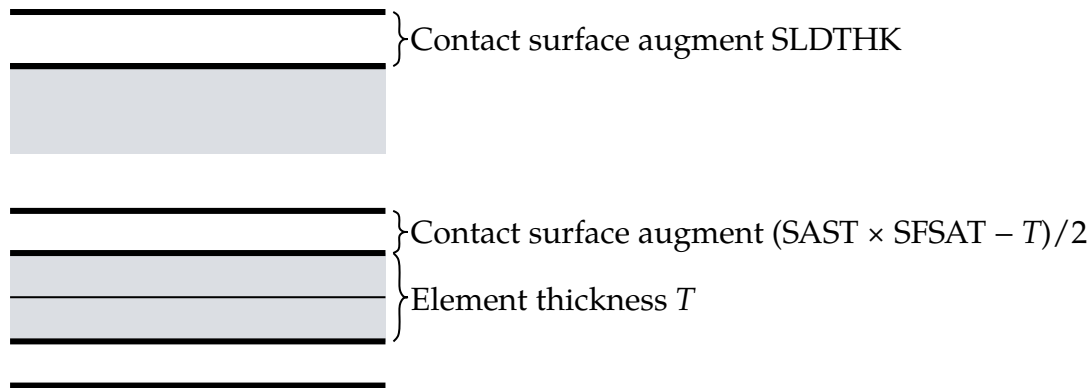


Figure 30.44. Illustration of contact surface location for automatic Mortar contact, solids on top and shells below.

- IGNORE < 0 See explanation for the corresponding positive value. The only difference is that contact between segments belonging to the same part is not treated.
- IGNORE = 1 Initial penetrations are tracked, meaning the tracked contact surface is translated to the level of the initial penetrations and subsequently follows the reference contact surface on separation until the unmodified level is reached
- IGNORE = 2 Initial penetrations are ignored, meaning the tracked contact surface is translated to the level of the initial penetrations, optionally with an initial contact stress governed by MPAR1. *This is the default option for Mortar contact.*
- IGNORE = 3 Initial penetrations are removed over time, meaning the tracked contact surface is translated to the level of the initial penetrations and pushed back to its unmodified level over a time determined by MPAR1
- IGNORE = 4 This option is the same as IGNORE = 3, but it allows for large penetrations by also setting MPAR2 to at least the maximum initial penetration.

The use of IGNORE depends on the problem. If no initial penetrations are present there is no need to use this parameter at all. If penetrations are relatively small relative to the maximum allowed penetration, IGNORE = 1 or 2 seem to be the appropriate choice. For IGNORE = 2 specifying an initial contact stress small enough to not significantly affect the physics but large enough to eliminate rigid body modes may avoid singularities in the stiffness matrix. The intention with this is to constrain loose parts that are initially close but not in contact by pushing out the contact surface using SLDTHK or SFSAT and applying IGNORE = 2. Increasing SFSAT for shells to a number larger than unity pushes the contact surface outside the geometry, leading to contact detection.

SLDTHK serves the same purpose for solid elements. See Figure 30.44. It is at least good for debugging problems with many singular rigid body modes.

IGNORE = 3 is the Mortar interference counterpart, used for instance if there is a desire to fit a rubber component in a structure or for eliminating initial penetrations by simulation. With this option, the contact surfaces are restored linearly in time from the beginning of the simulation to the time specified by MPAR1. If the intention is to eliminate initial penetrations completely, it may also in this case be of importance to use SFSAT or SLDTHK to reduce the possibility that the actual geometry is penetrated. If using a single surface definition on a complicated geometry with many parts, a negative value of IGNORE could be of interest because Mortar contact may otherwise detect spurious contacts between segments belonging to the same part.

A drawback of IGNORE = 3 is that initial penetration must be smaller than half the characteristic length of the contact. Otherwise, they are not detected in the first place. For this reason we developed IGNORE = 4, where initial penetrations may be of arbitrary size. IGNORE = 4, however, requires providing crude information on the level of penetration of the contact interface with MPAR2. MPAR2 must be larger than the maximum penetration. Otherwise, an error termination occurs. IGNORE = 4 only applies when the tracked surface is solid elements.

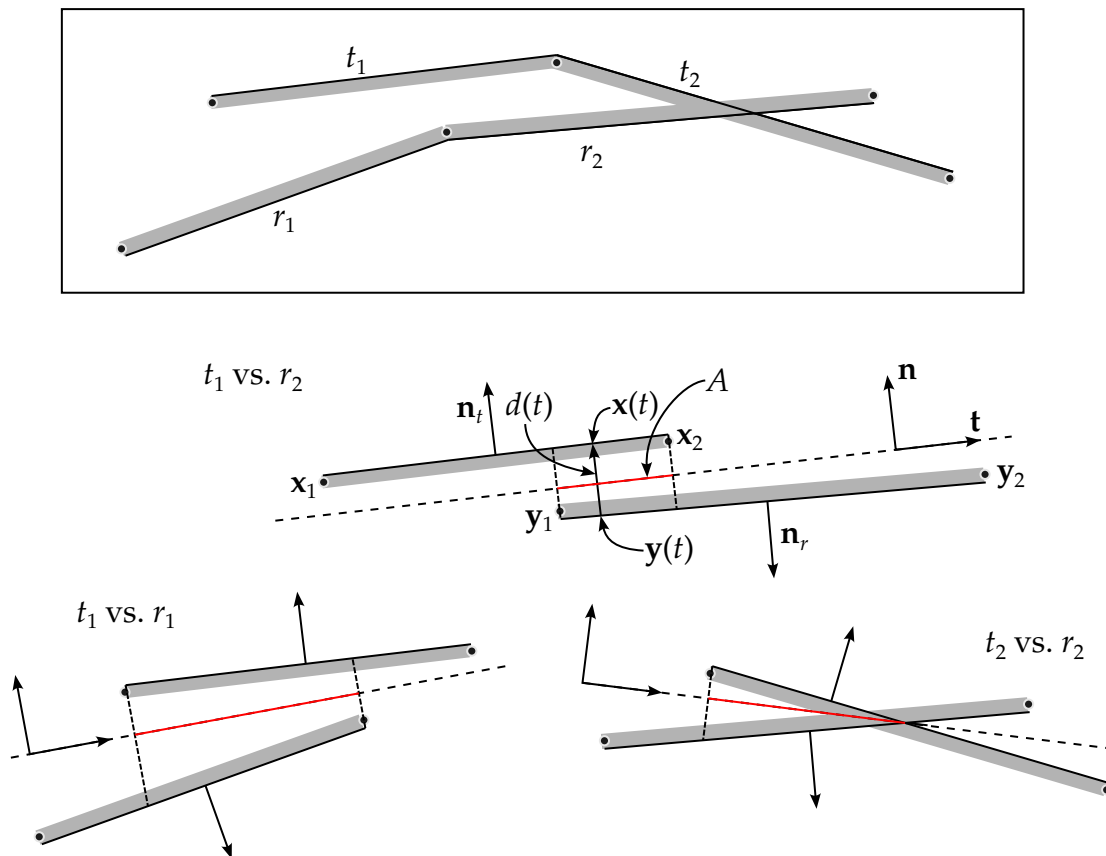


Figure 30.45. 2D Mortar contact with two tracked segments in contact with two reference segments, resulting in three separate treatments.

30.23.8 2D Mortar contact

Automatic single and surface-to-surface Mortar contacts are available in 2D. We describe 2D Mortar contact here in detail as an attempt to illustrate it in a setting that is hopefully easier to understand. [Figure 30.45](#) (top) provides an example of contact between two tracked and two reference segments. This particular configuration results in the treatment of three tracked versus reference contact pairs (bottom). For the mathematical treatment, we refer to this figure as an illustration.

Kinematics

First a common tangential direction is determined, this is given as:

$$\mathbf{t} = \frac{\mathbf{x}_2 + \mathbf{y}_2 - \mathbf{x}_1 - \mathbf{y}_1}{\|\mathbf{x}_2 + \mathbf{y}_2 - \mathbf{x}_1 - \mathbf{y}_1\|}. \quad (30.121)$$

The corresponding normal \mathbf{n} is perpendicular to \mathbf{t} , with the direction convention as illustrated. We can define a coordinate t along this tangential direction with the origin chosen arbitrarily. Then, $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are the projection coordinates along \mathbf{n} onto the tracked and reference segments, respectively. Obviously there is a finite interval $t \in [\tilde{t}_1, \tilde{t}_2]$ where both $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are well-defined. On this interval, we can define the penetration as:

$$d(t) = \mathbf{n}^T (\mathbf{x}(t) - \mathbf{y}(t)). \quad (30.122)$$

The overlapped interval is then further reduced to $t \in [t_1, t_2]$ for which $d(t) \geq 0$. Note that for the t_2 versus r_2 situation, these two intervals are not the same. The interval $[t_1, t_2]$ is indicated by *red* in the illustration.

In the tangential direction we need to define the kinematics for sliding. We do this by associating a history variable to each tracked segment. Let S be a weighted measure of the distance a tracked segment has slid along the reference surface. It is defined as:

$$S = S_{n-1} + \sum_i \int_{A_i} \{s(t) - \mathbf{t}_{n-1}^T (\mathbf{x}_{n-1}(t) - \mathbf{y}_{n-1}(t))\} dA_i, \quad (30.123)$$

where $s(t) = \mathbf{t}^T (\mathbf{x}(t) - \mathbf{y}(t))$ and the subscript $n - 1$ refers to the corresponding value in the previous step. The integral is taken over the domain A_i of the intersected interval between tracked segment and reference segment i , accounting for plane strain or axial symmetry. Note that the tracked segment can be in contact with several reference segments, hence the sum. Further note that by construction the first term in the integrand, $s(t)$, is zero. Thus, we can simplify this to:

$$S = S_{n-1} - \sum_i \int_{A_i} \mathbf{t}_{n-1}^T (\mathbf{x}_{n-1}(t) - \mathbf{y}_{n-1}(t)) dA_i. \quad (30.124)$$

In the illustrated situation, tracked segment 1 has sliding contributions from reference segments 1 and 2, while tracked segment 2 only has a contribution from reference segment 2.

Likewise, we define a weighted penetration:

$$D = \sum_i \int_{A_i} d(t) dA_i. \quad (30.125)$$

The weighted penetration is used together with S to define the friction law described below.

Constitutive relations

For simplicity, we drop the explicit dependence on t from now on and define the contact stress (pressure) as:

$$\sigma_n = Kd^2. \quad (30.126)$$

In the above, K is the contact stiffness defined as:

$$K = \frac{\alpha}{0.01} \frac{1}{T_t T_r} \frac{2K_t K_r}{K_t + K_r} f(\mathbf{n}_t^T \mathbf{n}_r). \quad (30.127)$$

Here, $\alpha = \text{PSF} \times \text{SLSFAC}$ is a stiffness scale factor, and $T_{t/r}$ and $K_{t/r}$ are characteristic lengths and material stiffnesses of the tracked and reference segments, respectively. The function f is used to linearly reduce stiffness for segments that are not parallel and is given by:

$$f(x) = \begin{cases} 0 & -\frac{1}{2} \leq x \\ \frac{1+2x}{1-\sqrt{3}} & -\frac{\sqrt{3}}{2} \leq x < -\frac{1}{2} \\ 1 & x < -\frac{\sqrt{3}}{2} \end{cases} \quad (30.128)$$

Note that \mathbf{n}_t and \mathbf{n}_r are the normal vectors to the tracked and reference segments.

The friction stress is defined as:

$$\sigma_t = \mu \sigma_n g\left(\frac{S}{\mu D}\right), \quad (30.129)$$

with μ being the Coulomb friction coefficient and g a continuously differentiable function defined as:

$$g(x) = \begin{cases} 1 & 1.03 \leq x \\ 1 - \frac{25}{3}(x - 1.03)^2 & 0.97 \leq x < 1.03 \\ x & -0.97 \leq x < 0.97 \\ -1 + \frac{25}{3}(x + 1.03)^2 & -1.03 \leq x < -0.97 \\ -1 & -1.03 > x \end{cases} \quad (30.130)$$

Thus, the magnitude of friction stress, σ_t , is at most $\mu \sigma_n$, and the fraction thereof is determined by the appropriate relation between the accumulated sliding and penetration. Upon convergence, S is updated for the next step according to $S_n = \max(-1.03\mu D, \min(1.03\mu D, S))$ to yield proper friction behavior.

Nodal forces

The nodal force contribution from a given segment pair is determined from the principle of virtual work, i.e.,

$$\delta W = \int_A \sigma_n \delta d \, dA + \int_A \sigma_t \delta s \, dA. \quad (30.131)$$

Here δ is the variation operator, and the independent variables subject to this variation are the nodal coordinates. Thus, replacing the left hand side with the expression involving nodal forces and coordinates, and using [Equations \(30.126\)](#) and [\(30.129\)](#) for the right-hand side, we obtain:

$$\sum_I \mathbf{f}_I^T \delta \mathbf{x}_I = K \int_A d^2 \delta d \, dA + \mu K g \int_A d^2 \delta s \, dA. \quad (30.132)$$

The nodal forces are then readily identified as:

$$f_I = K \int_A d^2 \frac{\partial d}{\partial x_I} \, dA + \mu K g \int_A d^2 \frac{\partial s}{\partial x_I} \, dA, \quad (30.133)$$

where the exact expressions for the integrals in terms of the nodal coordinates are, although straightforward to derive, a bit lengthy, and therefore omitted.

Stiffness matrix

The stiffness matrix is the second variation of the virtual work expression [Equation \(30.131\)](#), where we neglect the variation of the overlapped area and assume the variation can be taken on the integrand directly

$$\begin{aligned} \Delta \delta W_I = & 2K \int_A d \, \Delta d \, \delta d \, dA + 2\mu K g \int_A d \, \Delta d \, \delta s \, dA \\ & + \mu \frac{\partial g}{\partial x} \left\{ \frac{1}{\mu D} \Delta S - \frac{S}{\mu D^2} \Delta D \right\} \int_A \sigma_n \delta s \, dA. \end{aligned} \quad (30.134)$$

Here Δ is again the variation operator. We use different notation to distinguish it from δ , but it performs the same function. At this point we emphasize that the (geometric) terms involving $\Delta \delta d$ and $\Delta \delta s$ have been deliberately excluded as they, albeit being symmetric, contribute to the indefiniteness of the tangent matrix.

In [Equation \(30.134\)](#), the first term on the right-hand side is the normal-normal interaction and is nicely symmetric. The remaining terms come from friction (normal-tangent and tangent-tangent interaction) and need to be symmetrized and further simplified. We neglect any terms involving $\Delta d \delta s$ and $\Delta D \delta s$, which means that the only terms we need to deal with involve $\Delta S \delta s$. The simplifications made are to approximate:

$$\sigma_n \approx \frac{1}{A} \int_A \sigma_n \, dA \quad (30.135)$$

in the last integral, and then neglect all terms in ΔS that do not pertain to the present tracked-reference segment pair. This results in:

$$\Delta \mathbf{x}_I^T \mathbf{K}_{IJ}^T \delta \mathbf{x}_J \approx 2K \int_A d \, \Delta d \, \delta d \, dA + \frac{\partial g}{\partial x} \frac{1}{D} \left\{ \frac{1}{A} \int_A \sigma_n \, dA \right\} \left\{ \int_A \Delta s \, dA \right\} \left\{ \int_A \delta s \, dA \right\}. \quad (30.136)$$

The stiffness matrix can be identified as:

$$\mathbf{K}_{IJ} = 2K \int_A d \frac{\partial d}{\partial \mathbf{x}_I} \frac{\partial d}{\partial \mathbf{x}_J} dA + \frac{\partial g}{\partial x} \frac{1}{D} \left\{ \frac{1}{A} \int_A \sigma_n dA \right\} \left\{ \int_A \frac{\partial s}{\partial \mathbf{x}_I} dA \right\} \left\{ \int_A \frac{\partial s}{\partial \mathbf{x}_J} dA \right\}. \quad (30.137)$$

A characteristic feature of the Mortar contact, which can be deduced from the above expression, is that not only the nodal forces are continuous but also the stiffness matrix. This follows from the C^1 continuity of g , meaning that all involved functions above are continuous, and that \mathbf{K}_{IJ} tends to zero as either d or A tends to zero. A mathematical treatment yields

$$\|\mathbf{K}_{IJ}\| \leq KA \max_t d(t) \left\{ 2 \max_t \left\| \frac{\partial d(t)}{\partial \mathbf{x}_I} \right\| \left\| \frac{\partial d(t)}{\partial \mathbf{x}_J} \right\| + \max_t \left\| \frac{\partial s(t)}{\partial \mathbf{x}_I} \right\| \left\| \frac{\partial s(t)}{\partial \mathbf{x}_J} \right\| \right\}, \quad (30.138)$$

which tends to zero as d or A tends to zero as the terms inside the right bracket are bounded.

30.23.9 Tied and tiebreak option

A tied and tiebreak option with Mortar contact is available by appending MORTAR_TIED or TIEBREAK_MORTAR to the automatic surface-to-surface contact keyword. In principle, the tiebreak allows for specifying the contact stress, σ , as a function of the separation, d , to accomodate a given interface law that includes softening and failure (loss of interface stiffness). The law incorporates both normal and tangential directions, usually denoted mode I and mode II in the delamination community. The typical appearance of such a law is shown in [Figure 6.2.](#) The law is in general characterized by the *energy release rate*, E , which is the energy per unit area required to release the contact, and the *maximum interface stress*, σ_e , which is the peak contact stress before softening begins. Referring to [Figure 6.2.](#), the energy release rate is the integral of the curve describing the stress as a function of displacement. The law can indeed be quite complicated, and we refer to the keyword manual for more information regarding details of the cohesive models available. *For the Mortar tiebreak option only, OPTION = 7 and OPTION = 9 are supported.*

The internal treatment of the Mortar tied and tiebreak options are very similar to that sliding contacts, but the following remarks describe specific differences and special considerations:

- Tied Mortar contact is penalty-based and exhibits a linear relationship between the normal/tangential separation and resulting contact stress. Referring to [Equation \(30.114\)](#), $f(x) = x$ for all x in all separation directions. Since the nonlinearity is significantly reduced compared to sliding contacts, the implicit convergence characteristics are insensitive to scaling of the contact stiffness.
- Tiebreak Mortar contact is treated similarly, but the normal and tangential contact stress is given by the constitutive model. It is strongly recommended to set CN (OPTION = 7 and OPTION = 9) and CT2CN (only OPTION = 9) on the additional card associated with the tiebreak option. Otherwise the interface stiffness is determined internally and may be inconveniently large.
- Tiebreak Mortar contact is superposed by a sliding contact to take compressive contact stress, mainly to prevent penetrations in the post-failure regime but also

to prevent cohesive failure due to normal compression. This contact follows the theory described above and automatically applies the IGNORE = 2 option to start with an initial zero contact stress.

- The sliding contact associated with the tiebreak option is frictionless as long as the tied contact exists. This treatment is to avoid spurious interactions between the laws (mode II and friction) in the tangential directions. The friction is activated as soon as the tied contact is released. Furthermore, the tied interface will not take compressive stress as this is lent to the sliding contact.
- Tied and tiebreak Mortar contacts are applied if the initial normal distance between the tracked and reference segments (with respect to their outer geometries) is less than a critical distance, $d_t = 0.05d_c$. In this formula d_c is the characteristic length as described above. Here $d_t = \text{PENMAX}$ can be used to override this

tolerance for when to tie two segments, so the meaning of PENMAX is in this case

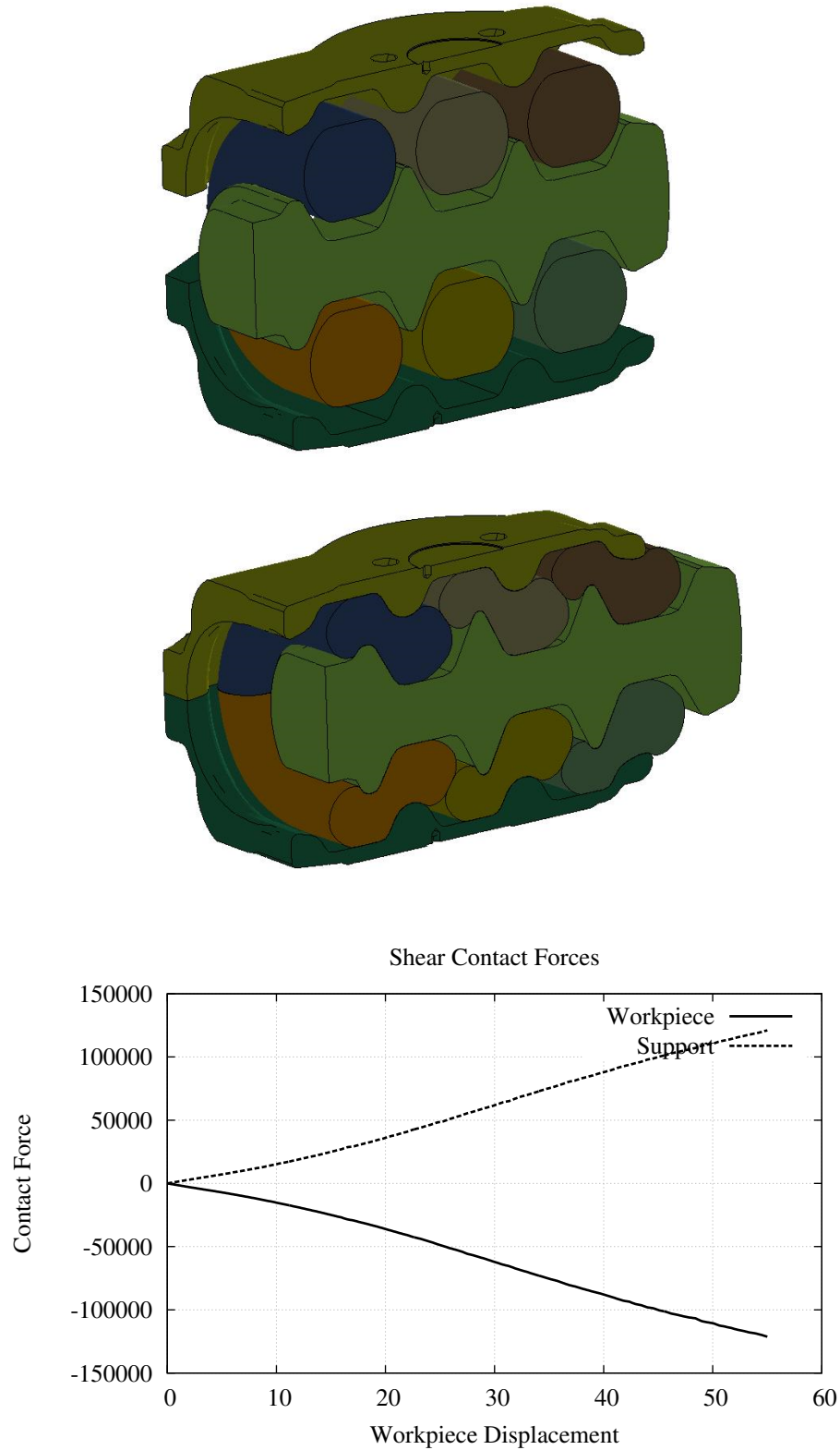


Figure 30.46. A rubber compression example solved in implicit with Mortar contact (Courtesy of Dellner Couplers AB). The graph shows the contact force between the rubber parts and the moving workpiece and between the rubber parts and the two supports, respectively.

31

Geometric Contact Entities

Contact algorithms in LS-DYNA work for any arbitrarily shaped surface by representing the surface with a faceted mesh. We can treat occupant modeling this way by using fine meshes to represent the head or knees. The generality of the faceted mesh contact suffers drawbacks when modeling occupants, however, due to storage requirements, computing costs, and mesh generation times. We added geometric contact entities as an alternate method to model cases of curved rigid bodies impacting deformable surfaces. Geometric contact entities require much less storage, and the computational cost decreases dramatically compared to more general contact. To enable contact with contact entities, use `*CONTACT_ENTITY` or `*CONTACT_GEBOD`,

We developed geometric contact entities using a standard solids modeling approach. A scalar function, $G(x, y, z)$, defines the geometric entity. The scalar function determines the solid as follows:

$$\begin{aligned} G(x, y, z) > 0 &\rightarrow \text{the point } (x, y, z) \text{ is outside the solid.} \\ G(x, y, z) = 0 &\rightarrow \text{the point } (x, y, z) \text{ is on the surface of the solid.} \\ G(x, y, z) < 0 &\rightarrow \text{the point } (x, y, z) \text{ is inside the solid.} \end{aligned} \quad (31.1)$$

Thus, by a simple function evaluation, a node can be immediately determined to be outside the solid or in contact. [Figure 31.1](#) illustrates this for a cylinder. The following equation determines the surface of an infinite cylinder:

$$G(x, y) = x^2 + y^2 - R^2, \quad (31.2)$$

If the node is in contact with the solid, a restoring force must be applied to eliminate further penetration. Several methods are available to do this, such as Lagrange multipliers or momentum-based methods. We selected the penalty method because it is the simplest and most efficient. Also, in our applications, the impact velocities are at a level where the penalty method provides answers almost identical to the exact solution.

Using the penalty method, the restoring force is proportional to the penetration distance into the solid and acts in the direction normal to the surface of the solid. Thus,

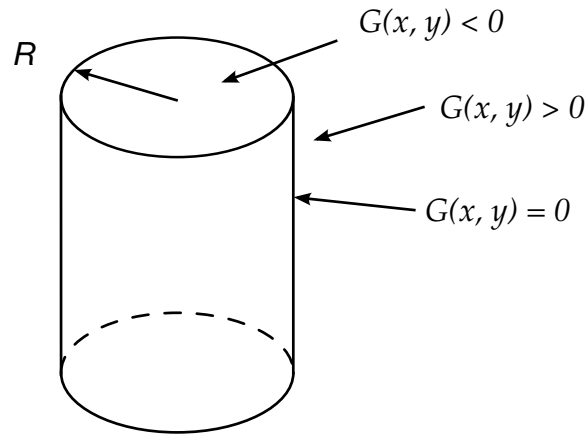


Figure 31.1. Determination of whether a node is interior or exterior to the cylindrical surface

the penetration distance and the normal vector must be determined. The gradient of the scalar function conveniently determines the surface normal vector:

$$\mathbf{N}(x, y, z) = \frac{\frac{\partial G}{\partial x} \mathbf{i} + \frac{\partial G}{\partial y} \mathbf{j} + \frac{\partial G}{\partial z} \mathbf{k}}{\sqrt{\left(\frac{\partial G}{\partial x}\right)^2 + \left(\frac{\partial G}{\partial y}\right)^2 + \left(\frac{\partial G}{\partial z}\right)^2}} , \quad (31.3)$$

for all (x, y, z) such that $G(x, y, z) = 0$. The definition of $G(x, y, z)$ guarantees that this vector faces in the outward direction. When penetration does occur, the function $G(x, y, z)$ will be slightly less than zero. For curved surfaces, this will result in some errors in calculating the normal vector because it is not evaluated exactly at the surface. In an implicit code, this would be important. However, the explicit time integration scheme in LS-DYNA uses such a small time step that penetrations are negligible, and the normal function can be evaluated directly at the tracked node, ignoring any penetration.

The penetration distance is the last item to be calculated. In general, the following equation determines the penetration distance, d :

$$d = |\mathbf{X}_n - \mathbf{X}'_n| , \quad (31.4)$$

where \mathbf{X}_n is the location of node n and \mathbf{X}'_n is the nearest point on the surface of the solid. To determine \mathbf{X}'_n , we define a line function that passes through \mathbf{X}_n and is normal to the surface of the solid:

$$\mathbf{L}(s) = \mathbf{X}_n + s\mathbf{N}(\mathbf{X}_n) . \quad (31.5)$$

Substituting the line function into the surface equation of a solid body gives:

$$G(\mathbf{X}_n + s\mathbf{N}(\mathbf{X}_n)) = 0 . \quad (31.6)$$

If Equation (31.6) has only one solution, this root provides the parametric coordinate s which locates \mathbf{X}'_n . If Equation (31.6) has more than one root, then the root which minimizes Equation (31.4) locates the point \mathbf{X}'_n .

We define the restoring forces as:

$$\mathbf{f} = p d\mathbf{N}(\bar{\mathbf{X}}'_n) , \quad (31.7)$$

where p is a penalty factor. p is effectively a spring constant. To minimize the penetration of the tracked node into the solid, the constant p is set to a large value. However, it should not be set so large that the Courant stability criterion is violated. This criterion for the tracked node tells us that:

$$\Delta t \leq \frac{2}{\omega_{\max}} = \frac{2}{\sqrt{\frac{K_n}{m_n}}} = 2\sqrt{\frac{m_n}{K_n}} , \quad (31.8)$$

where K_n is the stiffness of node n and m_n is the mass of node n .

We determine the penalty factor, p , by choosing a value which results in a penalty/tracked mass oscillator that has a characteristic time step that is ten times larger than the Courant time step:

$$10\Delta t = 2\sqrt{\frac{m_n}{p_n}} . \quad (31.9)$$

Solving for p_n gives:

$$p_n = \frac{4m_n}{(100\Delta t^2)} . \quad (31.10)$$

Inclusion of any structural elements into the occupant model typically results in very large stiffnesses due to the small time step and the $(1/\Delta t)^2$ term. Thus, the method is highly effective even with impact velocities on the order of 1km/s.

The scalar function $G(\mathbf{X})$ is frequently more conveniently expressed as $g(\mathbf{x})$, where g is the function defined in local coordinates and \mathbf{x} is the position in local coordinates. We relate the local entity to the global coordinates by:

$$\mathbf{x} = [\mathbf{T}](\mathbf{X}_j - \mathbf{Q}_j) , \quad (31.11)$$

where \mathbf{Q}_j is the offset and $[\mathbf{T}]$ is a rotation matrix. The solid scalar function and the penetration distance can be evaluated in either local or global coordinates with no difference to the results. When working in local coordinates, the gradient of the local scalar function provides a normal vector which is in the local system and must be transformed into the global by:

$$\mathbf{N}(\bar{\mathbf{X}}) = [\mathbf{T}]^T \mathbf{n}(\mathbf{x}) . \quad (31.12)$$

For example, the following function defines an ellipsoid:

$$G(x, y, z) = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 . \quad (31.13)$$

The gradient of G is $\frac{2x}{a^2}\mathbf{i} + \frac{2y}{b^2}\mathbf{j} + \frac{2z}{c^2}\mathbf{k}$ and the normal vector is:

$$\mathbf{n}(x, y, z) = \frac{\left(\frac{x}{a^2}\mathbf{i} + \frac{y}{b^2}\mathbf{j} + \frac{z}{c^2}\mathbf{k}\right)}{\sqrt{\frac{x^2}{a^4} + \frac{y^2}{b^4} + \frac{z^2}{c^4}}} , \quad (31.14)$$

Substituting [Equations \(31.5\)](#) and [\(31.13\)](#) into the second equation of [Equation \(31.1\)](#) gives:

$$\begin{aligned} \left[\left(\frac{n_x}{a}\right)^2 + \left(\frac{n_y}{b}\right)^2 + \left(\frac{n_z}{c}\right)^2 \right] s^2 + 2 \left[\frac{n_x x_n}{a^2} + \frac{n_y y_n}{b^2} + \frac{n_z z_n}{c^2} \right] s \\ + \left[\left(\frac{x_n}{a}\right)^2 + \left(\frac{y_n}{b}\right)^2 + \left(\frac{z_n}{c}\right)^2 - 1 \right] = 0 . \end{aligned} \quad (31.15)$$

Solving this quadratic equation for s provides the intercepts for the nearest point on the ellipsoid and the opposite point of the ellipsoid, which the normal vector, \mathbf{X}_n , also points toward.

We have implemented contact entities for planes, cylinders, spheres, ellipsoids, and user-defined axisymmetric surfaces with appropriate simplifications (see *CONTACT_ENTITY). We intended the ellipsoids to be used with rigid body dummy models and have a special contact card, *CONTACT_GEBOD, specifically for GEBOD dummies. The methods are, however, quite general, so we could implement many more shapes.

32

Nodal Constraints

In this section nodal constraints and linear constraint equations are described.

32.1 Nodal Constraint Sets

This option forces groups of nodes to move together with a common translational acceleration in either one or more degrees of freedom. The implementation is straightforward with the common acceleration defined by

$$a_{i_{\text{common}}} = \frac{\sum_j^n M_j a_i^j}{\sum_j^n M_j}, \quad (32.1)$$

where n is the number of nodes, a_i^j is the acceleration of the j th constrained node in the i th direction, and $a_{i_{\text{common}}}$ is the common acceleration.

Nodal constraint sets eliminate rigid body rotations in the body that contains the node set and, therefore, must be applied very cautiously.

32.2 Linear Constraint Equations

Linear constraint equations of the form:

$$\sum_{k=1}^n C_k u_k = C_0, \quad (32.2)$$

can be defined where n is the number of constrained degrees of freedom, u_k is a constrained nodal displacement, and the C_k are user-defined coefficients. Unless LS-DYNA

is initialized by linking to an implicit code to satisfy this equation at the beginning of the calculation, the constant C_0 is assumed to be zero. The first constrained degree of freedom is eliminated from the equations of motion:

$$u_1 = C_0 - \sum_{k=2}^n \frac{C_k}{C_1} u_k. \quad (32.3)$$

Its velocities and accelerations are given by

$$\begin{aligned} \dot{u}_1 &= - \sum_{k=2}^n \frac{C_k}{C_1} \dot{u}_k, \\ \ddot{u}_1 &= - \sum_{k=2}^n \frac{C_k}{C_1} \ddot{u}_k \end{aligned} \quad (32.4)$$

respectively. In the implementation a transformation matrix \mathbf{L} is constructed relating the unconstrained u and constrained $u_{\text{constrained}}$ degrees of freedom. The constrained accelerations used in the above equation are given by:

$$u_{\text{constrained}} = [\mathbf{L}^T \mathbf{M} \mathbf{L}]^{-1} \mathbf{L}^T \mathbf{F}, \quad (32.5)$$

where \mathbf{M} is the diagonal lumped mass matrix and \mathbf{F} is the righthand side force vector. This requires the inversion of the condensed mass matrix which is equal in size to the number of constrained degrees of freedom minus one. The inverse of the condensed mass matrix is computed in the initialization phase and stored in core.

33

Vectorization and Parallelization

33.1 Vectorization

In 1978, when the author first vectorized DYNA3D on the CRAY-1, a four-fold increase in speed was attained. This increase was realized by recoding the solution phase to process vectors in place of scalars. It was necessary to process elements in groups rather than individually as had been done earlier on the CDC-7600 supercomputers.

Since vector registers are generally some multiple of 64 words, vector lengths of 64 or some multiple are appropriate. In LS-DYNA, groups of 128 elements or possibly some larger integer multiple of 64 are utilized. Larger groups give a marginally faster code, but can reduce computer time sharing efficiency because of increased core requirements. If elements within the group reference more than one material model, subgroups are formed for consecutive elements that reference the same model. LS-DYNA internally sorts elements by material to maximize vector lengths.

Conceptually, vectorization is straightforward. Each scalar operation that is normally executed once for one element, is repeated for each element in the group. This means that each scalar is replaced by an array, and the operation is put into a DO-loop. For example, the nodal force calculation for the hexahedron element appeared in a scalar version of DYNA3D as:

```

E11=SGV1*PX1+SGV4*PY1+SGV6*PZ1
E21=SGV2*PY1+SGV4*PX1+SGV5*PZ1
E31=SGV3*PZ1+SGV6*PX1+SGV5*PY1
E12=SGV1*PX2+SGV4*PY2+SGV6*PZ2
E22=SGV2*PY2+SGV4*PX2+SGV5*PZ2
E32=SGV3*PZ2+SGV6*PX2+SGV5*PY2
E13=SGV1*PX3+SGV4*PY3+SGV6*PZ3
E23=SGV2*PY3+SGV4*PX3+SGV5*PZ3
E33=SGV3*PZ3+SGV6*PX3+SGV5*PY3
E14=SGV1*PX4+SGV4*PY4+SGV6*PZ4

```

$$\begin{aligned} E24 &= SGV2 * PY4 + SGV4 * PX4 + SGV5 * PZ4 \\ E34 &= SGV3 * PZ4 + SGV6 * PX4 + SGV5 * PY4 \end{aligned}$$

and in the vectorized version as:

```
DO 110 I = LFT, LLT
  E11(I) = SGV1(I)*PX1(I) + SGV4(I)*PY1(I) + SGV6(I)*PZ1(I)
  E21(I) = SGV2(I)*PY1(I) + SGV4(I)*PX1(I) + SGV5(I)*PZ1(I)
  E31(I) = SGV3(I)*PZ1(I) + SGV6(I)*PX1(I) + SGV5(I)*PY1(I)
  E12(I) = SGV1(I)*PX2(I) + SGV4(I)*PY2(I) + SGV6(I)*PZ2(I)
  E22(I) = SGV2(I)*PY2(I) + SGV4(I)*PX2(I) + SGV5(I)*PZ2(I)
  E32(I) = SGV3(I)*PZ2(I) + SGV6(I)*PX2(I) + SGV5(I)*PY2(I)
  E13(I) = SGV1(I)*PX3(I) + SGV4(I)*PY3(I) + SGV6(I)*PZ3(I)
  E23(I) = SGV2(I)*PY3(I) + SGV4(I)*PX3(I) + SGV5(I)*PZ3(I)
  E33(I) = SGV3(I)*PZ3(I) + SGV6(I)*PX3(I) + SGV5(I)*PY3(I)
  E14(I) = SGV1(I)*PX4(I) + SGV4(I)*PY4(I) + SGV6(I)*PZ4(I)
  E24(I) = SGV2(I)*PY4(I) + SGV4(I)*PX4(I) + SGV5(I)*PZ4(I)
110  E34(I) = SGV3(I)*PZ4(I) + SGV6(I)*PX4(I) + SGV5(I)*PY4(I)
```

where $1 \leq LFT \leq LLT \leq n$. Elements LFT to LLT inclusive use the same material model and n is an integer multiple of 64.

Gather operations are vectorized on most supercomputers. In the gather operation, variables needed for processing the element group are pulled from global arrays into local vectors. For example, the gather operation:

```
DO 10 I      =      LFT, LLT
X1(I)  =      X(1,IX1(I))
Y1(I)  =      X(2,IX1(I))
Z1(I)  =      X(3,IX1(I))
VX1(I) =      V(1,IX1(I))
VY1(I) =      V(2,IX1(I))
VZ1(I) =      V(3,IX1(I))
X2(I)  =      X(1,IX2(I))
Y2(I)  =      X(2,IX2(I))
Z2(I)  =      X(3,IX2(I))
VX2(I) =      V(1,IX2(I))
VY2(I) =      V(2,IX2(I))
VZ2(I) =      V(3,IX2(I))
X3(I)  =      X(1,IX3(I))
X3(I)  =      X(2,IX3(I))
X3(I)  =      X(3,IX3(I))
X8(I)  =      X(1,IX8(I))
Y8(I)  =      X(2,IX8(I))
Z8(I)  =      X(3,IX8(I))
VX8(I) =      V(1,IX8(I))
VY8(I) =      V(2,IX8(I))
```

```
10    VZ8(I) =      V(3,IX8(I))
```

initializes the nodal velocity and coordinate vector for each element in the subgroup LFT to LLT. In the scatter operation, element nodal forces are added to the global force vector. The force assembly does not vectorize unless special care is taken as described below.

In general, the element force assembly is given in FORTRAN by:

```
      DO 30 I = 1,NODFRC
      DO 20 N = 1,NUMNOD
      DO 10 L = LFT,LLT
      RHS(I,IX(N,L))=RHS(I,IX(N,L))+FORCE(I,N,L)
10    CONTINUE
20    CONTINUE
30    CONTINUE
```

where NODFRC is the number of force components per node (3 for solid elements, 6 for shells), LFT and LLT span the number of elements in the vector block, NUMNOD is the number of nodes defining the element, FORCE contains the force components of the individual elements, and RHS is the global force vector. This loop does not vectorize since the possibility exists that more than one element may contribute force to the same node. FORTRAN vector compilers recognize this and will vectorize only if directives are added to the source code. If all elements in the loop bounded by the limits LFT and LLT are disjoint, the compiler directives can be safely added. We therefore attempt to sort the elements as shown in [Figure 33.1](#) to guarantee disjointness.

ELEMENT BLOCKING FOR VECTORIZATION

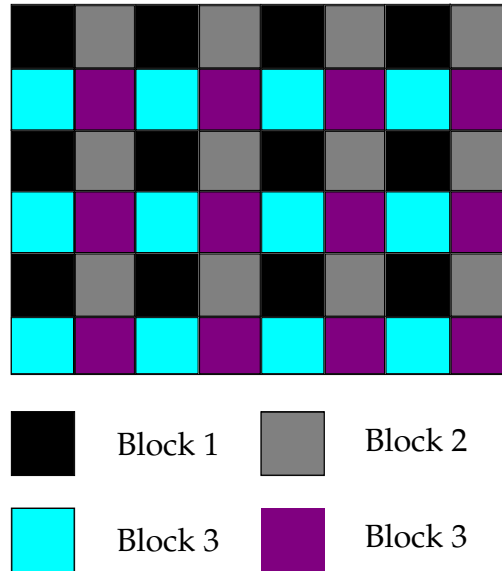


Figure 33.1. Group of 48 elements broken into 4 disjoint blocks.

The current implementation was strongly motivated by Benson [1989] and by work performed at General Motors [Ginsberg and Johnson 1988, Ginsberg and Katnik 1989], where it was shown that substantial improvements in execution speed could be realized by blocking the elements in the force assembly. Katnik implemented element sorting in a public domain version of DYNA3D for the Belytschko-Tsay shell element and added compiler directives to force vectorization of the scatter operations associated with the addition of element forces into the global force vector. The sorting was performed immediately after the elements were read in so that subsequent references to the stored element data were sequential. Benson performed the sorting in the element loops via indirect addressing. In LS-DYNA the published GM approach is taken.

Implementation of the vectorization of the scatter operations is implemented in for all elements including the solid, shell, membrane, beam, and truss elements. The sorting is completely transparent to the user.

33.2 Parallelization

In parallelization, the biggest hurdle is overcoming Amdahl's law for multitasking [Cray Research Inc. 1990]

$$S_m = \frac{1}{f_s + \frac{f_p}{N}}, \quad (33.1)$$

where

S_m = maximum expected speedup from multitasking
 N = number of processors available for parallel execution
 f_p = fraction of a program that can execute in parallel
 f_s = fraction of a program that is serial

Table 29.1 shows that to obtain a speed factor of four on eight processors it is necessary to have eighty-six percent of the job running in parallel. Obviously, to gain the highest speed factors the entire code must run in parallel.

LS-DYNA has been substantially written to function on all shared memory parallel machine architectures. Generally, shared memory parallel speed-ups of 5 on 8 processors are possible but this is affected by the machine characteristics. We have observed speeds of 5.6 on full car crash models on a machine of one manufacturer only to see a speed-up of 3.5 on a different machine of another manufacturer.

%	N = 2	N = 4	N = 8	N = 16	N = 32	N = 64	N = 128	N = 256
86.0%	1.75	2.82	4.04	5.16	5.99	6.52	6.82	6.98
90.0%	1.82	3.08	4.71	6.40	7.80	8.77	9.34	9.66
92.0%	1.85	3.23	5.13	7.27	9.20	10.60	11.47	11.96
94.0%	1.89	3.39	5.63	8.42	11.19	13.39	14.85	15.71
96.0%	1.92	3.57	6.25	10.00	14.29	18.18	21.05	22.86
98.0%	1.96	3.77	7.02	12.31	19.75	28.32	36.16	41.97
99.0%	1.98	3.88	7.48	13.91	24.43	39.26	56.39	72.11
99.2%	1.98	3.91	7.58	14.29	25.64	42.55	63.49	84.21
99.4%	1.99	3.93	7.68	14.68	26.98	46.44	72.64	101.19
99.6%	1.99	3.95	7.78	15.09	28.47	51.12	84.88	126.73
99.7%	1.99	3.96	7.84	15.31	29.28	53.83	92.69	145.04
99.8%	2.00	3.98	7.89	15.53	30.13	56.84	102.07	169.54
99.9%	2.00	3.99	7.94	15.76	31.04	60.21	113.58	203.98
100.0%	2.00	4.00	8.00	16.00	32.00	64.00	128.00	256.00

Table 29.1.Maximum theoretical speedup S_m , on N CPUs with parallelism [Cray Research Inc. 1990].

In the element loops element blocks with vector lengths of 64 or some multiple are assembled and sent to separate processors. All elements are processed in parallel. On the average a speed factor of 7.8 has been attained in each element class corresponding to 99.7% parallelization.

A significant complication in parallelizing code is that a variable can sometimes be updated simultaneously by different processors with incorrect results. To force the processors to access and update the variable in a sequential manner, a GUARD compiler directive must be introduced. This results in an interruption to the parallel execution and

can create a bottleneck. By sorting the data in the parallel groups or by allocating additional storage it is usually possible to eradicate the GUARDS from the coding. The effort may not be worth the gains in execution speed.

The element blocks are defined at the highest level and each processor updates the entire block including the right hand side force assembly. The user currently has two options: GUARD compiler directives prevent simultaneous updates of the RHS vector (recommended for single CPU processors or when running in a single CPU mode on a multi-processor), or assemble the right hand side in parallel and let LS-DYNA prevent conflicts between CPU's. This usually provides the highest speed and is recommended, i.e., no GUARDS.

When executing LS-DYNA in parallel, the order of operations will vary from run to run. This variation will lead to slightly different numerical results due to round-off errors. By the time the calculation reaches completion variations in nodal accelerations and sometimes even velocities are observable. These variations are independent of the precision and show up on both 32 and 64 bit machines. There is an option in LS-DYNA to use an ordered summation of the global right hand side force vector to eliminate numerical differences. To achieve this the element force vectors are stored. After leaving the element loop, the global force vector is assembled in the same order that occurs on one processor. The ordered summation option is slower and uses more memory than the default, but it leads to nearly identical, if not identical, results run to run.

Parallelization in LS-DYNA was initially done with vector machines as the target where the vector speed up is typically 10 times faster than scalar. On vector machines, therefore, vectorization comes first. If the problem is large enough then parallelization is automatic. If vector lengths are 128, for example, and if 256 beam elements are used only a factor of 2 in speed can be anticipated while processing beam elements. Large contact surfaces will effectively run in parallel, small surfaces having under 100 segments will not. The speed up in the contact subroutines has only registered 7 on 8 processors due to the presence of GUARD statements around the force assembly. Because real models often use many special options that will not even vectorize efficiently it is unlikely that more than 95% of a given problem will run in parallel on a shared memory parallel machine.

34

Airbags

Additional information on the airbag modeling and comparisons with experimental data can be found in a report [Hallquist, Stillman, Hughes, and Tarver 1990] based on research sponsored by the Motor Vehicles Manufacturers Association (MVMA).

34.1 Control Volume Modeling

A direct approach for modeling the contents of the airbag would be to discretize the interior of the airbag using solid elements. The total volume and pressure-volume relationship of the airbag would then be the sum of all the elemental contributions. Although this direct approach could be applied in a straight forward manner to an inflated airbag, it would become very difficult to implement during the inflation phase of the airbag deployment. Additionally, as the model is refined, the solid elements would quickly overwhelm all other computational costs and make the numerical simulations prohibitively expensive.

An alternative approach for calculating the airbag volume, that is both applicable during the inflation phase and less computationally demanding, treats the airbag as a control volume. The control volume is defined as the volume enclosed by a surface. In the present case, the 'control surface' that defines the control volume is the surface modeled by shell or membrane elements comprising the airbag fabric material.

Because the evolution of the control surface is known, i.e., the position, orientation, and current surface area of the airbag fabric elements are computed and stored at each time step, we can take advantage of these properties of the control surface elements to calculate the control volume, i.e., the airbag volume. The area of the control surface can be related to the control volume through Green's Theorem

$$\iiint \phi \frac{\partial \psi}{\partial x} dx dy dz = - \iiint \psi \frac{\partial \phi}{\partial x} dx dy dz + \oint \phi \psi n_x d\Gamma, \quad (34.1)$$

where the first two integrals are integrals over a closed volume, i.e., $dv = dxdydz$, the last integral is an integral over the surface enclosing the volume, and n_x is the direction cosine between the surface normal and the x direction (corresponding to the x -partial derivative); similar forms can be written for the other two directions. The two arbitrary functions ϕ and ψ need only be integrated over the volume and surface.

The integral form of the volume can be written as

$$V = \iiint dxdydz. \quad (34.2)$$

Comparing the first of the volume integrals in Equation (34.1) to Equation (34.2), we can easily obtain the volume integral from Equation (34.1) by choosing for the two arbitrary functions

$$\phi = 1, \quad (34.3)$$

$$\psi = x_x, \quad (34.4)$$

leading to

$$V = \int \int \int dxdydz = \oint xn_x d\Gamma. \quad (34.5)$$

The surface integral in Equation (34.5) can be approximated by a summation over all the elements comprising the airbag, i.e.,

$$\oint xn_x d\Gamma \approx \sum_{i=1}^N \bar{x}_i n_{ix} A_i, \quad (34.6)$$

where for each element i : \bar{x}_i is the average x coordinate, n_{ix} is the direction cosine between the elements normal and the x direction, and A_i is the surface area of the element.

Although Equation (34.5) will provide the exact analytical volume for an arbitrary direction, i.e., any n , the numerical implementation of Equation (34.5), and its approximation Equation (34.6), has been found to produce slightly different volumes, differing by a few percent, depending on the choice of directions: if the integration direction is nearly parallel to a surface element, i.e., the direction cosine is nearly zero, numerical precision errors affect the volume calculation. The implementation uses as an integration direction, a direction that is parallel to the maximum principle moment of inertia of the surface. Numerical experiments have shown this choice of integration direction produces more accurate volumes than the coordinate or other principle inertia directions.

Because airbag models may contain holes, e.g., holes for inflation and deflation, and Green's Theorem only applies to closed surfaces, a special treatment is needed for calculating the volume of airbags with holes. This special treatment consists of the following:

- The n -sided polygon defining the hole is identified automatically, using edge locating algorithms in LS-DYNA.

- The n -sided polygon is projected onto a plane, i.e., it is assumed to be flat; this is a good approximation for typical airbag hole geometries. Planar symmetry should work with the control volume capability for one symmetry plane.
- The area of the flat n -sided polygon is calculated using Green's Theorem in two dimensions.
- The resulting holes are processed as another surface element in the airbag control volume calculation.

34.2 Equation of State Model

As explained above, at each time step in the calculation the current volume of the airbag is determined from the control volume calculation. The pressure in the airbag corresponding to the control volume is determined from an equation of state (EOS) that relates the pressure to the current gas density (volume) and the specific internal energy of the gas.

The equation of state used for the airbag simulations is the usual 'Gamma Law Gas Equation of State',

$$p = (k - 1)\rho e, \quad (34.7)$$

where p is the pressure, k is a constant defined below, ρ is the density, and e is the specific internal energy of the gas. The derivation of this equation of state is obtained from thermodynamic considerations of the adiabatic expansion of an ideal gas. The incremental change in internal energy, dU , in n moles of an ideal gas due to an incremental increase in temperature, dT , at constant volume is given by

$$dU = nc_v dT, \quad (34.8)$$

where c_v is the specific heat at constant volume. Using the ideal gas law we can relate the change in temperature to a change in the pressure and total volume, v , as

$$d(pv) = nRdT, \quad (34.9)$$

where R is the universal gas constant. Solving the above for dT and substituting the result into Equation (34.8) gives

$$dU = \frac{c_v d(pv)}{R} = \frac{d(pv)}{(k - 1)}, \quad (34.10)$$

where we have used the relationship

$$R = c_p - c_v, \quad (34.11)$$

and the notation

$$k = \frac{c_p}{c_v}. \quad (34.12)$$

Equation (34.10) may be rewritten as

$$dU = \frac{\rho_0 v_0}{k-1} d\left(\frac{p}{\rho}\right), \quad (34.13)$$

and integrated to yield

$$e = \frac{U}{\rho_0 v_0} = \frac{p}{\rho(k-1)}. \quad (34.14)$$

Solving for the pressure

$$p = (k-1)\rho e. \quad (34.15)$$

The equation of state and the control volume calculation can only be used to determine the pressure when the specific internal energy is also known. The evolution equation for the internal energy is obtained by assuming the change in internal energy is given by

$$dU = -pdv, \quad (34.16)$$

where the minus sign is introduced to emphasize that the volume increment is negative when the gas is being compressed. This expression can be written in terms of the specific internal energy as

$$de = \frac{dU}{\rho_0 v_0} = -\frac{pdv}{\rho_0 v}. \quad (34.17)$$

Next, we divide the above by the equation of state, Equation (4.11.144), to obtain

$$\frac{de}{e} = -\frac{\rho(k-1)dv}{\rho_0 v_0} = -\frac{(k-1)dv}{v}, \quad (34.18)$$

which may be integrated to yield

$$\ln e = (1-k)\ln V, \quad (34.19)$$

or evaluating at two states and exponentiating both sides yields

$$e_2 = e_1 \left(\frac{v_2}{v_1}\right)^{(1-k)}. \quad (34.20)$$

The specific internal energy evolution equation, Equation (34.20), the equation of state, Equation (4.11.144), and the control volume calculation completely define the pressure-volume relation for an inflated airbag.

34.3 Airbag Inflation Model

Airbag inflation models have been used for many years in occupant simulation codes such as CAL3D [Fleck, 1981].

The inflation model we chose to implement in LS-DYNA is due to Wang and Nefske[1988] and more recent improvements to the model in LS-DYNA were suggested by Wang [1992]. In their development they consider the mass flow due to the vents and leakage through the bag. We assume that the mass flow rate and the temperature of the gas going into the bag from an inflator are provided as tabulated functions of time.

A pressure relation is defined:

$$Q = \frac{p_e}{p_2}, \quad (34.21)$$

where p_e is the external pressure and p_2 is the internal pressure in the bag. A critical pressure relationship is defined as:

$$Q_{\text{crit}} = \left(\frac{2}{k+1} \right)^{k/(k-1)}, \quad (34.22)$$

where k is the ratio of specific heats:

$$k = \frac{c_p}{c_v}. \quad (34.23)$$

If $Q \leq Q_{\text{crit}}$ then $Q = Q_{\text{crit}}$.

Wang and Nefske define the mass flow through the vents and leakage by

$$\dot{m}_{23} = C_{23} A_{23} \frac{p_2}{R \sqrt{T_2}} Q^{1/k} \sqrt{2g_c \left(\frac{kR}{k-1} \right) \left(1 - Q^{k-1/k} \right)}, \quad (34.24)$$

and

$$\dot{m}'_{23} = C'_{23} A'_{23} \frac{p_2}{R \sqrt{T_2}} Q^{1/k} \sqrt{2g_c \left(\frac{kR}{k-1} \right) \left(1 - Q^{k-1/k} \right)}, \quad (34.25)$$

where C_{23} , A_{23} , C'_{23} , A'_{23} , R and g_c are the vent orifice coefficient, vent orifice area, the orifice coefficient for leakage, the area for leakage, the gas constant, and the gravitational conversion constant, respectively. The internal temperature of the airbag gas is denoted by T_2 . We note that both A_{23} and A'_{23} can be defined as a function of pressure [Wang, 1992] or if they are input as zero they are computed within LS-DYNA. This latter option requires detailed modeling of the airbag with all holes included.

A uniform temperature and pressure is assumed; therefore, in terms of the total airbag volume V_2 and air mass, m_2 , the perfect gas law is applied:

$$p_2 V = m_2 R T_2. \quad (34.26)$$

Solving for T_2 :

$$T_2 = \frac{p_2 V}{m_2 R}, \quad (34.27)$$

and substituting Equation (34.27) into equations (34.25), we arrive at the mass transient equation:

$$\dot{m}_{\text{out}} = \dot{m}_{23} + \dot{m}'_{23} = \mu \sqrt{2 p_2 \rho} \sqrt{\frac{k \left(Q^{\frac{2}{k}} - Q^{k+1/k} \right)}{k-1}} \quad (34.28)$$

where

$$\begin{aligned} \rho &= \text{density of airbag gas,} \\ \mu &= \text{bag characterization parameter,} \\ \dot{m}_{\text{out}} &= \text{total mass flow rate out of bag.} \end{aligned}$$

In terms of the constants used by Wang and Nefske:

$$\mu = \sqrt{g_c} (C_{23} A_{23} + C'_{23} A'_{23}). \quad (34.29)$$

We solved these equations iteratively, via function evaluation. Convergence usually occurs in 2 to 3 iterations.

The mass flow rate and gas temperature are defined in load curves as a function of time. Using the mass flow rate we can easily compute the increase in internal energy:

$$\dot{E}_{\text{in}} = c_p \dot{m}_{\text{in}} T_{\text{in}}, \quad (34.30)$$

where T_{in} is the temperature of the gas flowing into the airbag. Initializing the variables pressure, p , density, ρ , and energy, E , to their values at time n , we can begin the iterations loop to compute the new pressure, p^{n+1} , at time $n+1$.

$$\begin{aligned} p^{n+1/2} &= \frac{p^n + p^{n+1}}{2} \\ \rho^{n+1/2} &= \frac{\rho^n + \rho^{n+1}}{2} \\ E^{n+1/2} &= \frac{E^n + E^{n+1}}{2} \\ Q^{n+1/2} &= \max \left(\frac{p_e}{p_2^{n+1/2}}, Q_{\text{crit}} \right). \end{aligned} \quad (34.31)$$

The mass flow rate out of the bag, \dot{m}_{out} can now be computed:

$$\dot{m}_{\text{out}}^{n+1/2} = \mu \sqrt{2 p_2^{n+1/2} \rho^{n+1/2}} \sqrt{\frac{k \left(Q^{n+1/2 \cdot 2/k} - Q^{n+1/2 \cdot k+1/k} \right)}{k-1}}, \quad (34.32)$$

where

$$p_2^{n+1/2} = p^{n+1/2} + p_e, \quad (34.33)$$

and the total mass updated:

$$\begin{aligned} m^{n+1} &= m^n + \Delta t \left(\dot{m}_{\text{in}}^{n+1/2} - \dot{m}_{\text{out}}^{n+1/2} \right) \\ m^{n+1/2} &= \frac{m^n + m^{n+1}}{2}. \end{aligned} \quad (34.34)$$

The energy exiting the airbag is given by:

$$\dot{E}_{\text{out}}^{n+1/2} = \dot{m}_{\text{out}}^{n+1/2} \frac{E^{n+1/2}}{m^{n+1/2}}, \quad (34.35)$$

we can now compute our new energy at time $n + 1$

$$E^{n+1} = E^n + \Delta t \left(\dot{E}_{\text{in}}^{n+1/2} - \dot{E}_{\text{out}}^{n+1/2} \right) - p^{n+1/2} \Delta V^{n+1/2}, \quad (34.36)$$

where $\Delta V^{n+1/2}$ is the change in volume from time n to $n + 1$. The new pressure can now be computed:

$$p^{n+1} = (k - 1) \frac{E^{n+1}}{V^{n+1}}, \quad (34.37)$$

which is the gamma-law (where $k = \gamma$) gas equation. This ends the iteration loop.

34.4 Wang's Hybrid Inflation Model

Wang's proposed hybrid inflator model [1995a, 1995b] provides the basis for the model in LS-DYNA. The first law of thermodynamics is used for an energy balance on the airbag control volume.

$$\frac{d}{dt}(mu)_{\text{cv}} = \sum \dot{m}_i h_i - \sum \dot{m}_o h_o - \dot{W}_{\text{cv}} - \dot{Q}_{\text{cv}}, \quad (34.38)$$

where

$$\begin{aligned} \frac{d}{dt}(mu)_{\text{cv}} &= \text{rate of change of airbag internal energy} \\ \sum \dot{m}_i h_i &= \text{energy into airbag by mass flow (e.g., inflator)} \\ \sum \dot{m}_o h_o &= \text{energy out of airbag by mass flow (e.g., vents)} \\ \dot{W}_{\text{cv}} &= \int P d\dot{V} = \text{work done by airbag expansion} \\ \dot{Q}_{\text{cv}} &= \text{energy out by heat transfer through airbag surface.} \end{aligned}$$

The rate of change of internal energy, the left hand side of Equation (34.38), can be differentiated:

$$\frac{d}{dt}(mu) = \frac{dm}{dt}u + m\frac{du}{dt} = \frac{dm}{dt}u + m\frac{d}{dt}(c_v T) = \dot{m}u + m\dot{c}_v T + mc_v \frac{dT}{dt}, \quad (34.39)$$

where we have used the definition

$$u = c_v T. \quad (34.40)$$

Then, the energy equation can be re-written for the rate of change in temperature for the airbag

$$\frac{dT_{cv}}{dt} = \frac{\sum \dot{m}_i h_i - \sum \dot{m}_o h_o - \dot{W}_{cv} - \dot{Q}_{cv} - (\dot{m}u)_{cv} - (m\dot{c}_v T)_{cv}}{(mc_v)_{cv}} \quad (34.41)$$

Temperature dependent heat capacities are used. The constant pressure molar heat capacity is taken as:

$$\bar{c}_p = \bar{a} + \bar{b}T, \quad (34.42)$$

and the constant volume molar heat capacity as:

$$\bar{c}_v = \bar{a} + \bar{b}T - \bar{r}, \quad (34.43)$$

where

$$\begin{aligned} \bar{r} &= \text{gas constant} = 8.314 \text{ J/gm-mole K} \\ \bar{a} &= \text{constant [J/gm-mole K]} \\ \bar{b} &= \text{constant [J/gm-mole K}^2\text{]} \end{aligned}$$

Mass based values are obtained by dividing the molar quantities by the molecular weight, M , of the gas

$$a = \frac{\bar{a}}{M}, \quad b = \frac{\bar{b}}{M}, \quad r = \frac{\bar{r}}{M}. \quad (34.44)$$

The constant pressure and volume specific heats are then given by

$$c_p = a + bT \quad (34.45)$$

$$c_v = a + bT - r. \quad (34.46)$$

The specific enthalpy and internal energy becomes:

$$h = \int_0^T c_p dT = aT + \frac{bT^2}{2} \quad (34.47)$$

$$u = \int_0^T c_v dT = aT + \frac{bT^2}{2} - rT. \quad (34.48)$$

For ideal gas mixtures the molecular weight is given as:

$$M = \frac{1}{\sum \frac{f_i}{M_i}}, \quad (34.49)$$

and the constant pressure and volume specific heats as:

$$c_p = \sum f_i c_{p(i)} \quad (34.50)$$

$$c_v = \sum f_i c_{v(i)}, \quad (34.51)$$

where

$$\begin{aligned} f_i &= \text{mass fraction of gas } i \\ M_i &= \text{molecular weight of gas } i \\ c_{p(i)} &= \text{constant pressure specific heat of gas } i \\ c_{v(i)} &= \text{constant volume specific heat of gas } i. \end{aligned}$$

The specific enthalpy and internal energy for an ideal gas mixture with temperature dependent heat capacity are

$$h = \int_0^T \sum f_i c_{p(i)} dT = \sum f_i \left(a_i T + \frac{b_i T^2}{2} \right) \quad (34.52)$$

$$u = \int_0^T \sum f_i c_{v(i)} dT = \sum f_i \left(a_i T + \frac{b_i T^2}{2} - r_i T \right). \quad (34.53)$$

The rate of change of temperature for the airbag is

$$\frac{dT_{cv}}{dt} = \frac{\sum \dot{m}_i h_i - \sum \dot{m}_o h_o - \dot{W}_{cv} - \dot{Q}_{cv} - (\dot{m}u)_{cv} - (m\dot{c}_v T)_{cv}}{(m c_v)_{cv}}. \quad (34.54)$$

The energy in by mass flow becomes:

$$\sum \dot{m}_i h_i = \sum \dot{m}_i \left(a_i T_i + \frac{b_i T_i^2}{2} \right), \quad (34.55)$$

\dot{m}_i is specified by an inflator mass inflow vs. time table

T_i is specified by an inflator temperature vs. time table

a, b are input constants for gas i

And the energy out by mass flow:

$$\sum \dot{m}_o h_o = \sum \dot{m}_o \left[\sum_{\text{gases}} f_i \left(a_i T_{cv} + \frac{b_i T_{cv}^2}{2} \right) \right]. \quad (34.56)$$

The gas leaves the airbag at the control volume temperature T_{cv} . The mass flow rate out through vents and fabric leakage is calculated by the one dimensional isentropic

flow equations per Wang and Nefske. The work done by the airbag expansion is given by:

$$\dot{W}_{cv} = \int P d\dot{V}, \quad (34.57)$$

P is calculated by the equation of state for a perfect gas, $p = \rho RT$ and \dot{V} is calculated by LS-DYNA

For the energy balance, we must compute the energy terms $(\dot{m}u)_{cv}$ and $(\dot{m}c_v)_{cv}$. Conservation of mass leads to:

$$\begin{aligned} \dot{m}_{cv} &= \dot{m}_i - \dot{m}_o \\ m_{cv} &= \int \dot{m}_{cv} dt. \end{aligned} \quad (34.58)$$

The internal energy is given by

$$u_{cv} = \sum f_i \left(a_i T_{cv} + \frac{b_i T_{cv}^2}{2} - r_i T_{cv} \right), \quad (34.59)$$

and the heat capacity at contact volume is:

$$(c_v)_{cv} = \sum f_i (a_i + b_i T_{cv} - r_i). \quad (34.60)$$

34.5 Constant Volume Tank Test

Constant volume tank tests are used to characterize inflators. The inflator is ignited within the tank and, as the propellant burns, gas is generated. The inflator temperature is assumed to be constant. From experimental measurements of the time history of the tank pressure it is straightforward to derive the mass flow rate, \dot{m} . From energy conservation, where T_i and T_t are defined to be the temperature of the inflator and tank, respectively, we obtain:

$$c_p \dot{m} T_i = c_v \dot{m} T_t + c_v \dot{m} \dot{T}_t. \quad (34.61)$$

For a perfect gas under constant volume, $\dot{V} = 0$, hence,

$$\dot{p}V = \dot{m}RT_t + mR\dot{T}_t, \quad (34.62)$$

and, finally, we obtain the desired mass flow rate:

$$\dot{m} = \frac{c_v \dot{p}V}{c_p R T_i}. \quad (34.63)$$

35

Dynamic Relaxation and System Damping

Dynamic relaxation allows LS-DYNA to approximate solutions to linear and nonlinear static or quasi-static processes. Control parameters must be selected with extreme care or bad results can be obtained. The current methods are not compatible with displacement or velocity boundary conditions, but various body loads, thermal loads, pressures, and nodal loads are allowed. The solutions to most nonlinear problems are path dependent, thus results obtained in the presence of dynamic oscillations may not be the same as for a nonlinear implicit code, and they may diverge from reality.

In LS-DYNA we have two methods of damping the solution. The first named “dynamic relaxation” is used in the beginning of the solution phase to obtain the initial stress and displacement field prior to beginning the analysis. The second is system damping which can be applied anytime during the solution phase either globally or on a material basis.

35.1 Dynamic Relaxation For Initialization

In this phase only a subset of the load curves is used to apply the static load which is flagged in the load curve section of the manual. The calculation begins and executes like a normal LS-DYNA calculation but with damping incorporated in the update of the displacement field.

Our development follows the work of Underwood [1986] and Papadrakakis [1981] with the starting point being the dynamic equilibrium equation, Equation (23.1) with the addition of a damping term, at time n :

$$\mathbf{M}\mathbf{a}^n + \mathbf{C}\mathbf{v}^n + \mathbf{Q}^n(\mathbf{d}) = 0, \quad (35.1)$$

$$\mathbf{Q}^n(\mathbf{d}) = \mathbf{F}^n - \mathbf{P}^n - \mathbf{H}^n, \quad (35.2)$$

where we recall that \mathbf{M} is the mass matrix, \mathbf{C} is the damping matrix, n indicates the n th time step, \mathbf{a}^n is the acceleration, \mathbf{v}^n the velocity, and \mathbf{d} is the displacement vector. With Δt as the fixed time increment we get for the central difference scheme:

$$\mathbf{v}^{n+1/2} = \frac{(\mathbf{d}^{n+1} - \mathbf{d}^n)}{\Delta t}; \quad \mathbf{a}^n = \frac{(\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2})}{\Delta t}. \quad (35.3)$$

For \mathbf{v}^n we can assume an averaged value

$$\mathbf{v}^n = \frac{1}{2}(\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2}), \quad (35.4)$$

and obtain

$$\mathbf{v}^{n+1/2} = \left(\frac{1}{\Delta t} \mathbf{M} + \frac{1}{2} \mathbf{C} \right)^{-1} \left[\left(\frac{1}{\Delta t} \mathbf{M} - \frac{1}{2} \mathbf{C} \right) \mathbf{v}^{n-1/2} - \mathbf{Q}^n \right], \quad (35.5)$$

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t \mathbf{v}^{n+1/2}. \quad (35.6)$$

In order to preserve the explicit form of the central difference integrator, \mathbf{M} and \mathbf{C} must be diagonal. For the dynamic relaxation scheme \mathbf{C} has the form

$$\mathbf{C} = c \cdot \mathbf{M}. \quad (35.7)$$

If Equation (35.7) is substituted into (35.5) the following form is achieved

$$\mathbf{v}^{n+1/2} = \frac{2 - c\Delta t}{2 + c\Delta t} \mathbf{v}^{n-1/2} + \frac{2\Delta t}{2 + c\Delta t} \cdot \mathbf{M}^{-1} \cdot \mathbf{Q}^n. \quad (35.8)$$

Since \mathbf{M} is diagonal, each solution vector component may be computed individually from

$$\mathbf{v}_i^{n+1/2} = \frac{2 - c\Delta t}{2 + c\Delta t} \mathbf{v}_i^{n-1/2} + \frac{2\Delta t}{2 + c\Delta t} \frac{\mathbf{Q}_i^n}{m_i}. \quad (35.9)$$

As a starting procedure it is suggested by Underwood

$$\begin{aligned} \mathbf{v}^0 &= 0 \\ \mathbf{d}^0 &= 0. \end{aligned} \quad (35.10)$$

Since the average value is used for \mathbf{v}^n , which must be zero at the beginning for a quasi-static solution

$$\mathbf{v}^{-1/2} = -\mathbf{v}^{1/2}, \quad (35.11)$$

thus the velocity at time $+1/2$ is

$$\mathbf{v}^{1/2} = -\frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{Q}^0. \quad (35.12)$$

A damping coefficient must now be selected to obtain convergence to the static solution in minimal time. The best estimate for damping values is based on the frequencies of the structure. One choice is to focus on an optimal damping parameter as suggested by

Papadrakakis [1981]. Then dynamic relaxation is nothing else but a critically damped dynamic system

$$C = C_{cr} = 2\omega_{\min}m, \quad (35.13)$$

with m as modal mass. The problem is finding the dominant eigenvalue in the structure related to the “pseudo-dynamic” behavior of the structure. As the exact estimate would be rather costly and not fit into the explicit algorithm, an estimate must be used. Papadrakakis suggests

$$\lambda_D = \frac{\|\mathbf{d}^{n+1} - \mathbf{d}^n\|}{\|\mathbf{d}^n - \mathbf{d}^{n-1}\|}. \quad (35.14)$$

When this quantity has converged to an almost constant value, the minimum eigenvalue of the structure can be estimated:

$$\omega_{\min}^2 = -\frac{(\lambda_D^2 - \lambda_D \cdot \beta + \alpha)}{\lambda_D \cdot \gamma}, \quad (35.15)$$

where

$$\begin{aligned} \alpha &= \frac{2 - c\Delta t}{2 + c\Delta t} \\ \beta &= \alpha + 1 \\ \gamma &= \frac{2\Delta t^2}{2 + c\Delta t}. \end{aligned} \quad (35.16)$$

The maximum eigenvalue determines the time step and is already known from the model

$$\omega_{\max}^2 = \frac{4.0}{(\Delta t)^2}. \quad (35.17)$$

Now the automatic adjustment of the damping parameter closely follows the paper of Papadrakakis, checking the current convergence rate compared to the optimal convergence rate. If the ratio is reasonably close, then an update of the iteration parameters is performed.

$$c = \frac{4.0}{\Delta t} \frac{\sqrt{\omega_{\min}^2 \cdot \omega_{\max}^2}}{(\omega_{\min}^2 + \omega_{\max}^2)}. \quad (35.18)$$

As is clearly visible from Equation (35.18) the value of highest frequency has always a rather high influence on the damping ratio. This results in a non-optimal damping ratio, if the solution is dominated by the response in a very low frequency compared to the highest frequency of the structure. This is typically the case in shell structures, when bending dominates the solution. It was our observation that the automatic choice following Papadrakakis results in very slow convergence for such structures, and this is also mentioned by Underwood for similar problems. The damping ratio should then be fully adjusted to the lowest frequency by hand by simply choosing a rather high damping ratio. An automatic adjustment for such cases is under preparation. For structures with

dominant frequencies rather close to the highest frequency, convergence is really improved with the automatically adjusted parameter.

If the automated approach is not used then we apply the damping as

$$\mathbf{v}^{n+1/2} = \eta \mathbf{v}^{n-1/2} + \mathbf{a}^n \Delta t, \quad (35.19)$$

where η is an input damping factor (defaulted to .995). The factor, η , is equivalent to the corresponding factor in Equations (31.7- 31.8).

The relaxation process continues until a convergence criterion based on the global kinetic energy is met, i.e., convergence is assumed if

$$E_{ke} < CVTOL \cdot E_{ke}^{\max}, \quad (35.20)$$

where CVTOL is the convergence tolerance (defaulted to .001). The kinetic energy excludes any rigid body component. Initial velocities assigned in the input are stored during the relaxation. Once convergence is attained the velocity field is initialized to the input values. A termination time for the dynamic relaxation phase may be included in the input and is recommended since if convergence fails, LS-DYNA will continue to execute indefinitely.

35.2 Mass Weighted Damping

With mass weighted damping, the Equation (23.2) is modified as:

$$\mathbf{a}^n = \mathbf{M}^{-1}(\mathbf{P}^n - \mathbf{F}^n + \mathbf{H}^n - \mathbf{F}_{\text{damp}}^n), \quad (35.21)$$

where

$$F_{\text{damp}}^n = D_s m v. \quad (35.22)$$

As seen from [Figure 35.1](#), the critical damping constant will serve to damp a dynamic solution to a static solution in minimal time. The critical damping constant is

$$D_s = 2\omega_{\min} \quad (35.23)$$

35.3 Dynamic Relaxation—How Fast Does it Converge?

The number of cycles required to reduce the amplitude of the dynamic response by a factor of 10 can be approximated by [see Stone, Krieg, and Beisinger 1985]

$$n_{\text{cycle}} = 1.15 \frac{\omega_{\text{max}}}{\omega_{\text{min}}}. \quad (35.24)$$

Structural problems which involve shell and beam elements can have a very large ratio and consequently very slow convergence.

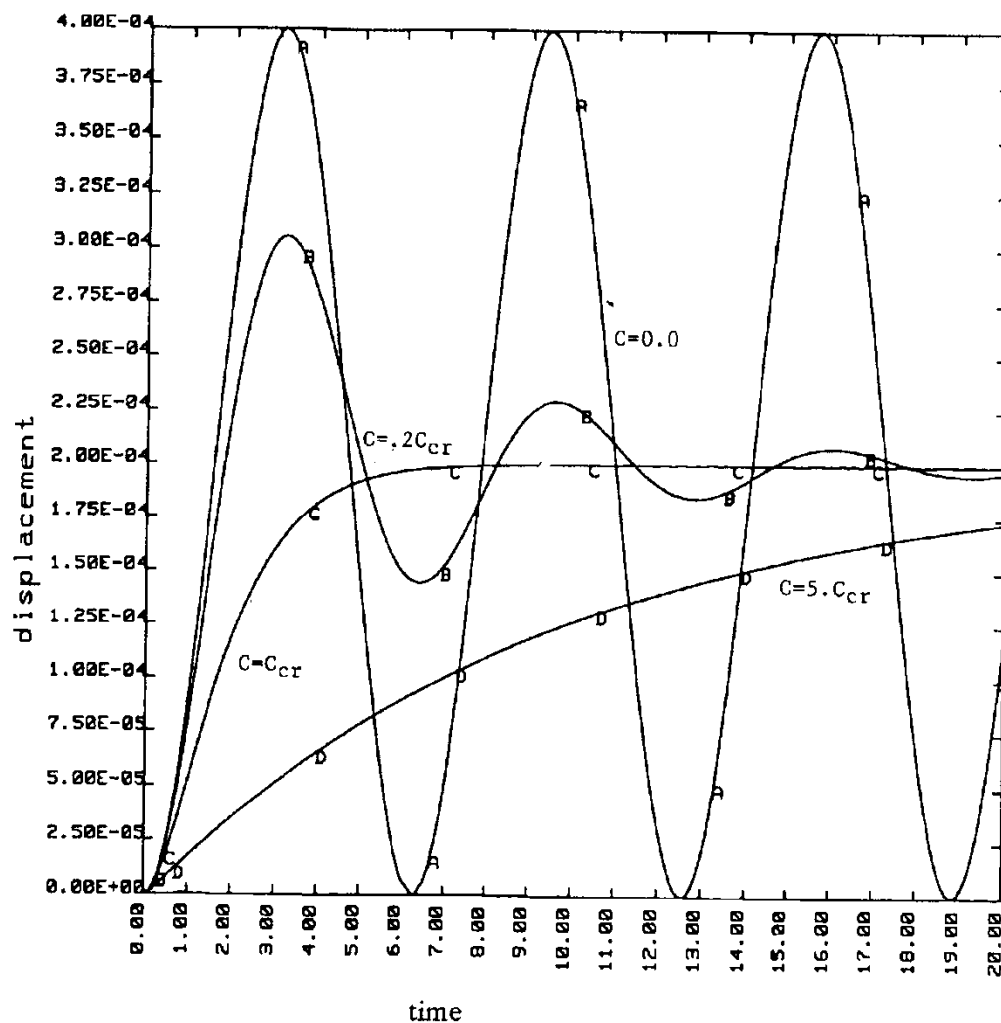


Figure 35.1. Displacement versus time curves with a variety of damping coefficients applied to a one degree-of-freedom oscillator.

36

Heat Transfer

LS-DYNA can be used to solve for the steady state or transient temperature field on three-dimensional geometries. Material properties may be temperature dependent and either isotropic or orthotropic. A variety of time and temperature dependent boundary conditions can be specified including temperature, flux, convection, and radiation. The implementation of heat conduction into LS-DYNA is based on the work of Shapiro [1985].

36.1 Conduction of Heat in an Orthotropic Solid

The differential equations of conduction of heat in a three-dimensional continuum is given by

$$\rho c_p \frac{\partial \theta}{\partial t} = (k_{ij} \theta_{,j})_{,i} + Q \quad (36.1)$$

subject to the boundary conditions, $\theta = \theta_s$ on Γ_1 , $k_{ij} \theta_{,j} n_i + \beta \theta = \gamma$ on Γ_2 , and initial conditions at t_0 :

$$\theta_\Gamma = \theta_0(x_i) \quad \text{at} \quad t = t_0. \quad (36.2)$$

where

- $\theta = \theta(x_i, t)$ temperature
- $x_i = x_i(t)$ coordinates as a function of time
- $\rho = \rho(x_i)$ density
- $c_p = c_p(x_i, \theta)$ specific heat capacity
- $k_{ij} = k_{ij}(x_i, \theta)$ thermal conductivity
- $Q = Q(x_i, \theta)$ internal heat generation rate per unit volume Ω
- $\theta_\Gamma =$ prescribed temperature on Γ_1
- $n_i =$ normal vector to Γ_2

Equations (36.1)-(36.2) represent the strong form of a boundary value problem to be solved for the temperature field within the solid.

DYNA3D employs essentially the same theory as TOPAZ [Shapiro 1985] in solving Equation (36.1) by the finite element method. Those interested in a more detailed description of the theory are referred to the TOPAZ User's Manual. Brick elements are integrated with a $2 \times 2 \times 2$ Gauss quadrature rule, with temperature dependence of the properties accounted for at the Gauss points. Time integration is performed using a generalized trapezoidal method shown by Hughes to be unconditionally stable for nonlinear problems. Newton's method is used to satisfy equilibrium in nonlinear problems.

The finite element method provides the following equations for the numerical solution of Equations (36.1)-(36.2)

$$\left[\frac{C_{n+\alpha}}{\Delta t} + \alpha H_{n+\alpha} \right] \{ \theta_{n+1} - \theta_n \} = \{ F_{n+\alpha} - H_{n+\alpha} \theta_n \} \quad (36.3)$$

where

$$[C] = \sum_e [C_{ij}^e] = \sum_e \int_{\Omega^e} N_i \rho c N_j d\Omega \quad (36.4)$$

$$[H] = \sum_e [H_{ij}^e] = \sum_e \left[\int_{\Omega^e} \nabla^T N_i K \nabla N_j d\Omega + \int_{\Gamma^e} N_i \beta N_j d\Gamma \right] \quad (36.5)$$

$$[F] = \sum_e [F_i^e] = \sum_e \left[\int_{\Omega^e} N_i q_g d\Omega + \int_{\Gamma^e} N_i \gamma d\Gamma \right] \quad (36.6)$$

The parameter α is taken to be in the interval $[0,1]$. Some well-known members of this α -family are

- α Method
- 0 forward difference; forward Euler
- 1/2 midpoint rule; Crank-Nicolson
- 2/3 Galerkin
- 1 backward difference, fully implicit

36.2 Thermal Boundary Conditions

Boundary conditions are represented by

$$k_x \frac{\partial \theta}{\partial x} n_x + k_y \frac{\partial \theta}{\partial y} n_y + k_z \frac{\partial \theta}{\partial z} n_z = \gamma - \beta \theta = \dot{q}'' \quad (36.7)$$

By convention, heat flow is positive in the direction of the surface outward normal vector. Surface definition is in accordance with the right hand rule. The outward normal vector points to the right as one progresses from node N1 to N2 to N3 and finally to N4. See [Figure 36.1](#).

Boundary conditions can be functions of temperature or time. More than one boundary condition can be specified over the same surface such as in a case of combined convection and radiation. For situations where it is desired to specify adiabatic (i.e., $q'' = 0$) conditions, such as at an insulated surface or on a line of symmetry, no boundary condition need be specified. This is the default boundary condition in LS-DYNA.

Temperature boundary condition can be specified on any node whether on the physical boundary or not.

Flux, convection, and radiation boundary conditions are specified on element surface segments defined by 3 (triangular surface) or 4 nodes (quadrilateral surface). These boundary conditions can be specified on any finite element surface whether on the physical boundary or not.

- Flux: Set $q'' = q_f$, where q_f is defined at the node points comprising the flux boundary condition surface.
- Convection: A convection boundary condition is calculated using $q'' = h(T - T_\infty)$, where h is heat transfer coefficient, $(T - T_\infty)$ is temperature potential. LS-DYNA evaluates h at the film temperature

$$T = \frac{1}{2}(T_{\text{surf}} + T_\infty) \quad (36.8)$$

- Radiation: A radiation boundary condition is calculated using $q'' = h_r(T^4 - T_\infty^4)$, where $h_r = \sigma \epsilon F$ is a radiant-heat-transfer coefficient.

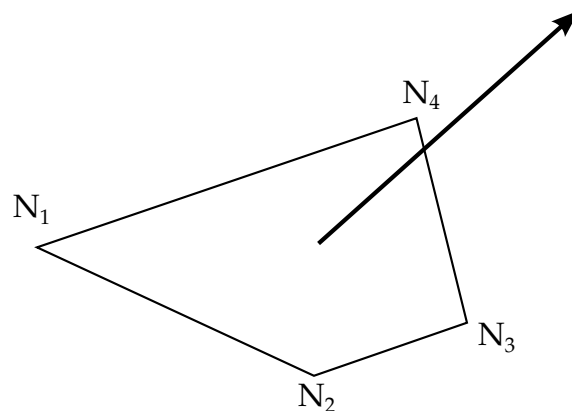


Figure 36.1. Definition of the outward normal vector

36.3 Thermal Energy Balances

Various energy terms are printed and written into the plot file for post processing using the code LS-PREPOST. The energy terms are:

- change in material internal energy for time step,
- change in material internal energy from initial time,
- heat transfer rates on boundary condition surfaces,
- heat transfer rates on enclosure radiation surfaces,
- x , y , and z fluxes at all nodes.

36.4 Heat Generation

Volumetric heat generation rates may be specified by element, by material, or both (in which case the effect is additive). Volumetric heat generation rates can be a function of time or temperature.

36.5 Initial Conditions

Initial temperature conditions can be specified on the nodal data input cards or on the nodal temperature initial condition cards. If no temperatures are specified, the default is 0. For nonlinear steady state problems the temperature initial condition serves as a first guess for the equilibrium iterations.

36.6 Material Properties

Heat capacity and thermal conductivity may be functions of temperature. Since the density and heat capacity appear only as a product in the governing equations, the temperature dependence of the density may be included in the temperature dependence of the heat capacity. Material properties are evaluated at the element Gauss point temperature or average element temperature.

The thermal conductivity may be either isotropic or orthotropic. For an orthotropic material, the three material axes (x'_1 , x'_2 , x'_3) are orthogonal and the thermal conductivity tensor \mathbf{K} is diagonal.

The thermal conductivity tensor \mathbf{K} in the global coordinate system is related by

$$K_{ij} = K'_{ij} \beta_{mi} \beta_{nj}, \quad (36.9)$$

where

$$\beta_{ij} = \cos(x'_i, x_j). \quad (36.10)$$

36.7 Nonlinear Analysis

In a nonlinear problem, C , H , and F are functions of temperature. Newton's method is used to transform equation 32.4 into an alternate form which contains temperature derivatives of C , H , and F (i.e., the tangent matrix). Iterations are required to solve this alternate form.

In a steady state nonlinear problem, an initial guess should be made of the final temperature distribution and included in the input file as an initial condition. If your guess is good, a considerable savings in computation time is achieved.

36.8 Units

Any consistent set of units with the governing equation may be used. Examples are:

Quantity	Units		
temperature	K	C	F
space	m	cm	ft
time	s	s	hr
density	kg/m ³	g/cm ³	Lbm/ft ³
heat capacity	J/kg K	cal/g C	Btu/LbmF
thermal conductivity	W/m K	cal/s cm C	Btu/hr ft F
thermal generation	W/M ³	cal/s cm ³	Btu/hr ft ³
heat flux	W/m ²	cal/s cm ²	Btu/hr ft ²

37

Adaptivity for Shell Elements

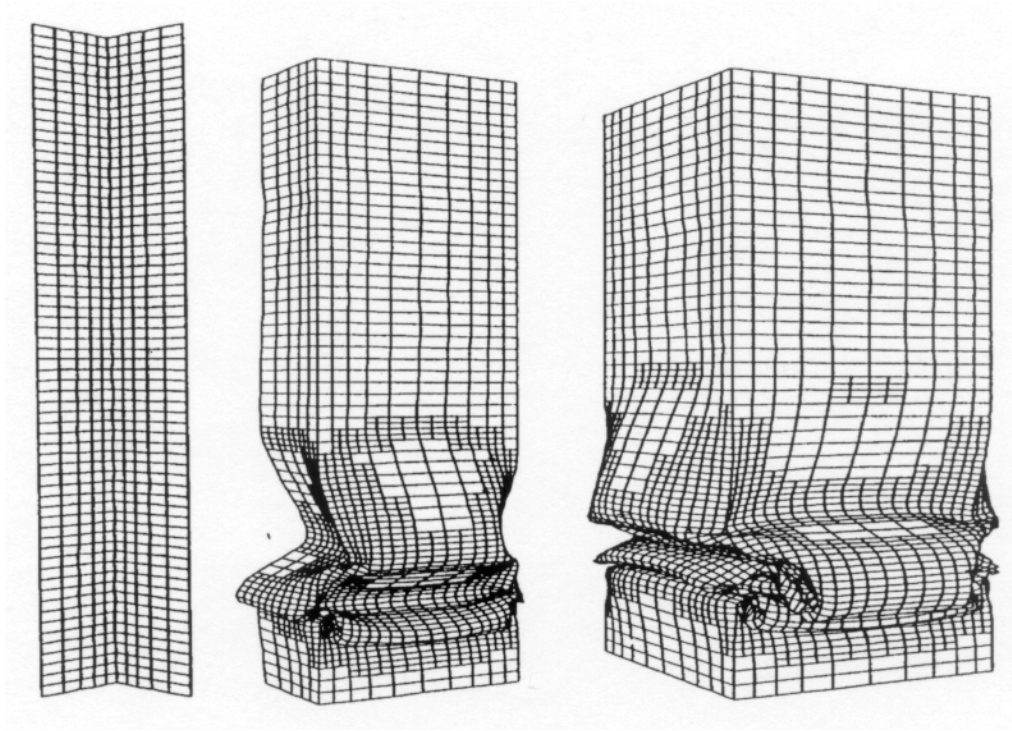


Figure 37.1. One level adaptive calculation on a square cross section beam.

LS-DYNA includes an h -adaptive method for shell elements. In an h -adaptive method, the elements are subdivided into smaller elements wherever an error indicator shows that subdivision of the elements will provide improved accuracy. [Figure 37.1](#) shows an example of an adaptive calculation on a thin wall square cross section beam. [Figures 37.2](#) through [37.4](#) illustrate a simple metal stamping simulation [also see Galbraith, Finn, et. al., 1991]. In the following, we will describe the methodologies used in the h -adaptive method in LS-DYNA. The objective of the adaptive process in LS-DYNA is to obtain the greatest accuracy for a given set of computational resources. You set the initial mesh and the maximum level of adaptivity, and LS-DYNA subdivides those elements with the largest error indicator (see `*CONTROL_ADAPTIVE`). Although this

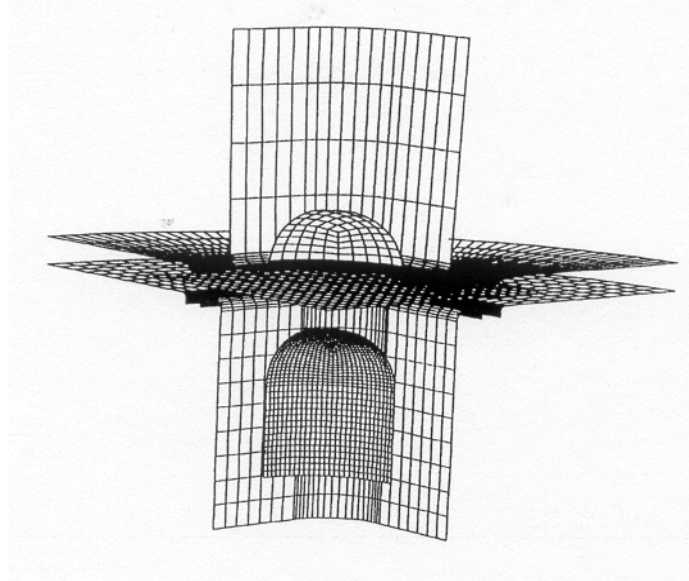


Figure 37.2. Aluminum blank with 400 shells in blank and four rigid tools.

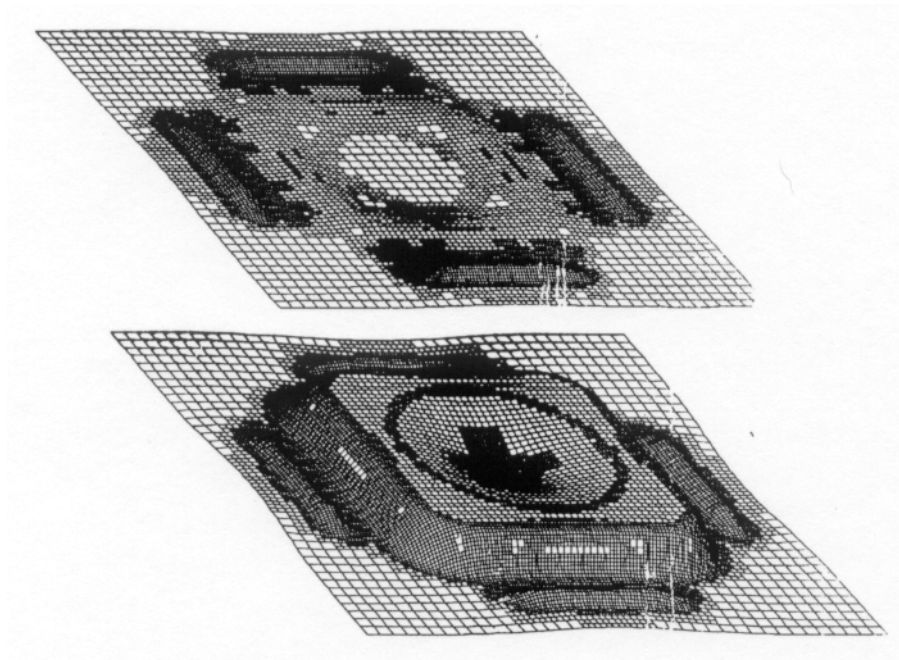


Figure 37.3. Adaptive calculations using two adaptive levels.

process does not provide control on the error of the solution, it makes it possible to obtain a solution of comparable accuracy with fewer elements, and, hence, less computational resources, than with a fixed mesh.

We based our implementation of the h -adaptive method on Belytschko, Wong, and Plaskacz [1989]. In Belytschko, Wong, and Plaskacz [1989], elements were also fused or combined when it was felt that they were no longer needed. Our initial implementations did not include this, but a fusion option was added between 2003 and 2005 for SMP. It

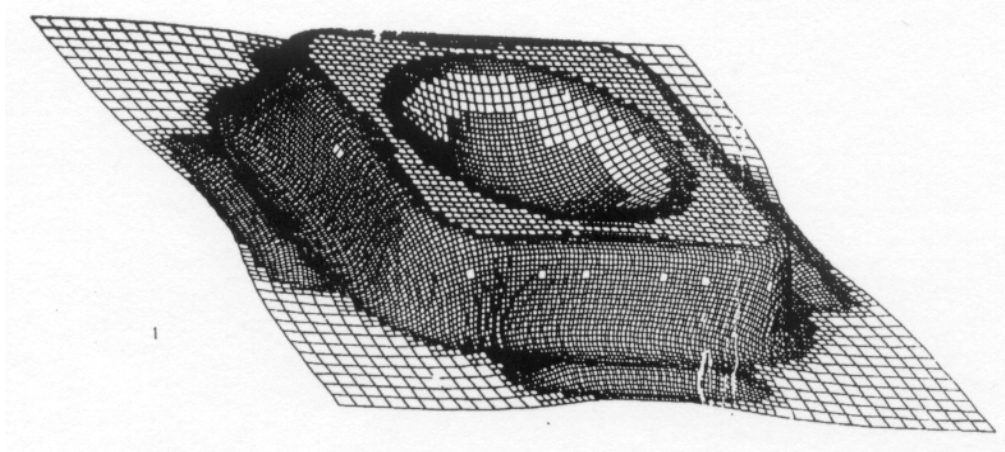


Figure 37.4. Final shape of formed part with 4315 shell elements per quarter.

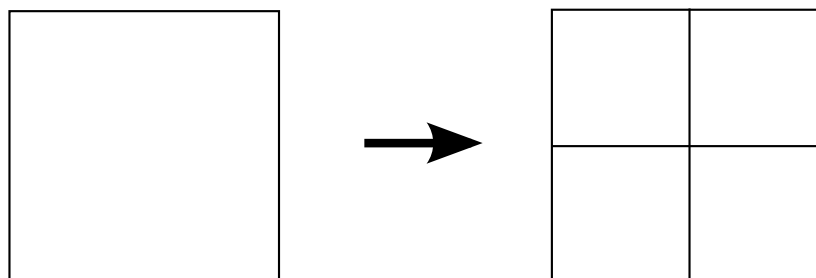


Figure 37.5. Fissioning of a Quadrilateral Element

was later extended to MPP in R11.0. Adaptivity in LS-DYNA can be restricted to specific groups of shell elements. Elements that fall in this group are said to be in the active adaptivity domain.

In the h -adaptive process, elements are subdivided into smaller elements where more accuracy is needed; this process is called fission. The elements involved in the fission process are subdivided into elements with sides $h/2$, where h is the characteristic size of the original elements. This subdivision is illustrated in [Figure 37.5](#) for a quadrilateral element. In fission, each quadrilateral is subdivided into four quadrilaterals (as indicated in [Figure 37.2](#)) by using the midpoints of the sides and the centroid of the element to generate four new quadrilaterals. The fission process for a triangular element is shown in [Figure 37.6](#) where the element is subdivided into four triangles by using the midpoints of the three sides.

The adaptive process can consist of several levels of fission. [Figure 37.5](#) shows one subdivision, which is called the second refinement level. In subsequent steps, the fissioned elements can again be fissioned in a third refinement level, and these elements can again, in turn, be fissioned in a fourth level, as shown in [Figure 37.7](#). The levels of adaptivity that occur in a mesh are restricted by three rules:

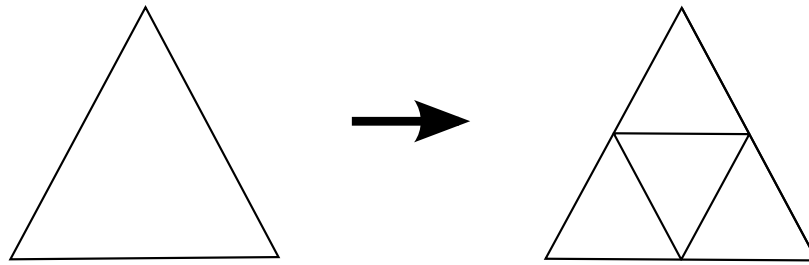


Figure 37.6. Fissioning of a Triangular Element

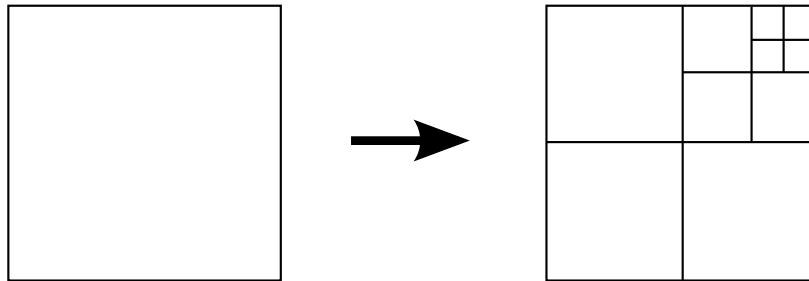


Figure 37.7. Quadrilateral Element Fissioned to the fourth level

- The number of levels is restricted by the maximum level of adaptivity that is allowed in the mesh, which is generally set at 3 or 4. At the fourth level up to 64 elements will be generated for each element in the initial mesh.
- The levels of adaptivity implemented in a mesh must be such that the levels of adaptivity implemented in adjacent elements differ by, at most, one level.
- The total number of elements can be restricted by available memory. Once the specified memory usage is reached, adaptivity ceases.

The second rule is used to enforce a 2-to-1 rule given by Oden, Devloo and Strouboulis [1986], which restricts the number of elements along the side of any element in the mesh to two. The enforcement of this rule is necessary to accommodate limitations in the data structure.

The original mesh provided by you is known as the parent mesh. The elements of this mesh are called the parent elements, and the nodes are called parent nodes. Any elements that are generated by the adaptive process are called descendant elements, and any nodes that are generated by the adaptive process are called descendant nodes. Elements generated by the second level of adaptivity are called first-generation elements, those generated by third level of adaptivity are called second-generation elements, etc.

The coordinates of the descendant nodes are generated by using linear interpolation. Thus, the coordinates of any node generated along the midpoint of the sides of an element during fission are given by

$$x_N = \frac{1}{2}(x_I + x_J), \quad (37.1)$$

where x_N is the position of the generated node and x_I and x_J are the nodes along the side on which x_N was generated for a typical element. The coordinate of the generated centroid node from a parent quadrilateral element is given by

$$x_M = \frac{1}{4}(x_I + x_J + x_K + x_L), \quad (37.2)$$

where x_I , x_J , x_K and x_L are the nodes of the original quadrilateral. The velocities of the nodes are also given by linear interpolation. The velocities of edge nodes are given by

$$v_N = \frac{1}{2}(v_I + v_J), \quad (37.3)$$

and the angular velocities are given by

$$\omega_N = \frac{1}{2}(\omega_I + \omega_J). \quad (37.4)$$

The velocities of a centroid node of a fissioned quadrilateral element are given by

$$v_M = \frac{1}{4}(v_I + v_J + v_K + v_L), \quad (37.5)$$

$$\omega_M = \frac{1}{4}(\omega_I + \omega_J + \omega_K + \omega_L). \quad (37.6)$$

The stresses in the descendant element are obtained from the parent element by setting the stresses in the descendant elements equal to the stresses in the parent element at the corresponding through-the-thickness quadrature points.

In subsequent steps, certain generated nodes are constrained by other nodes in the mesh. We will call these constrained nodes dependent nodes. The constraint only occurs when a node on the edge of an element created during an adaptive step is *not* on the boundary of the mesh and is *not* a node belonging to an element on the other side of the edge. If the element on the other side of the edge is adapted, then this constraint may be deleted. LS-DYNA adds the keyword `*CONSTRAINED_ADAPTIVITY` to the output input deck for the adaptive step for each constrained node.

Figure 37.9 illustrates which elements become dependent after adapting a single element mesh. The left element is adapted to create the elements on the right. In this figure, the red dots are dependent nodes while the blue dots are the edge nodes constraining the dependent nodes. For instance, node 2 is a dependent node constrained at the midpoint of the edge between nodes 1 and 3 and node 5 is a dependent node constrained at the midpoint of the edge between nodes 4 and 6.

Suppose that the element to the left of node 2 is adapted once. Then node 2 is no longer a dependent node and node 3 is no longer a constraining node as shown in

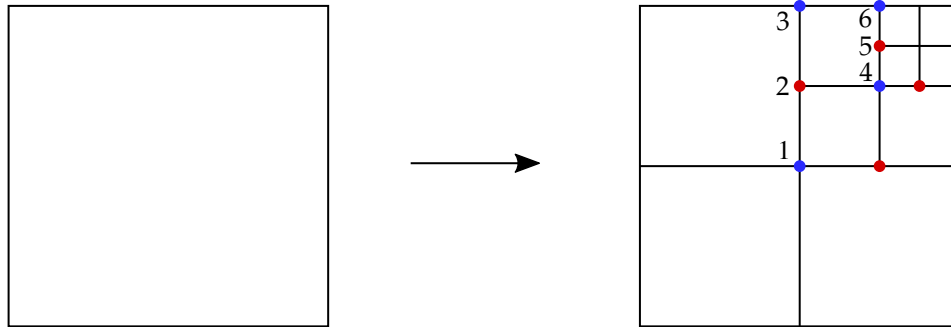


Figure 37.9. Illustration of which nodes are constrained after adaptive steps. The red nodes are dependent while the blue nodes constrain the dependent node that shares the edge.

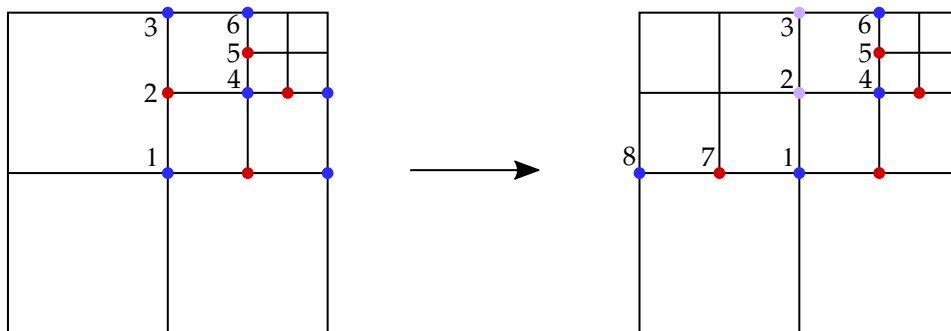


Figure 37.10. Illustration of which nodes are constrained after the element to the left of Node 2 is adapted. The purple nodes indicate which ones are no longer part of constraints.

Figure 37.10. These nodes are purple dots to illustrate that they are no longer part of a constraint. However, node 1 continues to constrain the node to its right and now constrains the node to its left (node 7) with node 8.

Refinement indicators determine the locations of mesh refinement. You can specify the indicator with ADPTYP and ADPTOL on *CONTROL_ADAPTIVE. One deformation-based approach checks for a change in angles between contiguous elements as shown in Figure 37.8. If $\zeta > \zeta_{tol}$, then refinement is indicated, where ζ_{tol} is user defined.

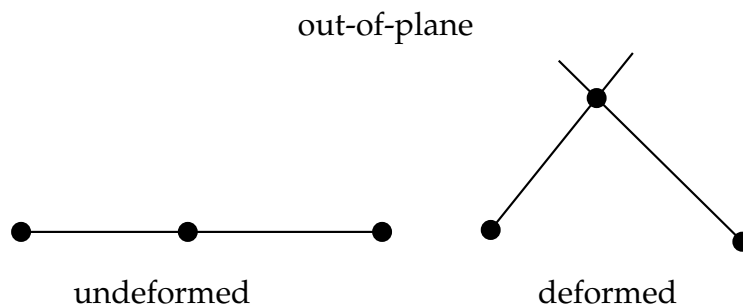


Figure 37.8. Refinement indicator based on angle change.

After the mesh refinement is determined, we can refine the mesh and continue the calculation or back up to an earlier time and repeat part of the calculation with the new mesh. For accuracy and stability reasons the latter method is generally preferred; however, the former method is preferred for speed. Whether LS-DYNA backs up and repeats the calculation or continues after remeshing is determined by an input parameter, ADPASS (see *CONTROL_ADAPTIVE). This feature is illustrated in [Figure 37.11](#).

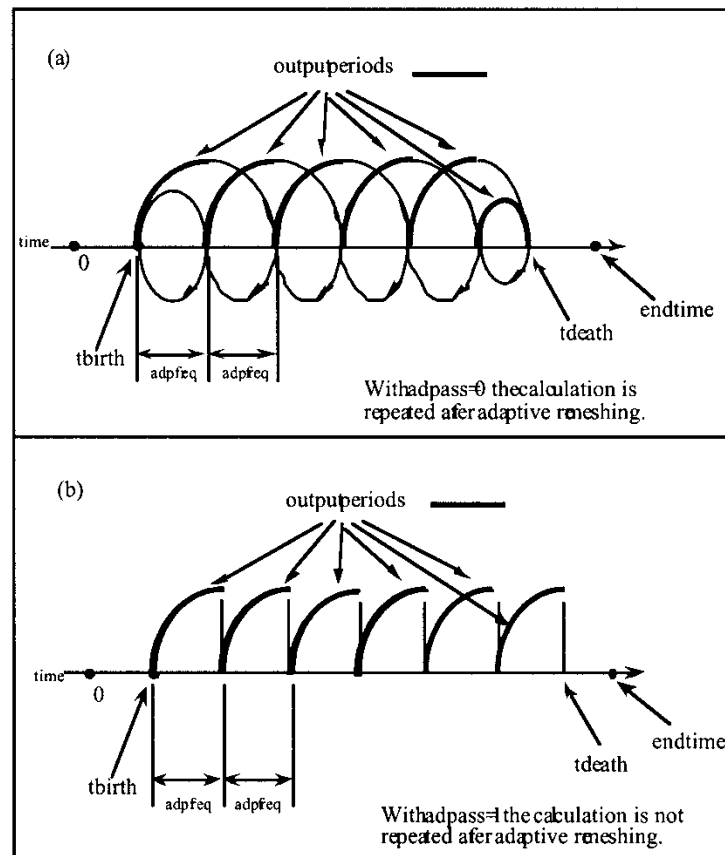


Figure 37.11. The input parameter, ADPASS, controls whether LS-DYNA backs up and repeats the calculation after adaptive refinement.

Mesh fusion was initially added to LS-DYNA for SMP between 2003 and 2005. It involves fusing portions of the mesh that were already fissioned but no longer need such a refined mesh. As of R11.0, it was added to the MPP version. Based on an extensive study of the MPP version by Fan et al [2017], the fusion feature reduces the computation time notably (average around 25%) and has little effects on formability analysis, such as thinning and effective strain predictions. The mesh fusion effect on springback prediction is found to be smaller than 10%. Therefore, you can apply fusion extensively in all formability related simulations since the leading formability indicators are little affected while the calculation can be sped-up by a factor of 25%. In springback simulations, however, you should use fusion with caution. Using this feature depends on the required

simulation accuracy. You can apply this feature if the springback results are to be used for a quick and rough estimation; however, if the results are to be used for compensating dies and for re-machining, then fusion may not be appropriate.

38

Implicit

38.1 Introduction

Implicit solvers are properly applied to static, quasi-static, and dynamic problems with a low frequency content. Such applications include but are not limited to

- Static and quasi-static structural design and analysis
- Metal forming, especially, the binderwrap and springback
- Gravitational loading of automotive structures
- Linear buckling and vibration analysis

An advantage of the implicit solver on explicit integration is that the number of load or time steps is typically 100 to 10000 times fewer. The major disadvantage is that the cost per step is unknown since the speed depends mostly on the convergence behavior of the equilibrium iterations which can vary widely from problem to problem.

An incremental-iterative numerical algorithm is implemented in LS-DYNA. The method is stable for wide range of nonlinear problems that involve finite strain and arbitrarily large rotations. Accuracy consideration usually limits the load increment or time step size. An inaccurate solution will often not converge. Nine iterative schemes are available including the full Newton method and eight quasi-Newton methods. These are:

- Full Newton,
- BFGS (default),
- Broyden,
- Davidon-Fletcher-Powell (DFP) [Schweizerhof 1986],
- Davidon symmetric, [Schweizerhof 1986],
- modified constant arc length with BFGS,
- modified constant arc length with Broyden's,

- modified constant arc length with DFP,
- modified constant arc length with Davidon.

A line search is combined with each of these schemes along with automatic stiffness reformations, as needed, to avoid non-convergence. LS-DYNA defaults to the BFGS quasi-Newton method which is the most robust although the other methods are sometimes superior. Generally, the quasi-Newton methods require fewer iterations than the modified Newton method since they exhibit superlinear local convergence due to the rank one or rank two updates of the stiffness matrix as the iterations proceed. In this chapter, important aspects of the static and dynamic algorithm are explained, hopefully, in a way that will be understandable to all users. The arc length methods are generally used in solving snap through buckling problems and details on one specific implementation is taken up in the next chapter.

38.2 Equations

38.2.1 Discretization

Neglecting constraints, discretization formally leads to the matrix equations of motion

$$\mathbf{R} = \mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_i - \mathbf{F}_e = \mathbf{0} \quad (38.1)$$

where

$$\begin{aligned} \ddot{\mathbf{x}} &= \text{acceleration vector of length } n \\ \mathbf{M} &= n \times n \text{ mass matrix} \\ \mathbf{F}_e &= \text{body force and external load vector of length } n \\ \mathbf{F}_i &= \text{internal force vector of length } n. \end{aligned}$$

It is implicitly assumed that the involved vectors \mathbf{F}_i and \mathbf{F}_e depend on⁸

$$\begin{aligned} \dot{\mathbf{x}}, \mathbf{x} &= \text{velocity and coordinate vectors of length } n \\ t &= \text{simulation time} \end{aligned}$$

as well as on some history of the deformation, while \mathbf{M} is constant. In practice, the only independent variable is \mathbf{x} since the velocity $\dot{\mathbf{x}}$ and acceleration $\ddot{\mathbf{x}}$ are typically expressed in terms of this coordinate vector by the time integration scheme used, and t is given. Deformation history is typically accounted for in internal variables associated with features in the model, such as plastic strains in materials and frictional sliding in contacts.

⁸ This is true prior to time discretization, vectors \mathbf{F}_i and \mathbf{F}_e depend on \mathbf{x} and on $\dot{\mathbf{x}}$ only through the exact time differentiation. Once time discretization is done according to some scheme, the dependence is on \mathbf{x} and not necessarily on the discretized $\dot{\mathbf{x}}$ but rather $\Delta\mathbf{x}/\Delta t$, see Section 1.4.1.

A diagonal lumped mass matrix is obtained by row summing according to

$$M_{ii} = \int_V \rho \phi_i \sum_j \phi_j dV = \int_V \rho \phi_i dV \quad (38.2)$$

where ρ is material density and V denotes the body over which integration occurs.

The primary nonlinearities, which are due to geometric effects and inelastic material behavior, are accounted for in F_i ,

$$F_i = \int_V B^T \sigma dV, \quad (38.3)$$

where B is the strain-displacement matrix and σ is the stress.

Additional nonlinearities arise in F_e due to geometry dependent applied loads, such as contacts and loads on segments.

Explicit integration trivially satisfies (38.1) since the calculation of the acceleration guarantees equilibrium, i.e., from time step j to $j + 1$ we use

$$\ddot{x}^j = M^{-1} [F_e^j - F_i^j]. \quad (38.4)$$

The explicit update of the velocities and coordinates is given by

$$\dot{x}^{j+1/2} = \dot{x}^{j-1/2} + \Delta t^j \ddot{x}^j, \quad (38.5)$$

$$x^{j+1} = x^j + \Delta t^{j+1/2} \dot{x}^{j+1/2}. \quad (38.6)$$

Stability places a limit on the time size. This step size may be very small and, consequently, a large number of steps may be required. Implicit analysis employs schemes that are unconditionally stable, thus allowing for larger steps at the cost of a more expensive update of the geometry.

38.2.2 Constraints

Constraints is obviously an important ingredient in nonlinear finite elements, and may include for instance simple point or motion constraints, slave nodes constrained to rigid bodies, joints and tied contacts. Constraints are divided into two categories, those that are directly eliminated (first kind) and those that are treated with Lagrangian multipliers λ (second kind). The principle behind (38.1) is that of virtual work, stating that

$$\delta x^T (R + C_\lambda^T \lambda) = 0 \quad (38.7)$$

for any *admissible* virtual displacement field δx . Admissible displacement fields are those that satisfy the first kind of constraints. The second kind is instead treated by the second term inside the parenthesis, for which we require satisfaction of m_λ additional constraints

$$H_\lambda(x, t) = 0 \quad (38.8)$$

and

$$C_\lambda = \frac{\partial H_\lambda}{\partial x}. \quad (38.9)$$

If there are no constraint of the first kind, any displacement field is admissible and hence $R + C_\lambda^T \lambda = 0$. The presence of first kind constraints put restrictions on δx and the following is an attempt to derive the proper nonlinear equations in this context.

Constraints of the first kind augment the n equations of motion by imposing m additional equations

$$H(x, t) = 0, \quad (38.10)$$

and consequently (38.1) is to be reduced to $n - m$ equations. To this end, we introduce the $m \times n$ constraint matrix

$$C = \frac{\partial H}{\partial x}, \quad (38.11)$$

and conveniently partition the global vector x into an independent (solution) part x_I of length $n - m$ and dependent part x_D of length m . This partitioning is represented by projection matrices P_I and P_D such that

$$x_I = P_I x, \quad x_D = P_D x \quad (38.12)$$

and

$$x = P_I^T x_I + P_D^T x_D, \quad (38.13)$$

and the space of admissible virtual displacements is that of any δx_I . The criterion for a valid partitioning is that the $m \times m$ matrix given by

$$C_D = C P_D^T, \quad (38.14)$$

is non-singular, which is always possible unless there are conflicting or redundant constraints. The Linear Constraint PACKage (LCPACK) in LS-DYNA performs this partitioning and invalid constraints will result in error termination. Introducing the $m \times (n - m)$ matrix

$$C_I = C P_I^T \quad (38.15)$$

we can combine the variations of (38.10) and (38.13) and use (38.14) and (38.15) to obtain

$$\delta x = P^T \delta x_I, \quad (38.16)$$

where

$$P = P_I - C_I^T C_D^{-T} P_D \quad (38.17)$$

is a projection-like matrix from the global variable space to the independent solution space. Using this expression for δx in the principle of virtual work (38.7), the arbitrariness of δx_I results in

$$Q = 0, \quad (38.18)$$

with

$$Q = P(R + C_\lambda^T \lambda). \quad (38.19)$$

This can be interpreted as to solve for zero force in the direction of deformation that is not constrained, in other directions the constraints induce reaction forces that are typically monitored in the LS-DYNA ascii and binout databases associated with the constraint type. The reaction forces corresponding to those constraints of the second kind is contained in λ .

38.2.3 Reaction forces due to constraints

Alternatively, we may form the lagrangian \mathcal{L} for the entire system as

$$\mathcal{L}(x, \lambda, t) = \mathcal{E}(x, t) + \lambda^T H(x, t) \quad (38.20)$$

where \mathcal{E} is an assumed potential for the residual force R . It is important to stress that the potential may not exist in general, this requires that the system is conservative in the sense of e.g. hyperelasticity. But it serves the purpose here for deriving the Lagrangian multiplier λ and subsequently the reaction forces due to the constraints H . The Karush-Kuhn-Tucker conditions for equilibrium is now (omitting arguments)

$$\begin{aligned} R + C^T \lambda &= 0 \\ H &= 0 \end{aligned} \quad (38.21)$$

where λ is a vector of length m to be interpreted as the resisting force needed to maintain the constraints. Continuing, we split R and C into their respective independent and dependent parts, leading to a rewrite of the first of (38.21)

$$\begin{aligned} R_I + C_I^T \lambda &= 0 \\ R_D + C_D^T \lambda &= 0 \end{aligned} \quad (38.22)$$

from which the Lagrangian multipliers can be solved from the second of (38.22) as

$$\lambda = -C_D^{-T} R_D. \quad (38.23)$$

This is the (generalized) force needed to enforce the second of (38.21). To obtain the corresponding nodal (reaction) force vector associated with a single constraint $j \in [1, m]$ we form

$$r_j = \lambda_j c_j \quad (38.24)$$

where c_j is the j :th row in the constraint matrix C , and λ_j is the corresponding component of λ . These are the equations that form the basis for the detailed database outputs, such as *bndout* in the case of prescribed motion.

38.2.4 Some common constraints

CONSTRAINED_INTERPOLATION (RBE3)

For the RBE3 constraint, we have one dependent node d (DNID) that is controlled by a set I (INID) of independent nodes. The constraint starts out from an overdetermined set of equations, for which each node $i \in I$ is attached to node d by a rigid link

$$\begin{aligned} \mathbf{v}_i &= \mathbf{v}_d + \boldsymbol{\omega}_d \times \mathbf{r}_i \\ \boldsymbol{\omega}_i &= \boldsymbol{\omega}_d \end{aligned} \quad (38.25)$$

where $\mathbf{v}_{i/d}$ and $\boldsymbol{\omega}_{i/d}$ denote translational and rotational velocities of the independent nodes and dependent node, respectively, and \mathbf{r}_i is the vector from node d to node i . This can be rewritten as a system of matrix vector equations

$$\mathbf{V}_i = \begin{pmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{R}_i \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{v}_d \\ \boldsymbol{\omega}_d \end{pmatrix} = \mathbf{M}_i \mathbf{V}_d. \quad (38.26)$$

Let L be a characteristic length, computed internally, and define

$$\mathbf{L} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & L\mathbf{I} \end{pmatrix} \quad (38.27)$$

which we use to premultiply (38.26), resulting in

$$\mathbf{L}\mathbf{V}_i = \mathbf{L}\mathbf{M}_i \mathbf{V}_d. \quad (38.28)$$

Furthermore, let \mathbf{W}_i be a diagonal matrix containing the weights (TWGHTX,...,RWGHTZ) associated with the degrees of freedom of independent node i , this is similarly used to premultiply (38.28)

$$\mathbf{W}_i \mathbf{L}\mathbf{V}_i = \mathbf{W}_i \mathbf{L}\mathbf{M}_i \mathbf{V}_d. \quad (38.29)$$

Collecting the equations for all independent nodes results in

$$\mathbf{V} = \mathbf{W}\mathbf{M}\mathbf{V}_d. \quad (38.30)$$

with

$$\mathbf{V}^T = \begin{pmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T & \dots & \mathbf{v}_n^T \end{pmatrix} \quad \mathbf{V}_d^T = \begin{pmatrix} \mathbf{v}_d^T & \boldsymbol{\omega}_d^T \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & & \\ & \mathbf{W}_2 & \\ & & \ddots \end{pmatrix} \quad \mathbf{M} = \begin{pmatrix} \mathbf{M}_1 & & \\ & \mathbf{M}_2 & \\ & & \ddots \end{pmatrix} \quad (38.31)$$

$$W = \begin{pmatrix} W_1 & \mathbf{0} & \dots \\ \mathbf{0} & W_2 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$M^T = (M_1^T L \quad M_2^T L \quad \dots).$$

Now premultiply (38.30) by M^T , then

$$M^T V = A V_d \tag{38.32}$$

with $A = M^T W M$. A singular value decomposition of A allows us rewrite this as

$$M^T V = Q \Lambda Q^T V_d \tag{38.33}$$

and

$$V_d = Q \Lambda^\dagger Q^T M^T V \tag{38.34}$$

which more or less completes the constraint. Here \dagger denotes the pseudo-inverse of a matrix, and the constraints that are actually applied correspond to those defined by the user, DDOF.

Now, assume that Λ is singular, which essentially means that the right hand side of (38.34) is incapable of spanning the space of a rigid body motion. More specifically, the subspace not spanned is q , corresponding to the eigenvectors in Q for which the singular value in Λ is zero. Looking at the left hand side of (38.34), we impose a constraint of V_d restricted to the subspace e defined by the user, DDOF. For the constraint to be valid, we must have $e \cap q = \emptyset$, which is checked when assembling the constraint matrix.

38.2.5 Inequality constraints

Inequality constraints cannot be incorporated in the framework presented above, since the state of inequalities (i.e., which ones of the constraints attain equality) is *a priori* unknown and part of the problem itself. A typical approach is to use Lagrangian multipliers for the inequalities and combine that with an active set strategy. An alternative is to reformulate the resulting KKT system in terms of a system of equalities with an augmentation of the constraints for numerical purposes. This is the proposed method here, and is described in the following.

For the sake of simplicity, we assume that there are no other constraints associated with the system, and impose m inequalities represented by the m -vector H . The problem is mathematically stated as

$$\begin{aligned}
R - C^T \lambda &= 0 \\
H &\leq 0 \\
\lambda &\leq 0 \\
H^T \lambda &= 0
\end{aligned} \tag{38.35}$$

where relation (38.11) applies. Note that H is a *displacement* vector while λ is a *force* vector. Now let's form G , which is a vector of the same size as H and λ , where each component is defined as

$$G_i = \kappa_i \max(H_i, \lambda_i), \quad i = 1, \dots, m. \tag{38.36}$$

Here κ_i is an arbitrary spring stiffness which converts λ to a *displacement* vector and assures that G is a *force* vector, and we let κ denote the diagonal matrix with those values on the diagonal. Furthermore, we use the subscripts H and λ to denote the submatrices of κ and C , and subvectors of G , H and λ where $G_i = H_i$ and $G_i \neq H_i$ in (38.36). We can now equivalently restate the problem as

$$\begin{aligned}
R + C_H^T \kappa_H (H_H - \lambda_H) &= 0 \\
-G &= 0
\end{aligned} \tag{38.37}$$

which is continuous (albeit not continuously differentiable) and can be solved by the methods presented herein. We emphasize that (38.35) and (38.37) are equivalent, with the exception of λ differing by the transformation using the matrix κ . The values used in κ determine the numerical sanity of the system, and should be chosen so that typical solution values in H and λ are of similar orders of magnitude. The stiffness matrix resulting from this formulation can be written as

$$K = \begin{pmatrix} \frac{\partial R}{\partial x} + C_H^T \kappa_H C_H & -C_H^T \kappa_H & 0 \\ -\kappa_H C_H & 0 & 0 \\ 0 & 0 & -\kappa_\lambda \end{pmatrix} \tag{38.38}$$

which is symmetric but indefinite.

CONSTRAINED_RIGID_BODY_STOPPERS

An example of inequality constraints is the use of rigid body stoppers, for which the motion of rigid bodies is limited as

$$H = \max(c^T u - u_{\max}, \Delta t \{c^T v - v_{\max}\}) \leq 0, \tag{38.39}$$

where c is a unit vector specifying the direction of stoppage, and u_{\max} and v_{\max} determine the upper bounds on displacements and velocities, respectively. We also use u as the displacement and v as the velocity of the rigid body, and Δt as the time step. Typically 2 constraints are associated with each rigid body stopper, one in positive and one in

negative direction, so the number of negative eigenvalues in the stiffness matrix increases accordingly.

38.3 Implicit Statics

38.3.1 Linearization

For the implicit static solution $M = 0$ in (38.1) and the residual and constraint vector at a given time t becomes an implicit function of x only. We seek the vector x and multiplier λ such that (38.8) and (38.10) holds together with

$$Q(x, \lambda) = 0. \quad (38.40)$$

Assume an approximation x^k to x and λ^k to λ for $k = 1, 2, 3 \dots$ etc. In the neighborhood of x^k and λ^k we use a linear approximation to (38.10) and (38.40) given by

$$K(x^k) \Delta x_I^k + P C_\lambda^T \Delta \lambda^k = F(x^k, \lambda^k), \quad (38.41)$$

and iterate for the solution

$$x_I^{k+1} = x_I^k + s \Delta x_I^k, \quad \lambda^{k+1} = \lambda^k + s \Delta \lambda^k \quad (38.42)$$

Here s is a convenient step size to be discussed below, as will be the update of the dependent part x_D^{k+1} . The linear system (38.41) is derived using the assumption that P is *independent* of x , which is generally not true. Many constraints in LS-DYNA are nonlinear and a strict linearization would have to take the second variation of constraints into account. But this would be inherently complex, thus linearizing (38.40) using (38.19) just becomes

$$Q(x^{k+1}, \lambda^{k+1}) \approx Q(x^k, \lambda^k) + P \frac{\partial R}{\partial x}(x^k) \Delta x^k + P C_\lambda^T \Delta \lambda^k = 0. \quad (38.43)$$

To be able to solve for Δx^k and $\Delta \lambda^k$ this needs first to be combined with a linearized equation of the constraint (38.10)

$$H(x^{k+1}) \approx H(x^k) + C(x^k) \Delta x^k = 0, \quad (38.44)$$

and an incremental correspondent to (38.13)

$$\Delta x^k = P_I^T \Delta x_I^k + P_D^T \Delta x_D^k. \quad (38.45)$$

These last two equations together with (38.14) and (38.15) can be used to yield an expression for the dependent part of the displacement increment

$$\Delta x_D^k = -C_D^{-1}(x^k) [H(x^k) + C_I(x^k) \Delta x_I^k]. \quad (38.46)$$

Using (20.22) in (38.45) in (38.43) results in expressions of the vector F and jacobian K in (38.41) as

$$F(x^k, \lambda^k) = -Q(x^k, \lambda^k) + P \frac{\partial R}{\partial x}(x^k) P_D^T C_D^{-1}(x^k) H(x^k) \quad (38.47)$$

and

$$K(\mathbf{x}^k) = \mathbf{P} \frac{\partial \mathbf{R}}{\partial \mathbf{x}}(\mathbf{x}^k) \mathbf{P}^T. \quad (38.48)$$

The dependent part of the solution vector is in general analogous to (38.42) of the independent part

$$\mathbf{x}_D^{k+1} = \mathbf{x}_D^k + s \Delta \mathbf{x}_D^k \quad (38.49)$$

except for those constraint equations where there is an explicit expression $\mathbf{x}_D = \mathbf{x}_D(\mathbf{x}_I)$, then of course

$$\mathbf{x}_D^{k+1} = \mathbf{x}_D(\mathbf{x}_I^{k+1}). \quad (38.50)$$

The constraint vector corresponding to those of the second kind needs also to be linearized,

$$\mathbf{H}_\lambda(\mathbf{x}^{k+1}) \approx \mathbf{H}_\lambda(\mathbf{x}^k) + \mathbf{C}_\lambda(\mathbf{x}^k) \Delta \mathbf{x}^k = \mathbf{0}, \quad (38.51)$$

and repeating the same procedure as above, we are lead to

$$\mathbf{C}_\lambda(\mathbf{x}^k) \mathbf{P}^T \Delta \mathbf{x}_I^k = \mathbf{G}(\mathbf{x}^k) \quad (38.52)$$

with

$$\mathbf{G}(\mathbf{x}) = -\mathbf{H}_\lambda(\mathbf{x}) + \mathbf{C}_\lambda(\mathbf{x}) \mathbf{P}_D^T \mathbf{C}_D^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x}), \quad (38.53)$$

which completes the linearization. In sum, the Newton step thus means solving the combined system (38.41) and (38.52) for $\Delta \mathbf{x}^k$ and $\Delta \lambda^k$ and update the solution using (38.42). For simplicity, we will hereforth assume $m_\lambda = 0$, i.e., all constraints are of the first kind, to simplify the exposition.

The jacobian matrix \mathbf{K} is the assembly of the tangent moduli of materials, external loads, contacts, etc., and is in practice only an approximation due to the complexity of taking all dependencies into account. In particular it does not include the geometric stiffness contribution from internal forces by default, see IGS on CONTROL_IMPLICIT_GENERAL. A speculative reason is that this has a smoothing effect and eliminates negative eigenvalues due to compressive stresses. If the deformation mode is known to be mainly in tension or if the material is hyper-elastic, including the geometric stiffness could improve convergence however. Often the stiffness matrix is assumed symmetric and positive definite, but is not limited to those characteristics. Note also that (38.47) indicates that both \mathbf{Q} and \mathbf{H} must vanish to render a zero \mathbf{F} , thus a zero displacement increment $\Delta \mathbf{x}$ in the iterative scheme.

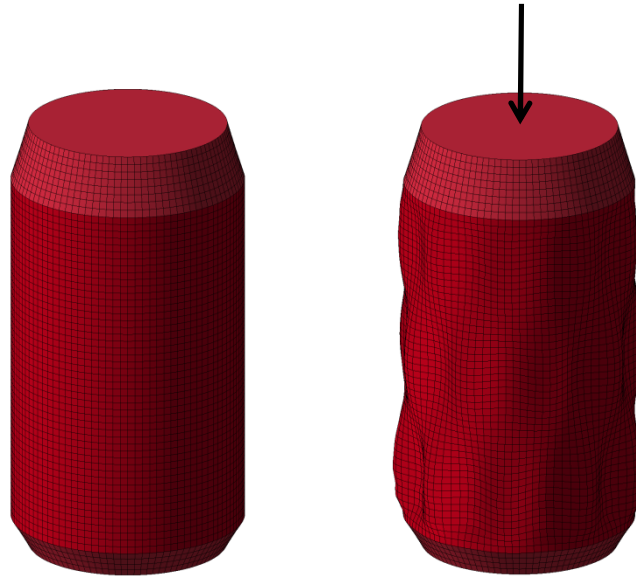


Figure 3838-1 Linear axial buckling of an aluminium beverage can.

38.3.2 Linear theory

For a linear solution, (38.41) is solved *once* to obtain the linear displacement vector $\mathbf{u} = \Delta \mathbf{x}$, and often the following assumptions apply. The configuration \mathbf{x} to which the linear approximation apply is stress free and all constraints are fulfilled, so the right hand side \mathbf{F} in (38.47) is essentially the external applied load \mathbf{F}_e , $\mathbf{F} = \mathbf{P}\mathbf{F}_e$, which is constant. Furthermore, \mathbf{P} is constant due to trivial constraints and the stiffness matrix \mathbf{K} in (38.48) thus evaluates from the linearization of internal forces

$$\mathbf{K} = \mathbf{P} \int_V \mathbf{B}^T \mathbf{E} \mathbf{B} dV \mathbf{P}^T, \quad (38.54)$$

where we used $\mathbf{E} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}$ to denote the constitutive matrix. So in more conventional terms, the linear equation is written

$$\mathbf{K}\mathbf{u} = \mathbf{F}. \quad (38.55)$$

Once solved, the stress

$$\boldsymbol{\sigma} = \mathbf{E}\mathbf{B}\mathbf{u} \quad (38.56)$$

can be evaluated from the constitutive law for the resulting deformation. The stiffness matrix \mathbf{K} in (38.55) is symmetric and positive definite if \mathbf{E} is and \mathbf{P} eliminates all rigid body modes, whence the solution \mathbf{u} is unique. Furthermore, linearity implies that substituting the right hand side for $\mathbf{F}_\lambda = \lambda \mathbf{F}$, the solution changes to $\mathbf{u}_\lambda = \lambda \mathbf{u}$ and the resulting stress changes to

$$\boldsymbol{\sigma}_\lambda = \lambda \boldsymbol{\sigma}. \quad (38.57)$$

A linear solution is obtained by putting NSOLVR = 1 on CONTROL_IMPLICIT_SOLUTION.

Now go back to the *nonlinear* static equation and scale a constant external load \mathbf{F}_e with λ

$$P[F_i - \lambda F_e] = 0 \quad (38.58)$$

and assume an updated configuration x_λ with resulting stress σ_λ to be the solution. To see how the solution changes with λ we can perturb (38.58) to obtain

$$P \left[\frac{\partial F_i}{\partial x} P^T \delta x_\lambda - \delta \lambda F_e \right] = 0 \quad (38.59)$$

where we applied (38.16). This is conveniently rewritten as

$$K_\lambda \delta x_\lambda = \delta \lambda F, \quad (38.60)$$

where the presence of stress will change the evaluation of the stiffness matrix from that in (38.54) to

$$K_\lambda = K + P \int_V \mathcal{B}^T \sigma_\lambda \mathcal{B} dV P^T. \quad (38.61)$$

Here the second term is the geometric contribution to the tangent stiffness matrix where \mathcal{B} is a matrix consisting of shape function derivatives for the finite element. We are interested in the question of uniqueness of δx_λ as the solution to (38.60), and this uniqueness is lost when

$$\det(K_\lambda) = 0. \quad (38.62)$$

If one assumes that deformations are small enough to keep V, B, E and \mathcal{B} independent of λ , and that (38.57) holds with (38.56) and (38.55), then (38.62) can be stated as the following eigenvalue problem

$$K \delta u = -\lambda K_\sigma \delta u \quad (38.63)$$

with the *geometric* (nonlinear) stiffness matrix given by

$$K_\sigma = P \int_V \mathcal{B}^T \sigma \mathcal{B} dV P^T \quad (38.64)$$

and K is the *material* (linear) stiffness matrix again given by (38.54). This is the theory behind linear buckling, see CONTROL_IMPLICIT_BUCKLE, with λ being the buckling load parameter and δu the associated buckling mode. Usually solutions with $\lambda > 0$ are of interest, and from (38.63) and (38.64) it then follows that the principal stresses cannot be positive throughout the domain of integration. In other words, the model must somehow be in a compressed state. The full procedure is

1. Assemble K by (38.54).
2. Solve (38.55) for a constant reference load F .
3. Evaluate resulting stress σ by (38.56).
4. Assemble K_σ by (38.64).
5. Solve (38.63) for λ and δu .

An example of linear buckling is shown in Figure 3838-1, simulating the stepping on an aluminium beverage can.

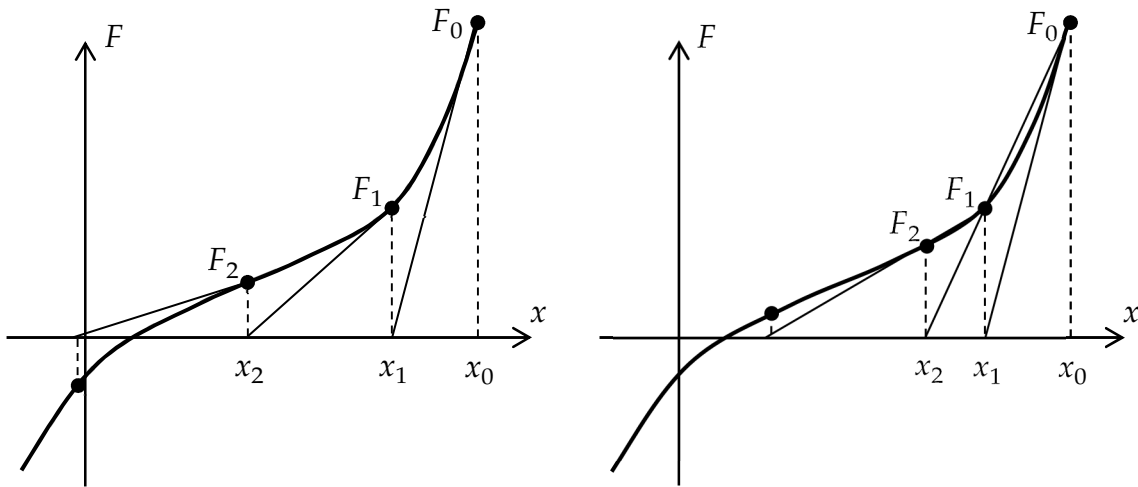


Figure 3838-2 Full Newton (left) compared to quasi-Newton secant updates (right), the linear approximation to obtain x_2 is for full Newton the exact tangent in (x_1, F_1) while it is the linear extension between the points (x_0, F_0) and (x_1, F_1)

38.3.3 Quasi-Newton iterations

Now back to the nonlinear theory. To obtain the solution at load increment $j + 1$ given the solution at load increment j , linearized equations of the form

$$K_k \Delta x_k = F_k, \quad (38.65)$$

are assembled and solved where k is the iteration number and

K_k = Tangent stiffness matrix
 Δx_k = Desired increment in displacements
 F_k = Residual load vector.

This should be seen a generalization of (38.41) in the sense that the tangent stiffness matrix K need not be calculated as (38.48) but can be based on other approximations to be presented. We do however assume, without loss of generality, that (38.65) is in the independent system of variables so the residual vector F is given by (38.47). We here just substitute the sub-index I in the incremental displacement Δx for the iteration counter k to simplify the notation in the following. The coordinate vector is updated

$$x_{k+1} = x_k + s \Delta x_k, \quad (38.66)$$

where s is a parameter between 0 and 1 found from a line search.

If the tangent stiffness matrix is calculated as (38.48) for each k then this is termed a *full Newton* method, but it may be beneficial to use so called *quasi-Newton* updates of K^{-1} to avoid the cost of solving a linear system of equations in each iteration. Four such methods for updating the stiffness matrix are available

- Broyden's first method

- Davidson
- DFP
- BFGS

and these involve rank 1 or rank 2 stiffness updates. Quasi-Newton methods are less expensive than the full Newton method but often still result in robust program. In one dimension, quasi-Newton corresponds to secant iterations and this method compared to full Newton is illustrated in [Figure 3838-2](#). Henceforth we assume that K_0 represents the last assembled matrix according to (38.48) and then K_k , $k = 1, 2, 3, \dots$ are quasi-Newton updates to be given in the following.

The secant matrices K_k are found via the quasi-Newton equations

$$K_k \Delta x_{k-1} = \Delta F_k, \quad (38.67)$$

where

$$\Delta F_k = F_{k-1} - F_k. \quad (38.68)$$

Two classes of matrix updates that satisfy the quasi-Newton equations are of interest; rank 1 updates

$$K_k = K_{k-1} + \alpha z y^T, \quad (38.69)$$

and rank 2 updates

$$K_k = K_{k-1} + \alpha z y^T + \beta v u^T. \quad (38.70)$$

Substituting (38.69) into (38.67) gives

$$K_{k-1} \Delta x_{k-1} + \alpha z y^T \Delta x_{k-1} = \Delta F_k. \quad (38.71)$$

By choosing $\alpha = \frac{1}{y^T \Delta x_{k-1}}$ and $z = -F_k$, [equation \(38.67\)](#) is satisfied. Note that y is an arbitrary vector but is restricted such that

$$y^T \Delta x_{k-1} \neq 0. \quad (38.72)$$

The tangent stiffness update is

$$K_k = K_{k-1} - \frac{F_k y^T}{y^T \Delta x_{k-1}}, \quad (38.73)$$

resulting generally in non-symmetric secant matrices. The inverse forms are found by the Sherman-Morrison formula

$$(A + a b^T)^{-1} = A^{-1} - \frac{A^{-1} a b^T A^{-1}}{1 + b^T A^{-1} a}. \quad (38.74)$$

where A is a nonsingular matrix and a and b are arbitrary vectors such that $1 + b^T A^{-1} a \neq 0$. The inverse form for (38.73) can be found by letting

$$A = K_{k-1}, \quad a = \frac{-F_k}{y^T \Delta x_{k-1}}, \quad b = y, \quad (38.75)$$

in the Sherman-Morrison formula. Therefore,

$$K_k^{-1} = \left[I + \frac{[K_{k-1}^{-1} F_k] y^T}{y^T \Delta F_k} \right] K_{k-1}^{-1}. \quad (38.76)$$

Broyden's method use $y = \Delta x_{k-1}$ resulting in non-symmetric stiffness updates, and an algorithm for this exploits that

$$K_k^{-1} = \left[I + \frac{d_k \Delta x_{k-1}^T}{\gamma_k} \right] \left[I + \frac{d_{k-1} \Delta x_{k-2}^T}{\gamma_{k-1}} \right] \dots \left[I + \frac{d_1 \Delta x_0^T}{\gamma_1} \right] K_0^{-1} \quad (38.77)$$

where

$$d_k = K_{k-1}^{-1} F_k, \quad \gamma_k = \Delta x_{k-1}^T \Delta F_k. \quad (38.78)$$

To illustrate the efficiency of a quasi-Newton method we outline the steps to obtain Δx_k .

1. Assume Δx_0 , F_k and F_{k-1} (recalling (38.68)) are known, as well as Δx_i , d_i and γ_i for $i = 1, \dots, k-1$.
2. Solve $e_0 = K_0^{-1} F_k$.
3. Recursively compute $e_i = e_{i-1} + \frac{\Delta x_{i-1}^T e_{i-1}}{\gamma_i} d_i$ for $i = 1, \dots, k-1$.
4. Set $d_k = e_{k-1}$ and $\gamma_k = \Delta x_{k-1}^T \Delta F_k$.
5. Compute $\Delta x_k = \left[1 + \frac{\Delta x_{k-1}^T d_k}{\gamma_k} \right] d_k$.

Noteworthy here is that an iteration consists of a forward and backward substitution of an already factorized matrix (step 2) plus a sequence of vector operations (steps 3-5), which altogether presumably is much less expensive than solving an entire system of linear equations. The algorithm requires an in-core storage consisting of $2k$ vectors and k scalars to complete iteration k .

Again, recalling the quasi-Newton equation (38.67), and substituting (38.70) gives

$$K_{k-1} \Delta x_{k-1} + \alpha z y^T \Delta x_{k-1} + \beta v u^T \Delta x_{k-1} = \Delta F_k, \quad (38.79)$$

and set $\alpha = \frac{1}{y^T \Delta x_{k-1}}$, $z = -F_{k-1}$, $\beta = \frac{1}{u^T \Delta x_{k-1}}$ and $v = \Delta F_k$. Here y and u are arbitrary vectors that are non-orthogonal to Δx_{k-1} , i.e.,

$$y^T \Delta x_{k-1} \neq 0, \quad (38.80)$$

and

$$u^T \Delta x_{k-1} \neq 0. \quad (38.81)$$

In the BFGS method

$$y = F_{k-1} \quad u = \Delta F_k, \quad (38.82)$$

which leads to the following update formula

$$K_k = K_{k-1} + \frac{\Delta F_k \Delta F_k^T}{\Delta x_{k-1}^T \Delta F_k} - \frac{F_{k-1} F_{k-1}^T}{\Delta x_{k-1}^T F_{k-1}}, \quad (38.83)$$

that preserves symmetry of the secant matrix. A double application of the Sherman-Morrison formula then leads to the inverse form.

Special product forms have been derived for the DFP and BFGS updates and exploited by Matthies and Strang [1979],

$$\mathbf{K}_k^{-1} = [\mathbf{I} + \mathbf{q}_k \mathbf{p}_k^T] \mathbf{K}_{k-1}^{-1} [\mathbf{I} + \mathbf{p}_k \mathbf{q}_k^T]. \quad (38.84)$$

The primary advantage of the product form is that the determinant of \mathbf{K}_k and therefore, the change in condition number can be easily computed to control updates. The updates vectors are defined as

$$\mathbf{p}_k = -\Delta \mathbf{F}_k - \mathbf{F}_{k-1} \sqrt{\frac{\Delta \mathbf{x}_{k-1}^T \Delta \mathbf{F}_k}{\Delta \mathbf{x}_{k-1}^T \mathbf{F}_{k-1}}}, \quad (38.85)$$

$$\mathbf{q}_k = \frac{\Delta \mathbf{x}_{k-1}}{\Delta \mathbf{x}_{k-1}^T \Delta \mathbf{F}_k}. \quad (38.86)$$

Noting that the determinant of \mathbf{K}_k is given by

$$\det(\mathbf{K}_k) = \det(\mathbf{K}_{k-1}) [1 + \mathbf{q}_k^T \mathbf{p}_k]^2, \quad (38.87)$$

it can be shown that the change in condition number, c , is

$$c = \frac{\left[\sqrt{\mathbf{p}_k^T \mathbf{p}_k \mathbf{q}_k^T \mathbf{q}_k} + \sqrt{\mathbf{p}_k^T \mathbf{p}_k \mathbf{q}_k^T \mathbf{q}_k + 4(1 + \mathbf{p}_k^T \mathbf{q}_k)} \right]^4}{\left[4(1 + \mathbf{p}_k^T \mathbf{q}_k) \right]^2}. \quad (38.88)$$

Following the approach of Matthies and Strang [1979] this condition number is used to decide whether or not to do a given update. The quasi-Newton condition (38.67) is easily verified using (38.84) for a real non-singular tangent matrix \mathbf{K}_{k-1} and it follows that BFGS preserves not only symmetry but also positive definiteness. Interestingly enough the converse is not true; if \mathbf{K}_{k-1} is indefinite then \mathbf{K}_k may still be positive definite.

An algorithm would utilize that

$$\mathbf{K}_k^{-1} = [\mathbf{I} + \mathbf{q}_k \mathbf{p}_k^T] \dots [\mathbf{I} + \mathbf{q}_1 \mathbf{p}_1^T] \mathbf{K}_0^{-1} [\mathbf{I} + \mathbf{p}_1 \mathbf{q}_1^T] \dots [\mathbf{I} + \mathbf{p}_k \mathbf{q}_k^T], \quad (38.89)$$

so the computation of $\Delta \mathbf{x}_k = \mathbf{K}_k^{-1} \mathbf{F}_k$ requires a sequence of vector operations, followed by a forward and backward substitution of a factorized matrix, followed by yet another sequence of vector operations. The BFGS algorithm requires an in-core storage of the vectors \mathbf{p}_i and \mathbf{q}_i , $i = 1, \dots, k-1$, and $\Delta \mathbf{x}_{k-1}$ and \mathbf{F}_{k-1} to complete iteration k . It is the default and most robust quasi-Newton algorithm for the nonlinear implicit solver and there is no reason to switch. The only practical issue is how to control when to reform the tangent stiffness matrix, which typically depends on the degree of nonlinearity for the problem at hand. In fact, if LCPACK = 2 on CONTROL_IMPLICIT_SOLVER, i.e., working with independent variables only, BFGS is mandatory and comes in two flavors, NSOLVR = 2 or NSOLVR = 12 on CONTROL_IMPLICIT_SOLUTION.

There is also the option LCPACK = 3, for which LS-DYNA uses a non-symmetric matrix assembly when solving the linear system of equations. This adds to the computational cost for each iteration but may improve convergence because of better tangents of certain

features. The non-symmetric solver is illustrated in Figure 3838-3 for simulating a clamped cantilever beam subject to a follower load. In general the stiffness matrix is symmetric when the force is derived from an energy potential, or in other words is *conservative*. This is for instance the case for a hyperelastic material response or a physically admissible pressure load, see Schweizerhof and Ramm [1984] for a discussion. Non-symmetry arise e.g. from non-conservative forces, such as frictional contact, or physically inadmissible design dependent loads. The example in Figure 3838-3 serves as one of the latter since the load is not coming from a physical source, such as a water pressure, and it should be seen as an academic example to prove a point.

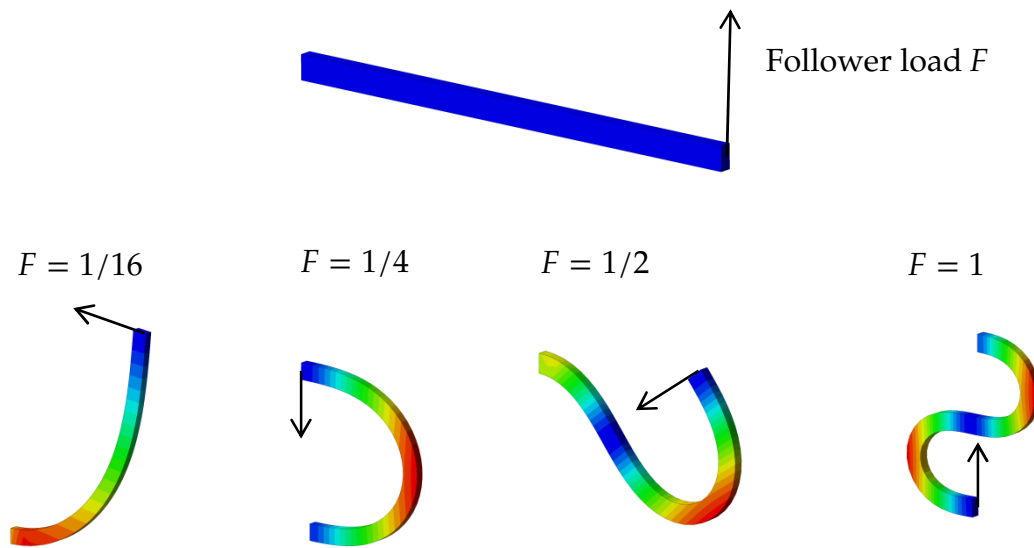


Figure 3838-3 Nonlinear implicit solution of an elastic cantilever beam with a follower force, von Mises stress is fringed..

38.3.4 Tangent stiffness reformations

In the previous section we have used K_0 in (38.77) and (38.89) to represent the last assembled tangent stiffness matrix according to (38.48). While the quasi-Newton updates K_k , $k = 1, 2, \dots$ are justified in the context of moderate nonlinearities and/or for a few iterations, it will eventually break down convergence for more complex problems. Whence there are criteria for reforming the tangent stiffness matrix and start over with the quasi-Newton (BFGS) updates.

The first criterion is simply a user defined upper limit of k , governed by ILIMIT on CONTROL_IMPLICIT_SOLUTION. Using ILIMIT = 1 means that no quasi-Newton updates are performed and should be used for highly nonlinear problems, larger values of ILIMIT can be used if the nonlinearities are considered less severe. ILIMIT = 11 is the default and is a reasonable value to use for starters.

A second criterion is that of increased residual norm, i.e., if any of the quantities

$$T_k = \|J_t F_k\|_1 \quad (38.90)$$

or

$$R_k = \|J_r F_k\|_1 \quad (38.91)$$

for some k is their respective largest attained since start iterating, the stiffness matrix is reformed. Here J_t and J_r are diagonal matrices with ones or zeros on the diagonal that extract the translational and rotational degrees of freedom, respectively, and one speaks of translational or rotational divergence. The notation $\|F\|_1$ indicates the L_1 -norm of F , meaning the sum of the absolute value of its components.

A third criterion is called energy explosion, if

$$\log(1 + |\Delta x_{k-1}^T J_t F_{k-1}|) + \log(1 + |\Delta x_{k-1}^T J_r F_k|) > 36 \quad (38.92)$$

or

$$\log(1 + |\Delta x_{k-1}^T J_r F_{k-1}|) + \log(1 + |\Delta x_{k-1}^T J_t F_k|) > 36 \quad (38.93)$$

the stiffness matrix is reformed. One can speak also here of translational and rotational energy explosion and this simply indicates that something really bad has happened with the last search direction.

Finally, a necessary criterion for continuing with BFGS updates is deemed

$$\Delta x_k^T F_k > 0 \quad (38.94)$$

since this would otherwise indicate a search direction corresponding to a zero or negative eigenvalue in the tangent matrix. If this criterion is violated the tangent stiffness matrix is reformed with a warning message issued that negative energy is detected during quasi-Newton updates. This can only happen if there are negative eigenvalues in the tangent stiffness matrix in the first place.

38.3.5 Line search

Line search is critical for decent convergence characteristics, and there are several criteria to choose from, see LSMTD and LSTOL on CONTROL_IMPLICIT_SOLUTION. One of the criteria available is that the norm of residual F must somehow decrease, LSMTD = 2, but this will inevitably lead to ridiculously small steps and is not recommended or presented further here. A more useful criterion is instead that an iterate $k + 1$ is accepted if

$$|\Delta x_k^T F_{k+1}| \leq \varepsilon_s \Delta x_k^T F_k \quad (38.95)$$

where ε_s is the line search convergence tolerance LSTOL. See [Figure 3838-4](#) for an illustration. This criterion is derived from a hypothetical assumption of the existence of an energy potential $W(x_k + s\Delta x_k)$ for the residual force F and that the potential is minimized along the search direction

$$\frac{\partial W}{\partial s} = \frac{\partial W}{\partial x} \frac{\partial x}{\partial s} = F^T \Delta x_k = 0. \quad (38.96)$$

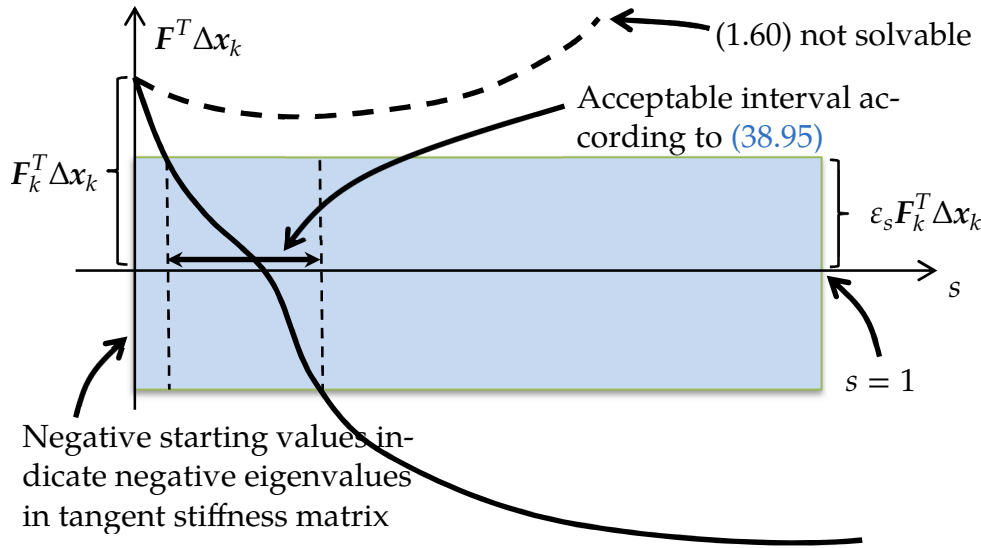


Figure 3838-4 Illustration of line search based on energy, may be complemented with norm of F . Dashed line indicates a line search that requires

So in a sense, (38.95) is the solution of (38.96) to within a certain tolerance and a Ridder algorithm is used to narrowing in on a candidate. It is not unusual however that (38.95) is not well-posed, an indefinite stiffness matrix may result in a negative right hand side or the search direction may be bad enough to not render a solution, see Figure 3838-4. In those special cases some conservative approach is taken to provide a reasonable iterate $k + 1$. This summarizes in brief LSMTD = 4.

A closer examination (and numerical experiments) reveals that F_{k+1} is not necessarily bounded by (38.95). Therefore this criterion can be complemented with

$$\|F_{k+1}\|_1 \leq [1 + \varepsilon_s] \|F_k\|_1. \quad (38.97)$$

where again $\|F\|_1$ means the L_1 -norm of F . This option is invoked with LSMTD = 5 and means that both (38.95) and (38.97) are to be satisfied for accepting an iterate $k + 1$. While this is a more robust approach, it also requires more residual force evaluations and is not recommended as default. It has proved to work well for implicit rubber simulations and complicated contact problems.

38.3.6 Convergence check

Convergence criteria are based on displacement, energy and residual force quantities. The exact definitions of norms and scalar products used depend to a great extent on parameter settings on CONTROL_IMPLICIT_SOLUTION, in particular on the parameter NLNORM. Historically it has been assumed that rotational degrees of freedom are not appropriate for checking convergence and therefore only translational degrees of freedom have been considered. With advancements in the development of the implicit solver, this is today considered an old fashioned way of thinking, but backward compatibility is maintained and the user is referred to the keyword manual for information regarding the choices in this context. Here the principles behind convergence checks are

presented from a pragmatic standpoint assuming the following generic displacement and force norms and (energy) scalar products

$$\|\Delta \mathbf{x}\| = \sqrt{\Delta \mathbf{x}^T \mathbf{J} \mathbf{J} \Delta \mathbf{x}}, \quad \|\mathbf{F}\|_1 = \sum_i |J_{ii}^\dagger F_i|, \quad \langle \Delta \mathbf{x}, \mathbf{F} \rangle = \Delta \mathbf{x}^T \mathbf{J} \mathbf{J}^\dagger \mathbf{F}. \quad (38.98)$$

Note that the norm of the incremental displacement is a L_2 -norm (euclidian norm) while the norm of the residual force is a L_1 -norm, this decision was made to allow for a more physical and less mesh dependent interpretation of this latter quantity. Here \mathbf{J} is a diagonal matrix that appropriately scales the various degrees of freedom to account for units, i.e., scaling radians with some characteristic length for consistency. It also accounts for user input in the sense that \mathbf{J} may contain zeros on the diagonal to indicate that rotational degrees of freedom should not be accounted for. \mathbf{J}^\dagger is the pseudo-inverse of \mathbf{J} , a notation introduced since \mathbf{J} actually may be singular due to the zeros on the diagonal. For some convergence option check is performed on translational and rotational degrees of freedom separately, we don't elaborate on such details here but instead refer to the keyword manual. In (38.98), the non-bold quantities with subscripts indicate components of the corresponding bold-faced vector.

Convergence for accepting an iterate $j + 1$ is assumed if the three conditions

$$\|\Delta \mathbf{x}_k\| < \max(\varepsilon_d u_{\max}, \sqrt{\max(\varepsilon_a, 0)}) \quad (38.99)$$

$$\langle \Delta \mathbf{x}_k, \mathbf{F}_k \rangle < \max(\varepsilon_e e_0, 10000 \max(\varepsilon_a, 0)) \quad (38.100)$$

$$\|\mathbf{F}_k\|_1 < \max(\varepsilon_r f_0, 10000 \max(\varepsilon_a, 0)) \quad (38.101)$$

are satisfied simultaneously. Here ε_d , ε_e , ε_r and ε_a are the displacement, energy, residual and absolute tolerances corresponding to DCTOL, ECTOL, RCTOL and ABSTOL, respectively. In the right hand side of (38.99), $u_{\max} = \|\mathbf{u}_{\max}\|$ is the maximum attained displacement in any iteration k measured from the position at the start of this implicit step j if DNORM = 1 and from the start of the simulation if DNORM = 2. In the right hand side of (38.100), $e_0 = \langle \Delta \mathbf{x}_0, \mathbf{F}_0 \rangle$ where $\Delta \mathbf{x}_0$ and \mathbf{F}_0 are the first incremental displacement and residual vectors for this implicit step j . Likewise $f_0 = \|\mathbf{F}_0\|_1$ in the right hand side of (38.101) is the norm of the first residual vector for this implicit step j . If \mathbf{J} has full rank and the tangent stiffness matrix is symmetric and positive definite then these conditions make intuitive sense, small changes in displacements and/or small residual forces indicate that an iterate is "close-enough" to the solution. Under these conditions the left hand side of (38.100) define norms in both the incremental displacement $\Delta \mathbf{x}_k$ and residual force \mathbf{F}_k , and $e_0 > 0$.

Optionally, these convergence criteria can be combined with bounds on the maximum norms, here denoted

$$\|\Delta \mathbf{x}\|_\infty = \max_i |\Delta \mathbf{x}|_i, \quad \|\mathbf{F}\|_\infty = \max_i |J_{ii}^\dagger F_i|, \quad \langle \Delta \mathbf{x}, \mathbf{F} \rangle_\infty = \max_i |\Delta \mathbf{x}^T \mathbf{J} \mathbf{J}^\dagger F_i|, \quad (38.102)$$

where the max is taken over all nodes and rigid bodies in the model, i.e., i ranges over all nodes and rigid bodies. If any of these options are activated, then convergence for

accepting an iterate $j + 1$ is assumed if the three conditions (38.99), (38.100) and (38.101) are satisfied *and*

$$\|\Delta \mathbf{x}_k\|_\infty < \varepsilon_d^\infty u_{\max}^\infty \quad (38.10 \text{ 3})$$

$$\langle \Delta \mathbf{x}_k, \mathbf{F}_k \rangle_\infty < \varepsilon_e^\infty e_0^\infty \quad (38.10 \text{ 4})$$

$$\|\mathbf{F}_k\|_\infty < \varepsilon_r^\infty f_0^\infty \quad (38.10 \text{ 5})$$

are satisfied. Here ε_d^∞ , ε_e^∞ and ε_r^∞ are the displacement, energy and residual tolerances corresponding to DMTOL, EMTOL and RMTOL, respectively, and if any of these parameters are zero that means the condition is not active. The remaining parameters on the right hand sides are interpreted in analogy to those appearing in the right hand sides of (38.99), (38.100) and (38.101), except for that the Euclidian norm is now substituted for the maximum norm.

A backdoor out is an absolute convergence test on the maximum norm on translational displacement. That is, convergence is always detected if the maximum value of any translational degree of freedom in the incremental displacement array $\Delta \mathbf{x}_k$ is smaller than $d\sqrt{\max(\varepsilon_a, 0)}$, where d is a characteristic size calculated as the diagonal of the smallest box aligned along the global coordinate system that encapsulates the model. Setting ABSTOL to a negative number is an alternative, then all absolute convergence checks above become inactive and instead the backdoor out is to assume convergence when $\|\mathbf{F}_k\|_1 < \max(0, -\varepsilon_a)$. This latter option requires to some extent an *a priori* knowledge of the force level and may require monitoring the residual norm for a few trial iterations to pick a decent ε_a , but it could be a sensible choice if the problem shows erratic behavior in displacement and energy due to severe nonlinearities. On the other hand, since it is a check on the L_1 -norm, the value ε_a might be taken as some reasonable fraction of an external load or as an desired error on the sum of imbalanced forces.

Another criterion that must be met for convergence is that simple prescribed motion constraints on nodes and rigid bodies, if they exist, must be “almost” satisfied. The background is that only partial line searches ($s < 1$) will not satisfy simple motion constraints, and at least one full line search step ($s = 1$) must be accomplished during the iterations. For difficult problems, this sometimes never happens and therefore convergence is prevented due to unfulfilled boundary conditions until all prescribed motion is satisfied to within 1%.

38.3.7 Automatic time stepping

If convergence is not attained after reforming the stiffness matrix a given amount of times, see MAXREF on CONTROL_IMPLICIT_SOLUTION, one of two things will happen. By default, LS-DYNA terminates with an error message, but if automatic time stepping is turned on, see IAUTO on CONTROL_IMPLICIT_AUTO, then LS-DYNA will try to resolve the problem with a smaller step size. More specifically, if convergence is not

attained with a time step Δt_{old} , then LS-DYNA backs up to beginning of step and retries with a time step

$$\Delta t_{\text{new}} = \max(\Delta t_{\text{min}}, 10^{-0.3} \Delta t_{\text{old}}) \quad (38.106)$$

where Δt_{min} is a user defined minimum step. If $\Delta t_{\text{old}} = \Delta t_{\text{min}}$ when attempting this then LS-DYNA will terminate with an error.

With the automatic time stepper turned on, LS-DYNA will not only cut the time step for convergence failure but also adjust the time step when converging. The adjustment is based on a user defined iteration window, i.e., a range of iteration numbers that are deemed acceptable for convergence, and if the number of iterations for convergence falls outside this window the next time step will be adjusted as follows. If convergence is attained for more iterations than acceptable then the time step for the next implicit step is given by (38.106). If instead convergence is attained for fewer iterations then the next time step is given by

$$\Delta t_{\text{new}} = \min(\Delta t_{\text{max}}, 10^{0.2} \Delta t_{\text{old}}) \quad (38.107)$$

where Δt_{max} is a maximum defined step. In this way LS-DYNA narrows in on an optimum time step for which the number of iterations to converge falls within the specified window.

There is much more information regarding this option in the keyword manual and the user is referred thereto for practical issues.

38.4 Implicit Dynamics

38.4.1 Newmark time integration

Nonlinear implicit dynamics principally follows the exact same solution algorithm as for nonlinear implicit statics, the only difference is that dynamic terms are included in the residual force. That is, (38.1) now reads

$$\mathbf{R} = \mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{F}_i - \mathbf{F}_e = \mathbf{0}, \quad (38.108)$$

where global damping is introduced through the matrix \mathbf{D} , so the residual is now essentially a function of \mathbf{x} , $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$. The dependence on the latter two is eliminated by introducing the Newmark time integration scheme

$$\ddot{\mathbf{x}} = \frac{\Delta \mathbf{x}}{\beta \Delta t^2} - \frac{\dot{\mathbf{x}}^j}{\beta \Delta t} - \frac{1}{\beta} \left(\frac{1}{2} - \beta \right) \ddot{\mathbf{x}}^j, \quad (38.109)$$

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}^j + \Delta t(1 - \gamma)\ddot{\mathbf{x}}^j + \gamma \Delta t \ddot{\mathbf{x}}, \quad (38.110)$$

$$\mathbf{x} = \mathbf{x}^j + \Delta \mathbf{x}. \quad (38.111)$$

Here, Δt is the time step size, β and γ are the free parameters of integration and we have used $\Delta \mathbf{x}$ to denote the total displacement from step j to step $j + 1$. For $\gamma = 1/2$ and $\beta = 1/4$ the method becomes the trapezoidal rule and is energy conserving. If

$$\gamma > \frac{1}{2}, \quad (38.112)$$

$$\beta > \frac{1}{4} \left(\frac{1}{2} + \gamma \right)^2, \quad (38.113)$$

numerical damping is induced into the solution leading to a loss of energy and momentum. By inserting (38.109) and (38.110) into (38.108) and using (38.111) to eliminate $\Delta \mathbf{x}$, we are back to an equation in \mathbf{x} only and can apply the algorithm starting with (38.40) and everything thereafter holds.

38.4.2 Energy balance

Following the notation in this section, the kinetic energy at time T is in principle given as the instantaneous value

$$E_k = \frac{1}{2} \mathbf{v}^T \mathbf{M} \mathbf{v}, \quad (38.114)$$

where \mathbf{v} is the nodal velocity vector and \mathbf{M} is the mass matrix at that particular point in time. This is what is output as the *Kinetic Energy* in glstat and matsum output files, and we note that \mathbf{v} is *not* the Newmark velocity $\dot{\mathbf{x}}$, but rather the velocity corresponding to

$$\mathbf{v} = \frac{\Delta \mathbf{x}}{\Delta t} \quad (38.115)$$

from (38.109)-(38.111). As an alternative, we can integrate (38.108) multiplied by \mathbf{v} over time to see that

$$\begin{aligned} 0 &= \int_0^T \mathbf{v}^T (\mathbf{M} \ddot{\mathbf{x}} + \mathbf{D} \dot{\mathbf{x}} + \mathbf{F}_i - \mathbf{F}_e) dt \\ &= \int_0^T \mathbf{v}^T \mathbf{M} \ddot{\mathbf{x}} dt + \int_0^T \mathbf{v}^T \mathbf{D} \dot{\mathbf{x}} dt + \int_0^T \mathbf{v}^T \mathbf{F}_i dt - \int_0^T \mathbf{v}^T \mathbf{F}_e dt \end{aligned} \quad (38.116)$$

which is the conservation of energy. We can identify the four terms at the end as *kinetic* (W_k), *damping* (W_d), *internal* (W_i) energy and *external work* (W_e), respectively. Given that the numerical problem is solved with high enough precision, it is guaranteed that (38.116) holds true. But here we see that the kinetic energy can also be written as

$$W_k = \int_0^T \mathbf{v}^T \mathbf{M} \ddot{\mathbf{x}} dt, \quad (38.117)$$

which should be interpreted as the work done by the inertial forces. The natural question now arises; under what conditions do we have $E_k \approx W_k$? The answer is that a consistent numerical time integration scheme, i.e., \mathbf{v} is close to $\dot{\mathbf{x}}$, in combination with a sufficiently small time step size Δt must allow for an accurate numerical integration of (38.117). This can be used as an assessment of the implicit solution in general.

Let

$$D_k = W_k - E_k \quad (38.118)$$

be the *numerically dissipated kinetic energy* which is currently stored as *Dissipated Kinetic Energy* in the output files. Ideally, this energy should be small when compared to the

kinetic energy E_k or otherwise energy is somehow lost or gained in the numerical treatment.

A similar reasoning can be made for the internal energy, by replacing (38.114) with

$$E_i = \iint \sigma : \dot{\epsilon} dt d\Omega, \quad (38.119)$$

and

$$W_i = \int_0^T v^T F_i dt. \quad (38.120)$$

The *numerically dissipated internal energy*

$D_i = W_i - E_i$ is stored as *Dissipated Internal Energy* in the output files. If this quantity is large, then either the nonlinearities in elements and/or materials aren't resolved sufficiently or the element violates the virtual energy principles.

In sum, there are three checks that can be made

1. $|D_k| \ll E_k \rightarrow$ Time integration scheme is energy conserving
2. $|D_i| \ll E_i \rightarrow$ Internal forces are trustworthy
3. $E_k + D_k + E_i + D_i + W_d \approx W_e \rightarrow$ Numerical problem solved accurately enough

38.4.3 Practical considerations

Linearizing (38.108) using (38.109), (38.110) and (38.111) results in a tangent matrix

$$\frac{\partial \mathbf{R}}{\partial \mathbf{x}} = \frac{\mathbf{M}}{\beta \Delta t^2} + \frac{\gamma \mathbf{D}}{\beta \Delta t} + \frac{\partial [\mathbf{F}_i - \mathbf{F}_e]}{\partial \mathbf{x}}, \quad (38.121)$$

and leads to an interesting observation. Since \mathbf{M} is symmetric and positive definite the first term assures that the resulting tangent is positive definite as long as the time step Δt is sufficiently small, so including dynamics will enhance the robustness of the numerical procedure. In fact, the eigenvalues of the resulting tangent can be made arbitrarily large by decreasing Δt at the cost of requiring more steps to obtain the solution.

Not only robustness but also the accuracy in dynamic implicit depends on the size of the time step Δt , roughly speaking only frequencies up to $\sim \Delta t^{-1}$ can be resolved. The method is therefore not perfectly suitable for contact-impact and restitution problems, for contact a crude rule of thumb is that the time a node or body is in contact should span at least a few time steps to appropriately resolve the resulting impulse. Having said this, these problems are still difficult to solve reasonably well.

For implicit static contact problems, a common usage of dynamics is to temporarily suppress rigid body modes while a desired contact state is being established. In such a case it is recommended to use numerical damping by for instance choosing $\gamma = 0.6$ and $\beta = 0.38$ to render a smooth response. This technique has proved successful and it is often sufficient to have dynamics initially turned on and then turned off at a time when all rigid body modes have been eliminated by contact. But it is also possible to arbitrarily switch dynamics on and off during a simulation, all this is explained on CONTROL_IMPLICIT_DYNAMICS in the keyword manual.

38.4.4 Linear theory

Linearization of (38.108) with respect to a configuration \mathbf{x} that is in static equilibrium, i.e., $\mathbf{P}[\mathbf{F}_i - \mathbf{F}_e] = \mathbf{0}$, can be written as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}, \quad (38.122)$$

where we use $\mathbf{u} = \Delta\mathbf{x}$ to denote the displacement measured from the reference point \mathbf{x} . For the sake of clarity we also abused some notation, using \mathbf{M} and \mathbf{D} to actually mean \mathbf{PMP}^T and \mathbf{PDP}^T , respectively. Furthermore, \mathbf{K} denotes the sum of the material (38.54) and geometric (38.64) contributions to the static tangent stiffness matrix as well as any contributions from the external force, e.g., contacts,

$$\mathbf{K} = \mathbf{P} \int_V [\mathbf{B}^T \mathbf{E} \mathbf{B} + \mathcal{B}^T \boldsymbol{\sigma} \mathcal{B}] dV \mathbf{P}^T - \mathbf{P} \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}} \mathbf{P}^T. \quad (38.123)$$

The right hand side \mathbf{F} in (38.122) should be interpreted as an external (small) load that only depends on time t and serves to dynamically perturb the static equilibrium. This equation can be discretized in time using the Newmark time integration scheme above and then efficiently integrated with no update of the tangent stiffness matrix between steps. This of course assumes that the displacements \mathbf{u} are small enough to not affect \mathbf{K} to great extent.

Consider undamped free vibration in (38.122), i.e., $\mathbf{D} = \mathbf{0}$ and $\mathbf{F} = \mathbf{0}$, and transform this equation from time to frequency plane assuming constant \mathbf{M} and \mathbf{K} . This results in an eigenvalue problem

$$\mathbf{K}\mathbf{u} = \omega^2 \mathbf{M}\mathbf{u} \quad (38.124)$$

that can be solved in LS-DYNA for the angular frequency ω and mode shape \mathbf{u} . The stress $\boldsymbol{\sigma}$ in (38.123) affects the frequency ω in the following way. If the model is in a tensile state, i.e., the principal stresses are positive, then the eigenvalues of \mathbf{K} increase compared to a stress free state and from (38.124) the frequencies will increase. Conversely, the frequencies will decrease if the principal stresses decrease, this is the effect of tuning a guitar string by increasing or decreasing its tension. If linearizing with respect to a contact state, the last term in (38.123) will in effect constrain relative motion between parts or nodes in contact. More specifically, the relative normal displacement in \mathbf{u} will be zero, and the relative tangential motion will be governed by the present stick/slip condition. If in stick mode the relative tangential displacement in \mathbf{u} will be zero, and if in slip mode it is unconstrained. See CONTROL_IMPLICIT_EIGENVALUE for the available options to solve (38.124).

An example using modal analysis in this respect is shown in Figure 38.5.

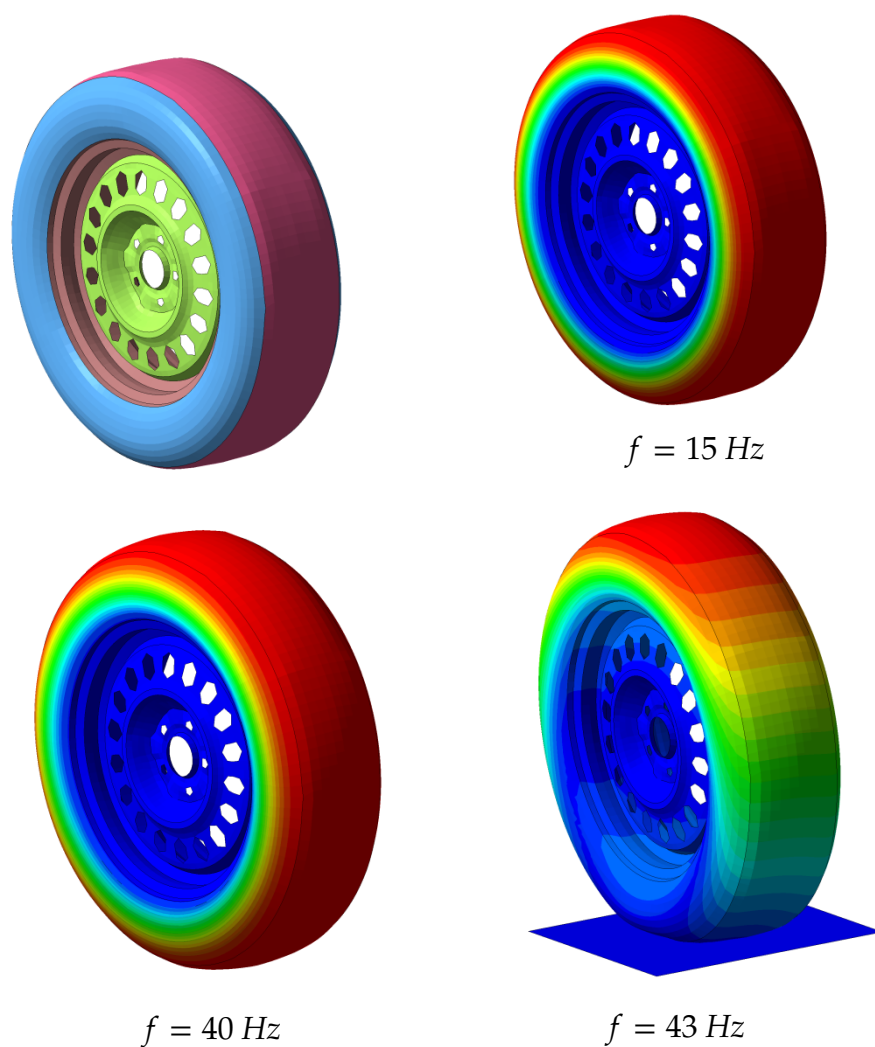


Figure 38.5. Intermittent eigenvalue analysis of tire. Model shown in top left, followed by lowest frequency modes for unpressurized tire (top right), inflated tire (bottom left) and inflated tire and frictional contact (bottom right). Resultant mode displacements are fringed.

In general $D \neq \mathbf{0}$ and D and/or K may be non-symmetric as discussed in Section 38.3.3, but we assume M is symmetric and positive definite and still $F = \mathbf{0}$. The characteristic equation approach for solving (38.122) makes use of the harmonic ansatz

$$\mathbf{u}(t) = \sum_j \exp(\mu_j t) \Phi_j \quad (38.125)$$

which is inserted into (38.122) to yield

$$\sum_j \exp(\mu_j t) \{\mu_j^2 M + \mu_j D + K\} \Phi_j = \mathbf{0} \quad (38.126)$$

to which the quadratic eigenvalue problem is associated⁹

$$\{\mu^2 M + \mu D + K\} \Phi = \mathbf{0}. \quad (38.127)$$

Because of the non-symmetry of the involved matrices, the eigenvalues to (38.127) may be complex but come in conjugate pairs. That is, if $\mu = r + is$ is an eigenvalue then $\bar{\mu} = r - is$ is also an eigenvalue, and if $\Phi = \mathbf{Y} + i\Psi$ is the eigenvector associated with μ then $\bar{\Phi} = \mathbf{Y} - i\Psi$ is the eigenvector for $\bar{\mu}$. However, examining the sum of two such terms in (38.126) yields

$$\begin{aligned} & \exp(\mu t) \{\mu^2 M + \mu D + K\} \Phi + \exp(\bar{\mu} t) \{\bar{\mu}^2 M + \bar{\mu} D + K\} \bar{\Phi} \\ &= 2 \exp(rt) \left\{ \left((r^2 - s^2) M + rD + K \right) (\cos(st) \mathbf{Y} - \sin(st) \Psi) \right. \\ & \quad \left. - s(2rM + D) (\sin(st) \mathbf{Y} + \cos(st) \Psi) \right\}, \end{aligned} \quad (38.128)$$

meaning that the solution in time domain is real as expected. If all eigenvalues are purely complex, i.e., $\mu_j = is_j$ and the real parts $r_j = 0$ vanish, then the solution to (38.122) is harmonic with angular frequencies $\omega_j = |s_j|$. This is apparent from (38.128) and corresponds to the traditional solution to (38.124), with the exception of a non-zero damping matrix D . But if an eigenvalue has a real part $r_j > 0$ then from (38.128) the solution is exponentially growing and thus unstable. Solving (38.127) can therefore be used to detect instabilities in a system, eigenvalues with positive real parts correspond to unstable modes. A common indicator used for this is the *damp ratio* defined as

$$\vartheta = -2 \frac{r}{|s|}, \quad (38.129)$$

so basically a negative damp ratio indicates an unstable mode. An application of this is brake squeal, see Figure 38.6, for which friction instabilities may result in unwanted noise and erratic behavior. Noticable is that if $D = \mathbf{0}$ and μ is an eigenvalue then $-\mu$ is also an eigenvalue, so then complex eigenvalues come in clusters of four, $\{\mu, \bar{\mu}, -\mu, -\bar{\mu}\}$. Then it suffice that eigenvalues have nonzero real parts $r_j \neq 0$ for a system to be unstable, which corresponds to negative eigenvalues in (38.124).

To solve the system (38.127), it is transformed to a first order eigenvalue problem using $\tilde{\Phi} = \mu \Phi$, resulting in

$$\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix} \begin{bmatrix} \Phi \\ \tilde{\Phi} \end{bmatrix} = \mu \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \Phi \\ \tilde{\Phi} \end{bmatrix}, \quad (38.130)$$

which can be solved by well-established eigenvalue algorithms. In the output files, LS-DYNA reports eigenvalues with positive imaginary parts only, i.e., $s_j > 0$, and the real and imaginary parts (when non-zero) of the associated eigenvector Φ .

⁹ The eigenvalue problem (1.127) is readily obtained by Laplace transforming (1.122).

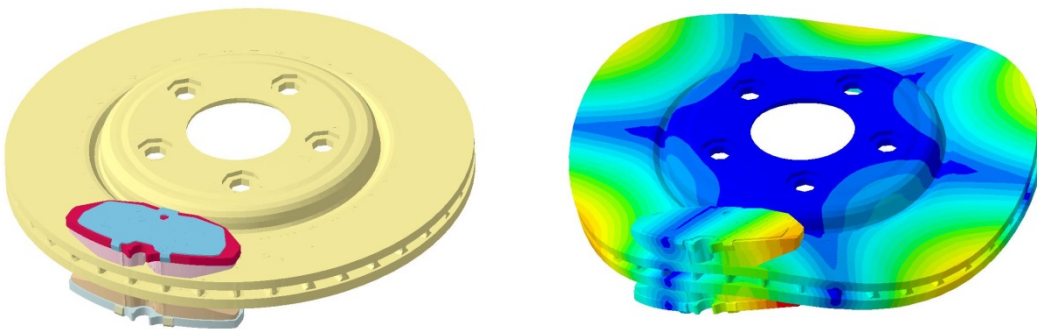
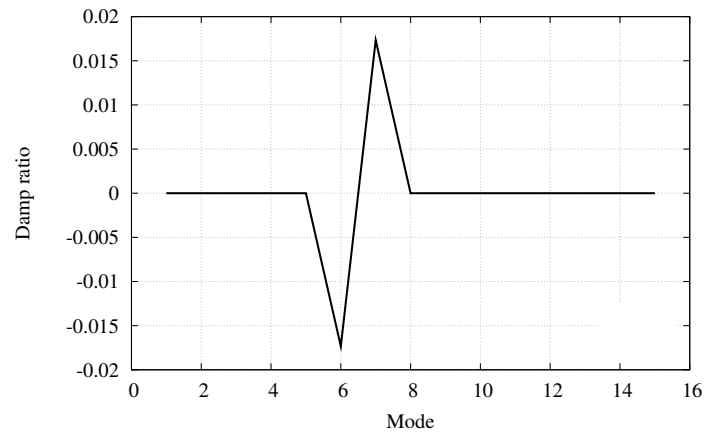


Figure 38.6. Brake squeal application. Pressure pads are applied to a rotating disc and a nonsymmetric eigenvalue solution reveals friction instabilities. The damp ratio ϑ is plotted at the top as function of mode number, see (38.129). Mode #6 is the unstable mode and is depicted bottom right with displacements fringed.

39

Arc-length

Arc-length methods are available in LS-DYNA for NSOLVR specified between 6 and 9, this and all other parameters in this Section are located on the *CONTROL_IMPLICIT_SOLUTION keyword. These solvers use the Riks/Crisfield methods but unfortunately go under the old LSDIR.EQ.1 option which makes them somewhat limited in terms of applicability. For LSDIR.EQ.2 the arc-length method described in Ritto-Corrêa and Camotim [2008] is implemented for the combination of NSOLVR.EQ.12 and ARCMTH.EQ.3. For this method the parameters ARCPSI (ψ), ARCALF (α) and ARCTIM apply, out of which the last simply tells at what time arc-length is initiated and the first two are to be described in more detail below. We begin by an explanatory overview of the arc-length method in general for which we will constantly be referring to [Figure 39.1](#) below. After that the mathematical details are revealed.

39.1 Overview

An implicit static problem is driven by a parameter t referred to as the time, and assuming that a solution is obtained at $t = t_n$ the objective is to determine the solution given the constraint $t = t_{n+1}$, where t_{n+1} is given. We assume that the problem can be associated with representative force and displacement parameters f and d and we distinguish between load- and displacement-driven problems. An example of a load-driven problem is when $f = f(t)$ is an external load and d is the resulting displacement of the associated nodes, and likewise a displacement-driven problem is when $d = d(t)$ is a prescribed displacement and f is the corresponding reaction force. A solution is likely obtained if the problem is stable, i.e., the force-displacement curve is monotonically increasing, but this method is not designed to handle *limit* or *turning* points. A limit point is illustrated in figure (a) and a turning point in figure (b), the solution in the next step may not be the one desired or not even exist, in both cases we want to find the solution that continuously follow the path corresponding to the force-displacement curve.

The solution for this is to set the stepping parameter t free, i.e., replace the constraint $t = t_{n+1}$ with an arc-length constraint $g(d, f) = 0$. In words, this basically means that a multidimensional sphere (arc) is put around the last converged solution and the next solution is to be found on that given sphere, the stepping parameter t is now a solution variable. This makes the problem well-posed but unfortunately there are multiple solutions to the problem, and it may turn out that the wrong solution is found. In figures (c) and (d), the effect of the arc-length constraints is illustrated and there are two possible solutions, one feasible that allows us to continue in the right direction and one infeasible that takes us in the wrong direction. The latter phenomenon is termed doubling back, and is in practice not easily avoided. Two additional parameters are available that have shown to improve the robustness in this respect.

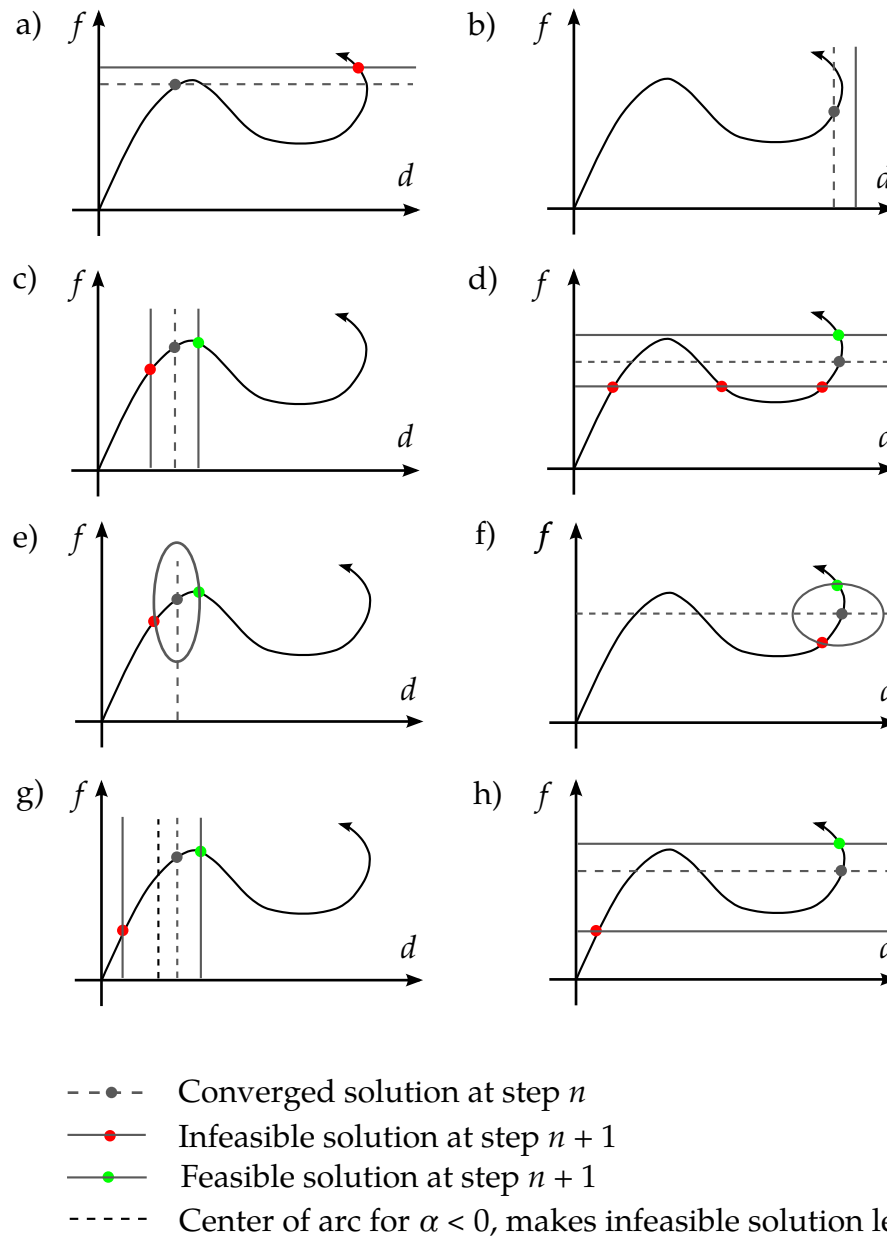


Figure 39.1. Representative force-displacement curves for illustrating arc-length and accompanying parameters

In figure (d), two of the infeasible solutions can in practice be avoided by including the stepping parameter in the arc-length constraint, thus converting a cylinder to a sphere in space-time. This is adjusted by the parameter $0 \leq \psi < 1$, and the constraint reads $(1 - \psi)g(d, f) + \psi(t - t_n)^2 = 0$, the effect of this constraint is illustrated in figures (e) and (f), and should be compared with figures (c) and (d). Note that two infeasible solutions are avoided when comparing figures (d) and (f), it may sometimes be worth using a non-zero value for ψ , e.g., $\psi = 0.1$.

Another problem is that the feasible and infeasible solution may be too close to the last converged solution, making the result from the simulation very unpredictable. For this a parameter α is introduced that translates the center of the spatial sphere in the direction of the linear prediction (i.e., the first Newton iterate of the implicit solution procedure). Assuming that this prediction is in the direction we want, using $\alpha < 0$ will move the center, and consequently the infeasible solution, away from where the iterates are taking place. In addition, the radius of the arc will increase making it less probable to find the incorrect solution. This option has shown effective in solving snap-through problems when using small steps to resolve maximum load values, and is illustrated in figures (g) and (h). For snap-back problems, using $\alpha = 1$ could be an interesting choice since this centers the arc right between the previously converged point and the first predictor in the arc length method, thus encouraging the next solution to be found in the reversed direction. An example of a snap-back problem is shown in [Figure 39-3](#).

39.2 Nonlinear equations

The following, except for a change in notation, is very similar to the nonlinear theory presented in the previous chapter. The generalization is that t will here be treated as an independent variable that is constrained by arc-length instead of given as a constant. The nonlinear variables are denoted

$$x = \begin{bmatrix} x_I \\ x_D \end{bmatrix}, \quad (39.1)$$

that we assume can be divided into a set of independent and dependent variables. Furthermore we have the time parameter t which may serve as the actual time (for dynamic problems) or just a stepping parameter (for quasi-static problems). The division into independent and dependent variables is motivated by the constraint equation that must be fulfilled, i.e.,

$$h(x, t) = h(x_I, x_D, t) = 0. \quad (39.2)$$

From the constraint, the constraint matrix is evaluated as

$$\frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h}{\partial x_I} & \frac{\partial h}{\partial x_D} \end{bmatrix} = [C_{DI} \quad C_{DD}], \quad (39.3)$$

which in turn determines the space of trial functions used to establish the nonlinear finite element equation,

$$\begin{bmatrix} I_{II} & -(C_{DD}^{-1} C_{DI})^T \end{bmatrix} \begin{bmatrix} r_I \\ r_D \end{bmatrix} = 0, \quad (39.4)$$

where

$$r(x, t) = r(x_I, x_D, t) = \begin{bmatrix} r_I \\ r_D \end{bmatrix} \quad (39.5)$$

is the full residual divided into the set of independent and dependent variables. See previous chapter for further details leading up to [\(39.4\)](#).

39.3 Newton iterations

Here we assume that we are in a given configuration given by

$$\begin{aligned} \mathbf{x}^{(n,i)} &= \mathbf{x}^{(n)} + \Delta \mathbf{x}^{(n,i)}, \\ t^{(n,i)} &= t^{(n)} + \Delta t^{(n,i)}. \end{aligned} \quad (39.6)$$

where superscript $n = 0, 1, 2, \dots$ represents converged implicit states and $i = 0, 1, 2, \dots$ represents non-converged Newton iterates. We implicitly assume that $\mathbf{x}^{(0)} = \bar{\mathbf{x}}$, $\Delta \mathbf{x}^{(n,0)} = \mathbf{0}$, $t^{(0)} = 0$ and $\Delta t^{(n,0)} = 0$ are given. In this configuration LS-DYNA computes the full residual given as

$$\mathbf{r}^{(n,i)} = \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_D \end{bmatrix}, \quad (39.7)$$

as well as its dependence on the time parameter

$$\left(\frac{\partial \mathbf{r}}{\partial t} \right)^{(n,i)} = \begin{bmatrix} \frac{\partial \mathbf{r}_I}{\partial t} \\ \frac{\partial \mathbf{r}_D}{\partial t} \end{bmatrix}, \quad (39.8)$$

together with the stiffness matrix given as

$$\left(\frac{\partial \mathbf{r}}{\partial \mathbf{x}} \right)^{(n,i)} = \begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{ID} \\ \mathbf{K}_{DI} & \mathbf{K}_{DD} \end{bmatrix}. \quad (39.9)$$

Likewise the constraint residual, its dependence on the time parameter and constraint matrix are evaluated and given by

$$\begin{aligned} \mathbf{h}^{(n,i)} &= \mathbf{h}, \\ \left(\frac{\partial \mathbf{h}}{\partial t} \right)^{(n,i)} &= \frac{\partial \mathbf{h}}{\partial t}, \\ \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^{(n,i)} &= [\mathbf{C}_{DI} \quad \mathbf{C}_{DD}]. \end{aligned} \quad (39.10)$$

The reduced residual and stiffness matrix are then formed as

$$\begin{aligned} \hat{\mathbf{K}}_{II} &= [\mathbf{I}_{II} \quad -(\mathbf{C}_{DD}^{-1} \mathbf{C}_{DI})^T] \begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{ID} \\ \mathbf{K}_{DI} & \mathbf{K}_{DD} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{II} \\ -\mathbf{C}_{DD}^{-1} \mathbf{C}_{DI} \end{bmatrix}, \\ \hat{\mathbf{r}}_I &= [\mathbf{I}_{II} \quad -(\mathbf{C}_{DD}^{-1} \mathbf{C}_{DI})^T] \left\{ \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_D \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{ID} \\ \mathbf{K}_{DD} \end{bmatrix} \mathbf{C}_{DD}^{-1} \mathbf{h} \right\}, \\ \frac{\partial \hat{\mathbf{r}}_I}{\partial t} &= [\mathbf{I}_{II} \quad -(\mathbf{C}_{DD}^{-1} \mathbf{C}_{DI})^T] \left\{ \begin{bmatrix} \frac{\partial \mathbf{r}_I}{\partial t} \\ \frac{\partial \mathbf{r}_D}{\partial t} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{ID} \\ \mathbf{K}_{DD} \end{bmatrix} \mathbf{C}_{DD}^{-1} \frac{\partial \mathbf{h}}{\partial t} \right\}, \end{aligned} \quad (39.11)$$

and the independent search direction is given by

$$\delta \mathbf{x}_I = s \delta \mathbf{x}_I^s + \delta t \frac{\partial \mathbf{x}_I^s}{\partial t}. \quad (39.12)$$

Here s is the line search parameter and

$$\begin{aligned}\delta \mathbf{x}_I^s &= -\hat{\mathbf{K}}_{II}^{-1} \hat{\mathbf{r}}_I, \\ \frac{\partial \mathbf{x}_I^s}{\partial t} &= -\hat{\mathbf{K}}_{II}^{-1} \frac{\partial \hat{\mathbf{r}}_I}{\partial t}.\end{aligned}\quad (39.13)$$

The full search direction is completed by computing the dependent part as

$$\delta \mathbf{x}_D = s \delta \mathbf{x}_D^s + \delta t \frac{\partial \mathbf{x}_D^s}{\partial t}, \quad (39.14)$$

where

$$\begin{aligned}\delta \mathbf{x}_D^s &= -\mathbf{C}_{DD}^{-1} \mathbf{C}_{DI} \delta \mathbf{x}_I^s - \mathbf{C}_{DD}^{-1} \mathbf{h}, \\ \frac{\partial \mathbf{x}_D^s}{\partial t} &= -\mathbf{C}_{DD}^{-1} \mathbf{C}_{DI} \frac{\partial \mathbf{x}_I^s}{\partial t} - \mathbf{C}_{DD}^{-1} \frac{\partial \mathbf{h}}{\partial t}.\end{aligned}\quad (39.15)$$

Finally the new configuration is updated by means of

$$\begin{aligned}\Delta \mathbf{x}_I^{(n,i+1)} &= \Delta \mathbf{x}_I^{(n,i)} + \delta \mathbf{x}_I, \\ \Delta \mathbf{x}_D^{(n,i+1)} &= \Delta \mathbf{x}_D^{(n,i)} + \delta \mathbf{x}_D, \\ \Delta t^{(n,i+1)} &= \Delta t^{(n,i)} + \delta t.\end{aligned}\quad (39.16)$$

Upon convergence we set

$$\begin{aligned}\Delta \mathbf{x}_I^{(n)} &= \Delta \mathbf{x}_I^{(n,i+1)}, \\ \Delta \mathbf{x}_D^{(n)} &= \Delta \mathbf{x}_D^{(n,i+1)}, \\ \Delta t^{(n)} &= \Delta t^{(n,i+1)},\end{aligned}\quad (39.17)$$

and hence

$$\begin{aligned}\mathbf{x}_I^{(n+1)} &= \mathbf{x}_I^{(n)} + \Delta \mathbf{x}_I^{(n)}, \\ \mathbf{x}_D^{(n+1)} &= \mathbf{x}_D^{(n)} + \Delta \mathbf{x}_D^{(n)}, \\ t^{(n+1)} &= t^{(n)} + \Delta t^{(n)}.\end{aligned}\quad (39.18)$$

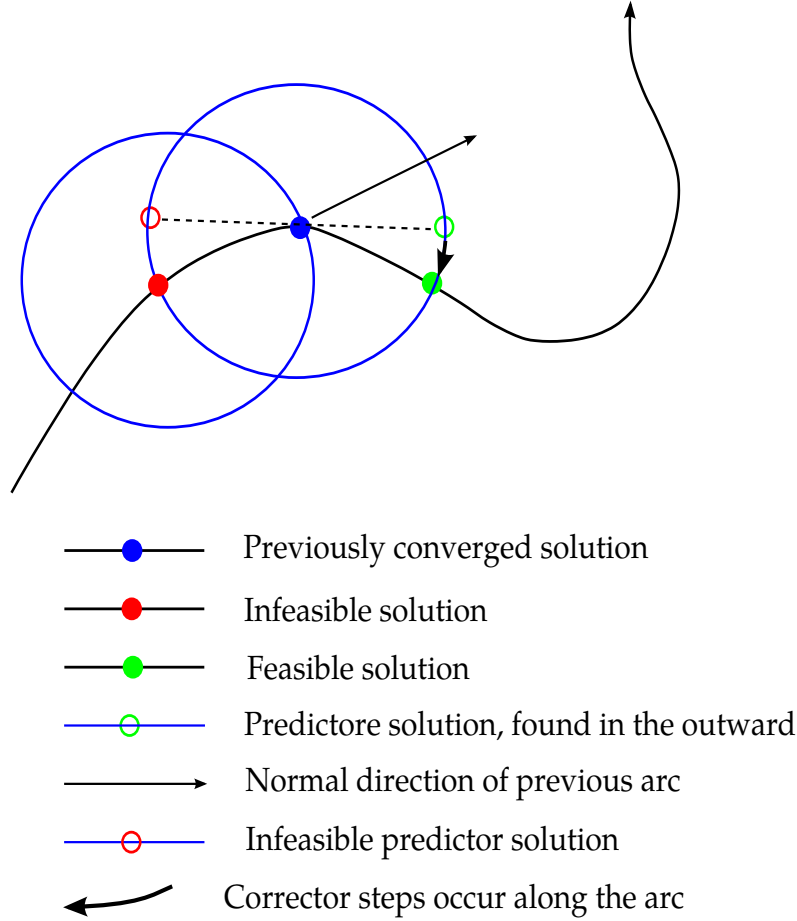


Figure 39.2. Predictor and corrector steps

39.4 Arc-length constraint – predictor step

For the predictor step, $i = 0$, the following constraint is imposed

$$(1 - \psi) \frac{(\Delta \mathbf{x}_I^{(n,1)})^T \Delta \mathbf{x}_I^{(n,1)}}{(\Delta \mathbf{x}_I^{(0,1)})^T \Delta \mathbf{x}_I^{(0,1)}} + \psi \frac{\Delta t^{(n,1)} \Delta t^{(n,1)}}{\Delta t^{(0,1)} \Delta t^{(0,1)}} - 1 = 0, \quad (39.19)$$

where for $n = 0$ we use

$$\begin{aligned} \Delta \mathbf{x}_I^{(0,1)} &= \Delta \bar{t} \frac{\partial \mathbf{x}_I^s}{\partial t} \\ \Delta t^{(0,1)} &= \Delta \bar{t}. \end{aligned} \quad (39.20)$$

Referring to [Figure 39.2](#) this constraint is geometrically interpreted as to find a predictor solution on the arc with the previously converged solution as its center. Writing out the above in terms of known quantities results in the following two possible values of the increment in time step parameter

$$\delta t_{\pm} = \pm \left(\psi_x \left(\frac{\partial \mathbf{x}_I^s}{\partial t} \right)^T \frac{\partial \mathbf{x}_I^s}{\partial t} + \psi_t \right)^{-1/2}, \quad (39.21)$$

corresponding to the two possible predictor solutions on the arc, where

$$\begin{aligned} \psi_x &= \frac{1 - \psi}{(\Delta \mathbf{x}_I^{(0,1)})^T \Delta \mathbf{x}_I^{(0,1)}}, \\ \psi_t &= \frac{\psi}{\Delta t^{(0,1)} \Delta t^{(0,1)}}. \end{aligned} \quad (39.22)$$

The actual value used is determined from the sign of

$$\frac{\psi_x}{(1 - \frac{\alpha}{2})^2} \left(\Delta \mathbf{x}_I^{(n-1)} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n-1,1)} \right)^T \frac{\partial \mathbf{x}_I^s}{\partial t} + \psi_t \Delta t^{(n-1)}, \quad (39.23)$$

if positive $\delta t = \delta t_+$, otherwise $\delta t = \delta t_-$. This condition is to say that the solution continues in the direction of the previously converged state, for the initial step $n = 0$, $\delta t = \delta t_+$. Again referring to [Figure 39.2](#) we simply want to avoid going backwards to the infeasible predictor solution.

39.5 Arc-length constraint – corrector steps

For the corrector steps, $i > 0$, the following constraint is imposed

$$(1 - \psi) \frac{\left(\Delta \mathbf{x}_I^{(n,i+1)} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n,1)} \right)^T \left(\Delta \mathbf{x}_I^{(n,i+1)} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n,1)} \right)}{\left(1 - \frac{\alpha}{2} \right)^2 (\Delta \mathbf{x}_I^{(0,1)})^T \Delta \mathbf{x}_I^{(0,1)}} + \psi \frac{\Delta t^{(n,i+1)} \Delta t^{(n,i+1)}}{\Delta t^{(0,1)} \Delta t^{(0,1)}} - 1 = 0, \quad (39.24)$$

which geometrically says that we should find the next iterate on the arc. Expanding, this amounts to

$$\begin{aligned} \alpha_x \left(\Delta \mathbf{x}_I^{(n,i)} + s \delta \mathbf{x}_I^s + \delta t \frac{\partial \mathbf{x}_I^s}{\partial t} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n,1)} \right)^T \left(\Delta \mathbf{x}_I^{(n,i)} + s \delta \mathbf{x}_I^s + \delta t \frac{\partial \mathbf{x}_I^s}{\partial t} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n,1)} \right) + \\ \alpha_t (\Delta t^{(n,i)} + \delta t) (\Delta t^{(n,i)} + \delta t) - 1 = 0, \end{aligned} \quad (39.25)$$

where

$$\begin{aligned} \alpha_x &= \frac{1 - \psi}{\left(1 - \frac{\alpha}{2} \right)^2 (\Delta \mathbf{x}_I^{(0,1)})^T \Delta \mathbf{x}_I^{(0,1)}} \\ \alpha_t &= \frac{\psi}{\Delta t^{(0,1)} \Delta t^{(0,1)}}. \end{aligned} \quad (39.26)$$

This can be written in terms of a polynomial in s and δt as

$$a_{ss}s^2 + a_{tt}\delta t^2 + 2a_{st}s\delta t + 2a_s s + 2a_t \delta t \quad (39.27)$$

where

$$\begin{aligned} a_{ss} &= \alpha_x (\delta \mathbf{x}_I^s)^T \delta \mathbf{x}_I^s \\ a_{tt} &= \alpha_x \left(\frac{\partial \mathbf{x}_I^s}{\partial t} \right)^T \frac{\partial \mathbf{x}_I^s}{\partial t} + \alpha_t \\ a_{st} &= \alpha_x (\delta \mathbf{x}_I^s)^T \frac{\partial \mathbf{x}_I^s}{\partial t} \\ a_{st} &= \alpha_x (\delta \mathbf{x}_I^s)^T \frac{\partial \mathbf{x}_I^s}{\partial t} \\ a_s &= \alpha_x \left(\Delta \mathbf{x}_I^{(n,i)} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n,1)} \right)^T \delta \mathbf{x}_I^s \\ a_t &= \alpha_x \left(\Delta \mathbf{x}_I^{(n,i)} - \frac{\alpha}{2} \Delta \mathbf{x}_I^{(n,1)} \right)^T \frac{\partial \mathbf{x}_I^s}{\partial t} + \alpha_t \Delta t^{(n,i)}. \end{aligned} \quad (39.28)$$

For a given line search parameter value, the time increment can have two possible values

$$\delta t_{\pm} = \frac{-a_{st}s - a_t \pm \sqrt{(a_{st}^2 - a_{tt}a_{ss})s^2 + 2(a_{st}a_t - a_s a_{tt})s + a_t^2}}{a_{tt}}. \quad (39.29)$$

and the value we use for the update is given by $\delta t = \delta t_+$ if $a_t \geq 0$, otherwise $\delta t = \delta t_-$. This decision is based on the requirement of having $\delta t \rightarrow 0$ when $s \rightarrow 0$.

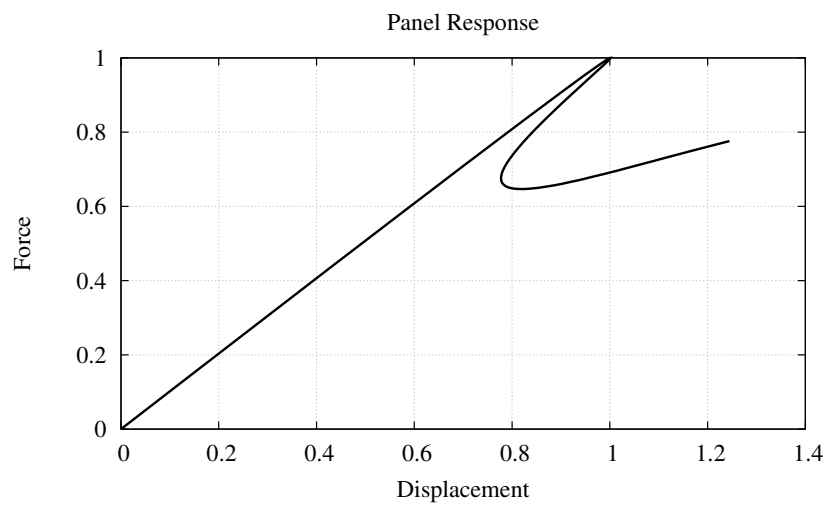
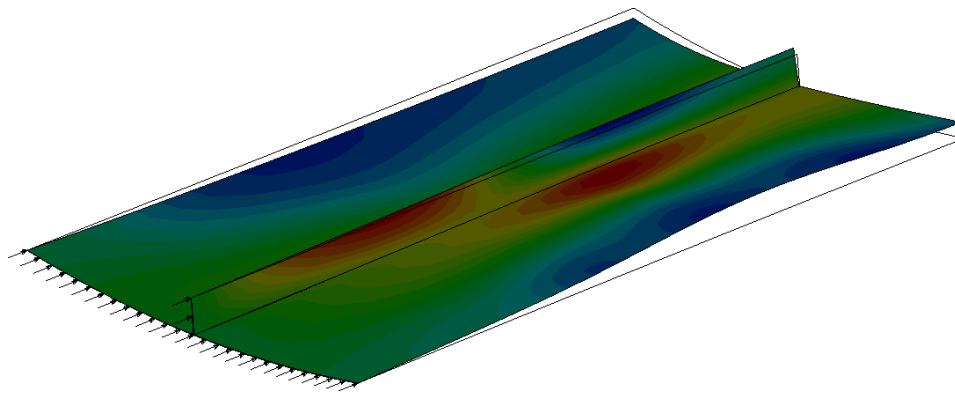


Figure 39-3 Snap-back buckling of panel, normalized force displacement curve shown

40

Sparse Direct Linear Equation Solvers

LS-DYNA has 5 options for direct solution of the sparse systems of linear equations that arise in LS-DYNA. All 5 options are based on the multifrontal algorithm [Duff and Reid, 1983]. Multifrontal is a member of the current generation of sparsity preserving factorization algorithms that also have very fast computational rates. That is multifrontal works with a sparsity preserving ordering to reduce the overall size of the direct factorization and the amount of work it takes to compute that factorization.

40.1 Sparsity Preserving Orderings

In LS-DYNA there are two ordering algorithms for preserving the sparsity of the direct factorization. The algorithms are Multiple Minimum Degree (MMD) and METIS [Karypis and Kumar, 1998]. MMD computes the ordering using locally based decisions and a bottom-up approach. It is inexpensive and very effective for small problems that are problems with fewer than 100,000 rows. METIS computes the ordering from a top down approach. While METIS usually takes more time than MMD to compute the ordering, the METIS ordering reduces the work for the factorization enough to recover the additional ordering cost. METIS is especially effective for large problems, especially those that are modeling three-dimensional solids.

The user can specify either algorithm using keyword `*CONTROL_IMPLICIT_LINEAR`. The default is to use MMD for problems with fewer than 100,000 rows and METIS for problems with more than 100,000 rows. We recommend that the user try both orderings as sometimes MMD is better than METIS on large problems that are not three-dimensional solids.

40.2 Multifrontal Algorithm

The multifrontal algorithm factors a sparse matrix in a way that vastly reduces the amount of work required to compute the factorization compared to methods such as the frontal, profile, skyline, and variable band. These older methods counted on clustering the nonzero entries of the factorization close to the diagonal to keep the size of the factorization and the amount of work required to compute the factorization to a minimum. The factorization was then computed in a serial, left-to-right fashion, essentially following a chain of computations.

The multifrontal algorithm instead follows a tree of computations where the tree structure is established by the sparsity preserving orderings, See [Figure 40.1](#). It is this tree structure that greatly reduces the work required to compute a factorization and the size of the resulting factorization. At the bottom of the tree, a frontal matrix is assembled with the original matrix data and those columns that are fully assembled are eliminated. The remainder of the frontal matrix is updated from the factored columns and passed up the tree to the parent front in what is called an update matrix. As the computation works its way up the tree, a frontal matrix is formed by assembling the original matrix data and the update matrices from its children in the tree. The fully assembled columns are factored and the remaining columns updated and passed up the tree. At the root (top) of the tree, the remaining columns are factored.

By organizing the factorization as a sequence of partial factorization of dense frontal matrices, the multifrontal algorithm can be very fast in performing the required computations. It can use all of the modern technology for dense linear algebra to get high performance computational kernels that should achieve near peak computational performance for a given processor. Only 1% to 5% of the work of the factorization is performed with slower operations such as scatter/gather.

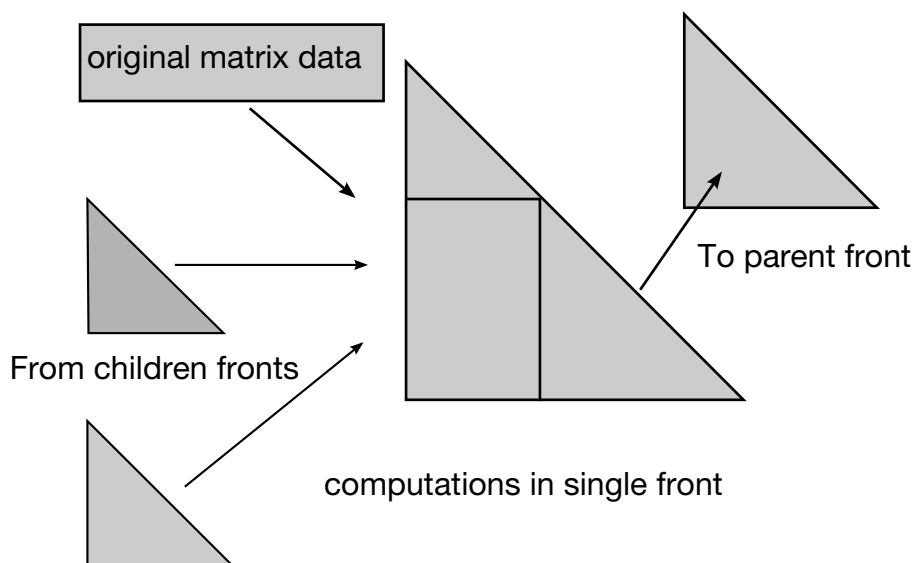
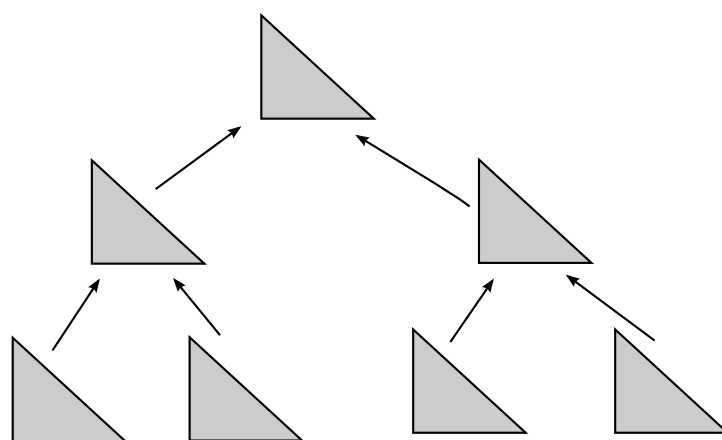


Figure 40.2. Single front algorithm.

40.3 The Five Solver Options

In LS-DYNA three new direct solution options were added. For backward compatibility, the two older options were kept. The five options are:



Multifrontal elimination tree

Figure 40.1. Multifrontal algorithm.

Solver No.	Method
1	Older implementaion of Solver No. 4. Uses Real*4 arithmetic, has out of memory capabilities as well as distributed memory parallelism. Only uses MMD ordering. Was former default method. Retained for backward compatibility. We recommend switching to Solver No. 4 for improved performance.
3	Same as 1 except uses Real*8 arithmetic. We recommend switching to Solver No. 5 or 6 for improved performance.
4	Real*4 implementation of multifrontal which includes automatic out-of-memory capabilities as well as distributed memory parallelism. Can use either MMD or METIS orderings. Default method.
5	Real*8 implementation of Solver No. 4.
6	Multifrontal solver from BCSLIB-EXT [Boeing Company, 1999]. Uses Real*8 arithmetic with extensive capabilities for large problems and some Shared Memory Parallelism. Can use either MMD or METIS orderings. If the other solvers cannot factor the problem in the allocated memory, try using this solver.

We strongly recommend using Solvers 4 through 6. Solvers 1 and 3 are included for backward compatibility with older versions of LS-DYNA but are slower the Solvers 4 through 6. Solvers 4 and 5 are 2 to 6 times faster than the older versions, respectively. Solver 6 on a single processor computer should be comparable to Solver 5 but has more extensive capabilities for solving very large problems with limited memory. Solvers 4 and 5 should be used for distributed memory parallel implementations of LS-DYNA. Solver 6 can be used in shared memory parallel.

In an installation of LS-DYNA where both integer and real numbers are stored in 8 byte quantities, then Solvers 1 and 3 are equivalent and Solvers 4 and 5 are equivalent.

40.4 Treating Matrix Singularities

LS-DYNA has two different techniques for preventing singularities in the stiffness matrix, K . The most common type of matrix singularity arises from the use of certain types of shell elements. These shell elements generate no matrix contribution in the normal direction for each node. Depending on the geometry around the node and what other types of elements are connected to the node, there may or may not be a matrix singularity associated with the rotation around the normal direction at one or more nodes. This is commonly called the drilling rotation singularity.

The first way LS-DYNA has for preventing such matrix singularities is to add a small amount of stiffness in the normal direction at each node of every shell element that has the drilling rotation problem. This “drilling” stiffness matrix is orthogonal to rigid

body motions. The user can control whether this approach is used and how much stiffness is added via the `*CONTROL_IMPLICIT_SOLUTION` keyword card. `DRLMTH` and `DRLPARM` are set in fields 5 and 6. If `DRLMTH = 1` then this approach is used. The amount of stiffness added is controlled via `DRLPARM`. The default for `DRLPARM` is 1.0 for linear problems and 100.0 for nonlinear problems. `DRLPARM * .0001` is added in the normal direction at each node to the diagonal terms associated with the rotational degrees of freedom for certain types of elemental matrices. For eigenvalue problems the amount of stiffness added is 1.E-12.

Adding stiffness to handle the drilling rotation problem has been used extensively. While a robust and reliable approach, its drawback is that the added stiffness may affect the quality of the computed results. The user can also select not to use this approach and depend solely on AUTOSPC, the other method for preventing matrix singularities.

AUTOSPC stands for AUTOMATIC Single Point Constraints. AUTOSPC examines `K` after all of the elemental matrices have been assembled and all of the constraints have been applied for columns that are singular. The user controls AUTOSPC using `ASPCMTH` and `ASPCTOL`, fields 7 and 8 of the `CONTROL_IMPLICIT_SOLUTION` keyword card. If `ASPCMTH = 1`, AUTOSPC is used. For every set of columns of `K` that correspond to the translational or rotational degrees of freedom for a node or rigid body those columns are examined. The singular values of the diagonal block of the columns are computed. If the ratio of the smallest and largest singular values is less than `ASPCTOL` then the set of the columns is declared singular and a constraint is imposed to remove the singularity. The defaults for `ASPCTOL` is 1.E-6 when the matrix is assembled in `REAL*4` precision and 1.E-8 when `REAL*8` is used. The imposed constraint sets the degree of freedom to zero that is associated with the column that has the largest component in the null space of the columns. If all of the singular values are less than `ASPCTOL` all of the degrees of freedom in the block are constrained to zero.

41

Sparse Eigensolver

LS-DYNA now includes the Block Shift and Invert Lanczos eigensolver from BCSLIB-EXT. This eigensolver is used in LS-DYNA to compute the normal modes and mode shapes for the vibration analysis problem

$$\mathbf{K}\Phi = \mathbf{M}\Phi\Lambda, \quad (41.1)$$

where \mathbf{K} and \mathbf{M} are the assembled stiffness and mass matrices, Φ are the eigenvectors (normal mode shapes) and Λ are the eigenvalues (normal modes).

The Lanczos algorithm iteratively computes a better and better approximation to the extreme eigenvalues and the corresponding eigenvectors of the ordinary eigenvalue problem $\mathbf{A}\Phi = \Phi\Lambda$ where \mathbf{A} is a real symmetric matrix using only matrix-vector multiplies. To use Lanczos on the vibration analysis problem it must be changed to

$$(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\Phi = \Phi\Theta, \quad (41.2)$$

where each shifted and inverted eigenvalue $\theta_i = 1/(\lambda_i - \sigma)$. This change to an ordinary eigenvalue problem makes the eigenvalues of the original problem near σ become the extreme eigenvalues of the ordinary eigenvalue problem. This helps the Lanczos algorithm compute those eigenvalues quickly.

BCSLIB-EXT uses a sophisticated logic to choose a sequence of shifts, σ_i , to enable the computation of a large number of eigenvalues and eigenvectors. At each shift the factorization of $\mathbf{K} - \sigma\mathbf{M}$ is computed. The factorization provides the matrix inertia that tells the algorithm how many eigenvalues are to the left of any given σ_i . Given the inertia information, BCSLIB-EXT can tell how many eigenvalues are in a given interval and determine if all of the eigenvalues in that interval have been computed. As a result, BCSLIB-EXT is a very robust eigensolver.

The implementation of BCSLIB-EXT in LS-DYNA includes a shared memory implementation. However only limited parallel speed-up is available for most problems. This is because the eigensolution requires a vast amount of data that for most problems this data has to be stored on I/O files. The wall clock time for the eigensolver is as much

a function of the speed of the I/O subsystem on the computer as the CPU time. Parallelism can only speed up the CPU time and does nothing to speed-up the I/O time.

The user can request how many and which eigenvalues to compute using the keyword `*CONTROL_IMPLICIT_EIGENVALUE`. Via the parameters on this keyword, the user can request any of the following problems:

- Compute the lowest 50 modes (that is nearest to zero)
- Compute the 20 modes nearest to 30 Hz.
- Compute the lowest 20 modes between 10 Hz and 50 Hz.
- Compute all of the modes between 10 Hz and 50 Hz.
- Compute all of the modes below 50 Hz.
- Compute the 30 modes nearest to 30 Hz between 10 Hz and 50 Hz.

41.1 The Eigenvalue Problem for Rotating Systems

Rotating systems, such as the compressor and turbine assembly of a jet engine, have large inertial forces that are functions of the distance from the axis of rotation. These forces are naturally generated in LS-DYNA if the system is modeled as rotating at the proper angular velocity. However, this is often inconvenient for postprocessing because the solution has an oscillatory character imposed it due to the rotation. A commonly used approach to bypass this difficulty is to impose body forces that are equivalent to the inertial forces due to rotation. In LS-DYNA, these forces are imposed through `*LOAD_BODY_GENERALIZED` and related keywords.

For a system with a constant angular velocity $\boldsymbol{\omega} = \{\omega_x, \omega_y, \omega_z\}^T$, the body force added to the applied load is

$$\mathbf{F}_B = -\mathbf{M}\{2\boldsymbol{\omega} \times \dot{\mathbf{u}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{r} + \mathbf{u}))\}. \quad (41.3)$$

In this equation, \mathbf{r} is the initial coordinate at a point and \mathbf{u} is the displacement. Because the body force is a function of both the velocity and displacement, it contributes both damping and stiffness matrices to the eigenvalue problem. Furthermore, since the term involving the initial coordinate creates an initial stress in the structure, the initial stress matrix \mathbf{K}^σ (also called the nonlinear stiffness) is also added to the eigenvalue problem.

The damping and stiffness terms are easily derived in matrix form once the cross product is expressed in matrix form.

$$\boldsymbol{\omega} \times \mathbf{r} = \boldsymbol{\Omega} \mathbf{r} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}. \quad (41.4)$$

The linearized equation for vibration is

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + [\mathbf{K} + \mathbf{K}^\sigma] \mathbf{u} = -\mathbf{M}\{\boldsymbol{\Omega}\dot{\mathbf{u}} + \boldsymbol{\Omega}^2 \mathbf{u}\}. \quad (41.5)$$

Rewriting this equation into the traditional form for eigenvalue analysis produces:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}^R \dot{\mathbf{u}} + \mathbf{K}^R \mathbf{u} &= 0 \\ \mathbf{C}^R &= \mathbf{C} + \mathbf{M}\boldsymbol{\Omega} \\ \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}^R \dot{\mathbf{u}} + \mathbf{K}^R \mathbf{u} &= 0. \end{aligned} \quad (41.6)$$

The inertial contribution to the damping matrix is not symmetric, nor does it fulfill the requirements for Rayleigh damping, and therefore the resulting eigenvectors and eigenvalues are complex. The inertial term to the stiffness matrix is, however, symmetric and it softens the structure, thereby reducing its natural frequencies.

If the damping term is omitted, the matrices are real and symmetric, and the resulting eigenvalue problem may be solved with the standard eigenvalue methods. The natural frequencies won't be correct, but they are typically close enough to the complex solution that they can be used for initial design calculations.

42

Boundary Element Method

LS-DYNA can be used to solve for the steady state or transient fluid flow about a body using a boundary element method. The method is based on the work of Maskew [1987], with the extension to unsteady flow with arbitrary body motion following the work of Katz and Maskew [1988]. The theory which underlies the method is restricted to inviscid, incompressible, attached fluid flow. The method should not be used to analyze flows where shocks or cavitation are present.

In practice the method can be successfully applied to a wider class of fluid flow problems than the assumption of inviscid, incompressible, attached flow would imply. Many flows of practical engineering significance have large Reynolds numbers (above 1 million). For these flows the effects of fluid viscosity are small if the flow remains attached, and the assumption of zero viscosity may not be a significant limitation. Flow separation does not necessarily invalidate the analysis. If well-defined separation lines exist on the body, then wakes can be attached to these separation lines and reasonable results can be obtained. The Prandtl-Glauert rule can be used to correct for non-zero Mach numbers in air, so the effects of aerodynamic compressibility can be correctly modeled (as long as no shocks are present).

42.1 Governing Equations

The partial differential equation governing inviscid, incompressible fluid flow is given by Laplace's equation

$$\nabla^2 \Phi = 0, \quad (42.1)$$

where Φ is the velocity potential (a scalar function). The fluid velocity anywhere in the flow field is equal to the gradient of Φ . The boundary condition on this partial differential equation is provided by the condition that there must be no flow in the direction normal to the surface of the body. Note that time does not appear in [Equation \(42.1\)](#). This is

because the assumption of incompressibility implies an infinite sound speed; any disturbance is felt everywhere in the fluid instantaneously. Although this is not true for real fluids, it is a valid approximation for a wide class of low-speed flow problems.

[Equation \(42.1\)](#) is solved by discretizing the surface of the body with a set of quadrilateral or triangular surface segments (boundary elements). Each segment has an associated source and doublet strength. The source strengths are computed from the free-stream velocity, and the doublet strengths are determined from the boundary condition. By requiring that the normal component of the fluid velocity be zero at the center of each surface segment, a linear system of equations is formed with the number of equations equal to the number of unknown doublet strengths. When this system is solved, the doublet strengths are known. The source and doublet distributions on the surface of the body then completely determine the flow everywhere in the fluid.

The linear system for the unknown doublet strengths is shown in [Equation \(1.2\)](#).

$$[\text{mic}]\{\mu\} = \{\text{rhs}\}. \quad (42.2)$$

In this equation μ are the doublet strengths, $[\text{mic}]$ is the matrix of influence coefficients which relate the doublet strength of a given segment to the normal velocity at another segment's mid-point, and rhs is a right-hand-side vector computed from the known source strengths. Note that mic is a fully-populated matrix. Thus, the cost to compute and store the matrix increases with the square of the number of segments used to discretize the surface of the body, while the cost to factor this matrix increases with the cube of the number of segments. Users should keep these relations in mind when defining the surface segments. A surface of 1000 segments can be easily handled on most any computer, but a 10,000 segment representation would not be feasible on any but the most powerful supercomputers.

42.2 Surface Representation

The surface of the body is discretized by a set of triangular or quadrilateral surface segments. The best fluid-structure interaction results will be obtained if the boundary element segments coincide with, and use identical nodes as, the structural segments used to define the body. An input format has been implemented to make this easy if thin shell elements are used to define the structure (see the User's Manual). Using the same nodes to define the boundary elements and the structure guarantees that the boundary elements follow the structure as it deforms, and provides a means for the fluid pressure to load the structure.

The nodes used to define the corners of the boundary element segments must be ordered to provide a normal vector which points into the fluid ([Figure 42.1](#)).

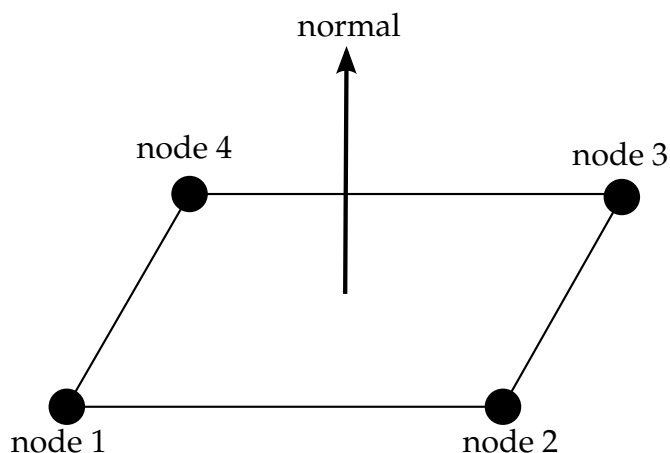


Figure 42.1. Counter-clockwise ordering of nodes when viewed from fluid looking towards solid provides unit normal vector pointing into the fluid.

Triangular segments are specified by using the same node for the 3rd and 4th corner of the segment (the same convention used for shell elements in LS-DYNA). Very large segments can be used with no loss of accuracy in regions of the flow where the velocity gradients are small. The size of the elements should be reduced in areas where large velocity gradients are present. Finite-precision arithmetic on the computer will cause problems if the segment aspect ratios are extremely large (greater than 1000). The most accurate results will be obtained if the segments are rectangular, and triangular segments should be avoided except for cases where they are absolutely required.

42.3 The Neighbor Array The fluid velocities (and, therefore, the fluid pressures) are determined by the gradient of the velocity potential. On the surface of the body, this can be most easily computed by taking derivatives of the doublet distribution on the surface. These derivatives are computed using the doublet strengths on the boundary element segments. The “Neighbor Array” is used to specify how the gradient is computed for each boundary element segment. Thus, accurate results will not be obtained unless the neighbor array is correctly specified by the user.

Each boundary element segment has 4 sides (see [Figure 42.2](#)). Side 1 connects the 1st and 2nd nodes, side 2 connects the 2nd and 3rd nodes, etc. The 4th side is null for triangular segments.

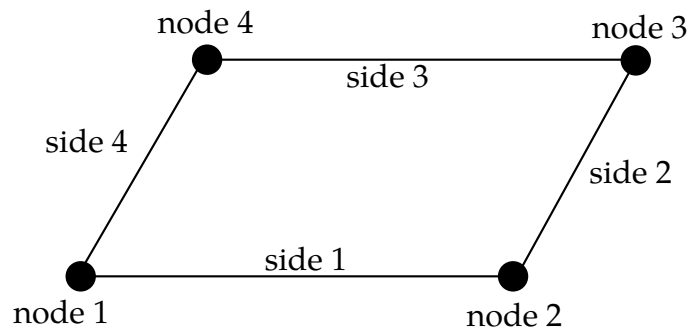


Figure 42.2. Each segment has 4 sides.

For most segments the specification of neighbors is straightforward. For the typical case a rectangular segment is surrounded by 4 other segments, and the neighbor array is as shown in [Figure 42.3](#). A biquadratic curve fit is computed, and the gradient is computed as the analytical derivative of this biquadratic curve fit evaluated at the center of segment j .

There are several situations which call for a different specification of the neighbor array. For example, boundary element wakes result in discontinuous doublet distributions, and the biquadratic curve fit should not be computed across a wake. [Figure 42.4](#) illustrates a situation where a wake is attached to side 2 of segment j . For this situation two options exist. If neighbor $(2, j)$ is set to zero, then a linear computation of the gradient in the side 2 to side 4 direction will be made using the difference between the doublet strengths on segment j and segment neighbor $(4, j)$. By specifying neighbor $(2, j)$ as a negative number the biquadratic curve fit will be retained. The curve fit will use segment j , segment neighbor $(4, j)$, and segment $-\text{neighbor}(2, j)$; which is located on the opposite side of segment neighbor $(4, j)$ as segment j . The derivative in the side 2 to side 4 direction is then analytically evaluated at the center of segment j using the quadratic curve fit of

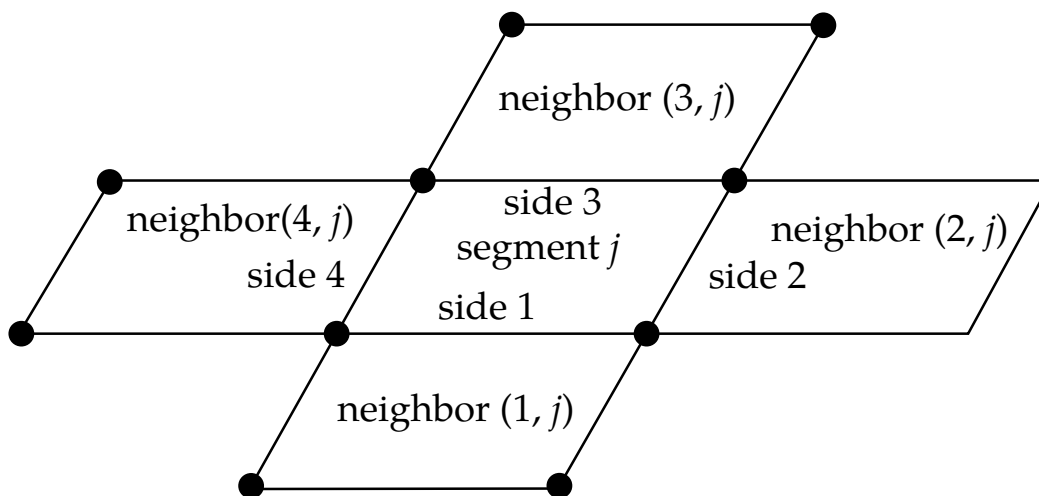


Figure 42.3. Typical neighbor specification.

the doublet strengths on the three segments shown.

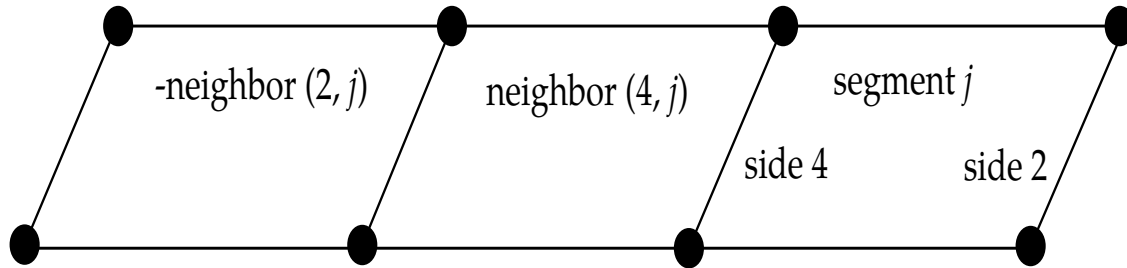


Figure 42.4. If neighbor $(2, j)$ is a negative number it is assumed to lie on the opposite side of neighbor $(4, j)$ as segment j .

A final possibility is that no neighbors at all are available in the side 2 to side 4 direction. In this case both neighbor $(2, j)$ and neighbor $(4, j)$ can be set to zero, and the gradient in that direction will be assumed to be zero. This option should be used with caution, as the resulting fluid pressures will not be accurate for three-dimensional flows. However, this option is occasionally useful where quasi-two dimensional results are desired. All of the above options apply to the side 1 to side 3 direction in the obvious ways.

For triangular boundary element segments side 4 is null. Gradients in the side 2 to side 4 direction can be computed as described above by setting neighbor $(4, j)$ to zero (for a linear derivative computation) or to a negative number (to use the segment on the other side of neighbor $(2, j)$ and a quadratic curve fit). There may also be another triangular segment which can be used as neighbor $(4, j)$ (see [Figure 42.5](#)).

42.4 Wakes

Wakes should be attached to the boundary element segments at the trailing edge of a lifting surface (such as a wing, propeller blade, rudder, or diving plane). Wakes should also be attached to known separation lines (such as the sharp leading edge of a

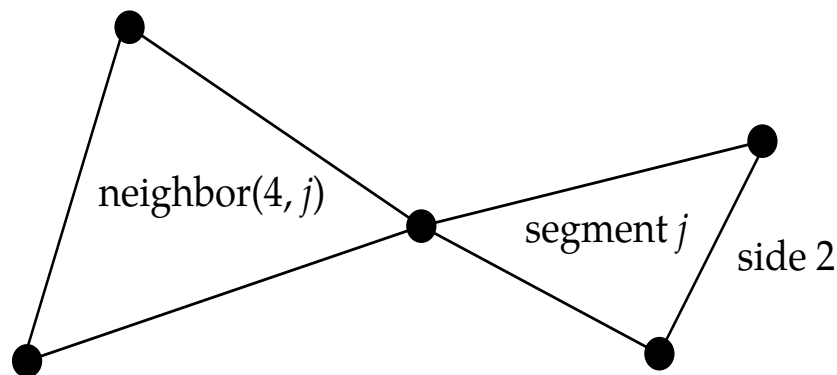


Figure 42.5. Sometimes another triangular boundary element segment can be used as neighbor $(4, j)$.

delata wing at high angles of attack). Wakes are required for the correct computation of

surface pressures for these situations. As described above, two segments on opposite sides of a wake should never be used as neighbors. Correct specification of the wakes is required for accurate results.

Wakes convect with the free-stream velocity. The number of segments in the wake is controlled by the user, and should be set to provide a total wake length equal to 5-10 times the characteristic streamwise dimension of the surface to which the wake is attached. For example, if the wake is attached to the trailing edge of a wing whose chord is 1, then the total length of the wake should at least 5, and there is little point in making it longer than 10. Note that each wake segment has a streamwise length equal to the magnitude of the free stream velocity times the time increment between calls to the Boundary Element Method routine. This time increment is the maximum of the LS-DYNA time step and DTBEM specified on Card 1 of the BEM input. The influence coefficients for the wake segments must be recomputed for each call to the Boundary Element Method, but these influence coefficients do not enter into the matrix of influence coefficients which must be factored.

42.5 Execution Time Control

The Boundary Element Method will dominate the total execution time of a LS-DYNA calculation unless the parameters provided on Card 1 of the BEM input are used to reduce the number of calls to the BEM. This can usually be done with no loss in accuracy since the characteristic time of the structural dynamics and the fluid flow are so different. For example, the characteristic time in LS-DYNA is given by the characteristic length of the smallest structural element divided by the speed of sound of the material. For typical problems this characteristic time might be on the order of microseconds. Since the fluid is assumed to be incompressible (infinite speed of sound), the characteristic time of the fluid flow is given by the streamwise length of the smallest surface (e.g. a rudder) divided by the fluid velocity. For typical problems this characteristic time might be on the order of milliseconds. Thus, for this example, the boundary element method could be called only once for every 1000 LS-DYNA iterations, saving an enormous amount of computer time.

The parameter DTBEM on Card 1 of the BEM input is used to control the time increment between calls to the boundary element method. The fluid pressures computed during the last call to the BEM will continue to be used for subsequent LS-DYNA iterations until DTBEM has elapsed.

A further reduction in execution time may be obtained for some applications using the input parameter IUPBEM. This parameter controls the number of calls to the BEM routine between computation (and factorization) of the matrix of influence coefficients (these are time-consuming procedures). If the motion of the body is entirely rigid body

motion there is no need to recompute and factor the matrix of influence coefficients, and the execution time of the BEM can be significantly reduced by setting IUPBEM to a very large number. For situations where the motion of the body is largely rigid body motion with some structural deformation an intermediate value (e.g. 10) for IUPBEM can be used. It is the user's responsibility to verify the accuracy of calculations obtained with IUPBEM greater than 1.

The final parameter for controlling the execution time of the boundary element method is FARBEM. The routine which calculates the influence coefficients switches between an expensive near-field and an inexpensive far-field calculation depending on the distance from the boundary element segment to the point of interest. FARBEM is a non-dimensional parameter which determines where the far-field boundary lies. Values of FARBEM of 5 and greater will provide the most accurate results, while values as low as 2 will provide slightly reduced accuracy with a 50% reduction in the time required to compute the matrix of influence coefficients.

42.6 Free-Stream Flow

The free-stream flow is specified in the second card of input. The free-stream velocity is assumed to be uniform. The free-stream static pressure is assumed to be uniform, and can be used to load the structure for hydrostatic pressure. If the structure has an internal pressure, the free-stream static pressure should be set to the difference between the external and internal static pressures. The Mach number can be used to correct for the effect of compressibility in air (as long as no shocks are present). Following the Prandtl-Glauert correction, the pressures due to fluid flow are increased as follows

$$dp_{\text{corrected}} = \frac{dp_{\text{uncorrected}}}{\sqrt{1 - M^2}} \quad (42.3)$$

where M is the free-stream Mach number. Note that this correction is only valid for flows in a gas (it is not valid for flows in water).

43

SPH

Smoothed Particle Hydrodynamics (SPH) is an N-body integration scheme developed by Lucy, Gingold and Monaghan [1977]. The method was developed to avoid the limitations of mesh tangling encountered in extreme deformation problems with the finite element method. The main difference between classical methods and SPH is the absence of a grid. Therefore, the particles are the computational framework on which the governing equations are resolved. This new model requires a new calculation method, which is briefly explained in the following.

43.1 SPH Formulation

43.1.1 Definitions

The particle approximation of a function is:

$$\Pi^h f(x) = \int f(y) W(x - y, h) dy, \quad (43.1)$$

where W is the kernel function.

The Kernel function W is defined using the function θ by the relation:

$$W(\mathbf{x}, h) = \frac{1}{h(\mathbf{x})^d} \theta(\mathbf{x}). \quad (43.2)$$

where d is the number of space dimensions and h is the so-called smoothing length which varies in time and in space.

$W(\mathbf{x}, h)$ should be a centrally peaked function. The most common smoothing kernel used by the SPH community is the cubic B-spline which is defined by choosing θ as:

$$\theta(u) = C \times \begin{cases} 1 - \frac{3}{2}u^2 + \frac{3}{4}u^3 & \text{for } |u| \leq 1 \\ \frac{1}{4}(2-u)^3 & \text{for } 1 \leq |u| \leq 2 \\ 0 & \text{for } 2 < |u| \end{cases} \quad (43.3)$$

where C is a constant of normalization that depends on the number of space dimensions.

The SPH method is based on a quadrature formula for moving particles ($(\mathbf{x}_i(t))$ $i \in \{1..N\}$, where $\mathbf{x}_i(t)$ is the location of particle i , which moves along the velocity field \mathbf{v} .

The particle approximation of a function can now be defined by:

$$\Pi^h f(\mathbf{x}_i) = \sum_{j=1}^N w_j f(\mathbf{x}_j) W(\mathbf{x}_i - \mathbf{x}_j, h), \quad (43.4)$$

where $w_j = \frac{m_j}{\rho_j}$ is the “weight” of the particle. The weight of a particle varies proportionally to the divergence of the flow.

The SPH formalism implies a derivative operator. A particle approximation for the derivative operator must be defined. Before giving the definition of this approximation, we define the gradient of a function as:

$$\nabla f(x) = \nabla f(x) - f(x) \nabla 1(x), \quad (43.5)$$

where 1 is the unit function.

Starting from this relation, we can define the particle approximation to the gradient of a function:

$$\Pi^h \nabla f(\mathbf{x}_i) = \sum_{j=1}^N \frac{m_j}{\rho_j} [f(\mathbf{x}_j) A_{ij} - f(\mathbf{x}_i) A_{ij}], \quad (43.6)$$

where $A_{ij} = \frac{1}{h^{d+1}} \theta'(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{h})$.

We can also define the particle approximation of the partial derivative $\frac{\partial}{\partial x^\alpha}$:

$$\Pi^h (\frac{\partial f}{\partial x^\alpha})(\mathbf{x}_i) = \sum_{j=1}^N w_j f(\mathbf{x}_j) A^\alpha(\mathbf{x}_i, \mathbf{x}_j), \quad (43.7)$$

where \mathbf{A} is the operator defined by: $\mathbf{A}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{h^{d+1}(\mathbf{x}_i, \mathbf{x}_j)} \frac{(\mathbf{x}_i - \mathbf{x}_j)}{\|\mathbf{x}_i - \mathbf{x}_j\|} \theta'(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{h(\mathbf{x}_i, \mathbf{x}_j)})$, A^α is the component α of the \mathbf{A} vector.

43.1.2 Discrete Form of Conservation Equations

We are looking for the solution of the equation:

$$L_v(\phi) + \text{div} \mathbf{F}(\mathbf{x}, t, \phi) = S, \quad (43.8)$$

where $\phi \in R^d$ is the unknown, \mathbf{F}^β with $\beta \in \{1..d\}$ represents the conservation law and L_v is the transport operator defined by:

$$L_v: \phi \rightarrow L_v(\phi) = \frac{\partial \phi}{\partial t} + \sum_{l=1}^d \frac{\partial(\mathbf{v}^l \phi)}{\partial x^l}. \quad (43.9)$$

The strong formulation approximation:

In the search of the strong solution, the equation is kept at its initial formulation. The discrete form of this problem implies the definition of the operator of derivation D defined by:

$$D: \phi \rightarrow D\phi(x) = \nabla \phi(x) - \phi(x) \nabla 1(x). \quad (43.10)$$

The particle approximation of this operator is:

$$D_h \phi(\mathbf{x}_i) = \sum_{j=1}^N w_j (\phi(\mathbf{x}_j) - \phi(\mathbf{x}_i)) A_{ij}, \quad (43.11)$$

where A_{ij} is defined previously.

Finally, the discrete form of the strong formulation is written:

$$\frac{d}{dt} (w_i \phi(\mathbf{x}_i)) + w_i D_h F(\mathbf{x}_i) = w_i S(\mathbf{x}_i), \quad (43.12)$$

But this form is not conservative; therefore the strong formulation is not numerically acceptable. Thus, we are compelled to use the weak form.

The weak formulation approximation:

In the weak formulation, the adjoint of the L_v operator is used:

$$L_v^*: \phi \rightarrow L_v^*(\phi) = \frac{\partial \phi}{\partial t} + \sum_{l=1}^d v^l \frac{\partial \phi}{\partial x^l}. \quad (43.13)$$

The discrete form of this operator corresponds to the discrete formulation of the adjoint of $D_{h,s}$:

$$D_{h,s}^* \phi(\mathbf{x}_i) = \sum_{j=1}^N w_j (\phi(\mathbf{x}_i) A_{ij} - \phi(\mathbf{x}_j) A_{ji}). \quad (43.14)$$

A discrete adjoint operator for the partial derivative is also necessary, and is taken to be the α - th component of the operator:

$$D_{\alpha}^* \phi(\mathbf{x}_i) = \sum_{j=1}^N w_j \phi(\mathbf{x}_j) A^{\alpha}(\mathbf{x}_i, \mathbf{x}_j) - w_j \phi(\mathbf{x}_i) A^{\alpha}(\mathbf{x}_j, \mathbf{x}_i) \quad (43.15)$$

These definitions are leading to a conservative method. Hence, all the conservative equations encountered in the SPH method will be solved using the weak form.

43.1.3 Applications to Conservation Equations

With the definitions explained above, the conservation equations can now be written in their discrete form.

Momentum conservation equation:

$$\frac{d\mathbf{v}^{\alpha}}{dt}(\mathbf{x}_i(t)) = \frac{1}{\rho_i} \frac{\partial(\sigma^{\alpha\beta})}{\partial x_i}(\mathbf{x}_i(t)), \quad (43.16)$$

where α, β are the space indices.

The particle approximation of the weak form of this equation is:

$$\frac{d\mathbf{v}^{\alpha}}{dt}(\mathbf{x}_i) = \sum_{j=1}^N m_j \left(\frac{\sigma^{\alpha,\beta}(\mathbf{x}_i)}{\rho_i^2} A_{ij} - \frac{\sigma^{\alpha,\beta}(\mathbf{x}_j)}{\rho_j^2} A_{ji} \right). \quad (43.17)$$

Energy conservation equation:

$$\frac{dE}{dt} = -\frac{P}{\rho} \nabla \cdot \mathbf{v}. \quad (43.18)$$

The particle approximation of the weak form of this equation is:

$$\frac{dE}{dt}(\mathbf{x}_i) = -\frac{P_i}{\rho_i^2} \sum_{j=1}^N m_j \left(v(\mathbf{x}_j) - v(\mathbf{x}_i) \right) A_{ij}. \quad (43.19)$$

43.1.4 Formulation Available in LS-DYNA

It is easy from the general formulation displayed in [Equation \(43.14\)](#) to extend the SPH formalism to a set of equations of discretization for the momentum equation.

For example, if we choose the smoothing function to be symmetric, this can lead to the following equation:

$$\frac{d\mathbf{v}^{\alpha}}{dt}(\mathbf{x}_i) = \sum_{j=1}^N m_j \left(\frac{\sigma^{\alpha,\beta}(\mathbf{x}_i)}{\rho_i^2} + \frac{\sigma^{\alpha,\beta}(\mathbf{x}_j)}{\rho_j^2} \right) A_{ij}. \quad (43.20)$$

This is what we call the “symmetric formulation”, which is chosen in the *CONTROL_-SPH card (IFORM = 2).

Another possible choice is to define the momentum equation by:

$$\frac{d\mathbf{v}^\alpha}{dt}(\mathbf{x}_i) = \sum_{j=1}^N m_j \left(\frac{\sigma^{\alpha,\beta}(\mathbf{x}_i)}{\rho_i \rho_j} A_{ij} - \frac{\sigma^{\alpha,\beta}(\mathbf{x}_j)}{\rho_i \rho_j} A_{ji} \right). \quad (43.21)$$

This is the “fluid formulation” invoked with IFORM = 5 which gives better results than other SPH formulations when fluid material are present, or when material with very different stiffness are used.

43.2 Sorting

In the SPH method, the location of neighboring particles is important. The sorting consists of finding which particles interact with which others at a given time. A bucket sort is used that consists of partitioning the domain into boxes where the sort is performed. With this partitioning the closest neighbors will reside in the same box or in the closest boxes. This method reduces the number of distance calculations and therefore the CPU time.

43.3 Artificial Viscosity

The artificial viscosity is introduced when a shock is present. Shocks introduce discontinuities in functions. The role of the artificial viscosity is to smooth the shock over several particles. To take into account the artificial viscosity, an artificial viscous pressure term Π_{ij} [Monaghan & Gingold 1983] is added such that:

$$p_i \rightarrow p_i + \Pi_{ij}, \quad (43.22)$$

where $\Pi_{ij} = \frac{1}{\bar{\rho}_{ij}} (-\alpha \mu_{ij} \bar{c}_{ij} + \beta \mu_{ij}^2)$.

The notation $\bar{X}_{ij} = \frac{1}{2}(X_i + X_j)$ has been used for median between X_i and X_j , c is the adiabatic sound speed, and

$$\mu_{ij} = \begin{cases} \bar{h}_{ij} \frac{v_{ij} r_{ij}}{r_{ij}^2 + \eta^2} & \text{if } v_{ij} r_{ij} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (43.23)$$

Here, $v_{ij} = (v_i - v_j)$, and $\eta^2 = 0.01 \bar{h}_{ij}^2$ which prevents the denominator from vanishing.

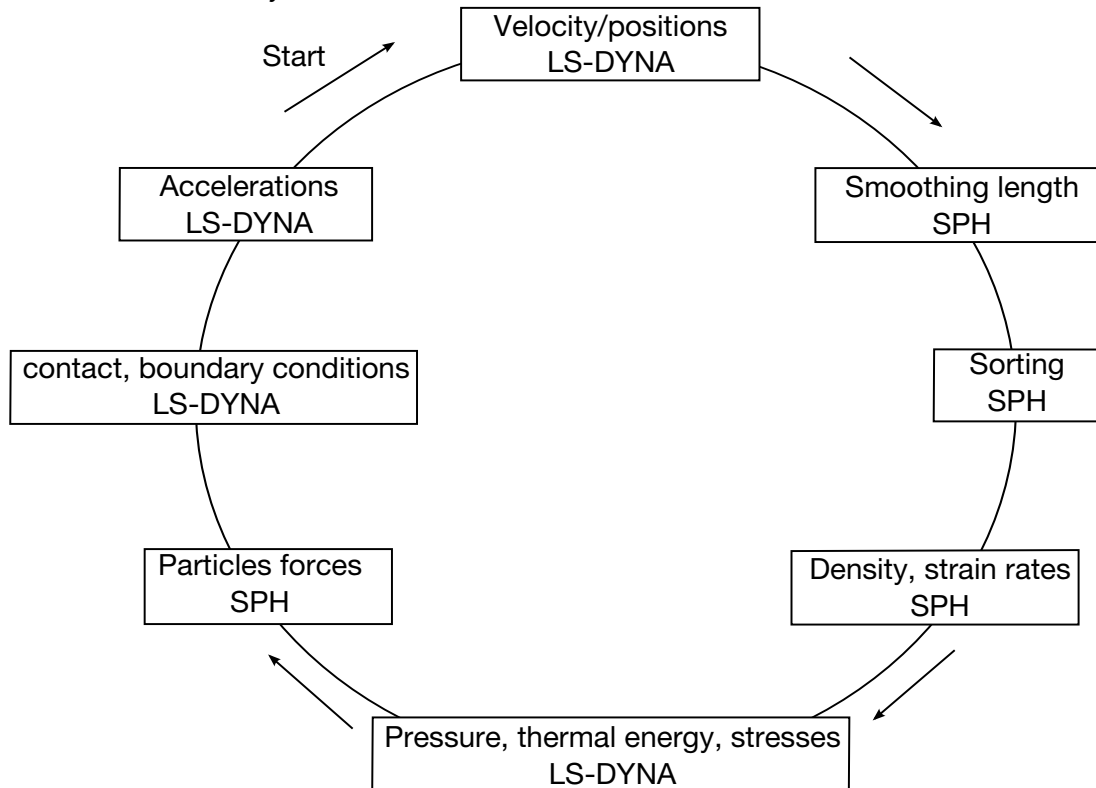
43.4 Time Integration

We use a simple and classical first-order scheme for integration. The time step is determined by the expression:

$$\delta t = C_{\text{CFL}} \text{Min}_i \left(\frac{h_i}{c_i + v_i} \right), \quad (43.24)$$

where the factor C_{CFL} is a numerical constant.

The calculation cycle is:



43.5 Initial Setup

Initially, we have a set of particles with two kinds of properties: physical and geometrical properties.

Physical Properties:

The mass, density, constitutive laws are defined in the ELEMENT_SPH and the PART cards.

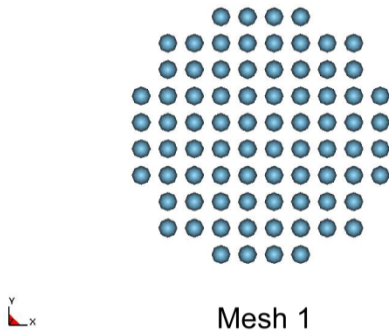
Geometrical Properties:

The geometrical properties of the model concern the way particles are initially placed. Two different parameters are to be fixed: Δx_i lengths and the CSLH coefficient. These parameters are defined in the SECTION_SPH card.

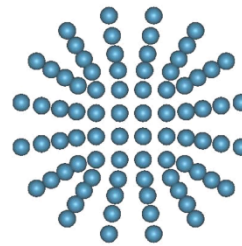
A proper SPH mesh must satisfy the following conditions: it must be as regular as possible and must not contain too large variations.

For instance, if we consider a cylinder SPH mesh, we have at least two possibilities:

TAYLOR TEST (LAW3)
Time = 0



TAYLOR TEST (LAW3)
Time = 0



The mesh number 2 includes too many inter-particle distance discrepancies. Therefore, the first mesh, more uniform, is better.

Finite element coupling

Coupling finite elements and SPH elements is realized by using contact algorithms. Users can choose any “nodes_to_surface” contact type where the slave part is defined with SPH elements and the master part is defined with finite elements.

44

Element-Free Galerkin

Mesh-free methods, which construct the approximation entirely in terms of nodes, permit reduced restriction in the discretization of the problem domain and are less susceptible to distortion difficulties than finite elements. For a variety of engineering problems with extremely large deformation, moving boundaries or discontinuities, mesh-free methods are very attractive. The two most commonly used approximation theories in mesh-free methods are the moving least-squares (MLS) approximation in the Element-free Galerkin (EFG) method [Belytschko et al. 1994], and the reproducing kernel (RK) approximation in the reproducing kernel particle method (RKPM) [Liu et al. 1995]. Since these two methods lead to an identical approximation when monomial basis functions are used, the MLS approximation is used as a basis to formulate the mesh-free discrete equations in this section.

44.1 Moving least-squares

The Element-free Galerkin method uses the moving least-squares approximation to construct the numerical discretization. The discrete MLS approximation of a function $u(\mathbf{x})$, denoted by $u^h(\mathbf{x})$, is constructed by a combination of the monomials as

$$u^h(\mathbf{x}) = \sum_{i=1}^n H_i(\mathbf{x}) b_i(\mathbf{x}) \equiv \mathbf{H}^T(\mathbf{x}) \mathbf{b}(\mathbf{x}), \quad (44.1)$$

where n is the order of completeness in this approximation, the monomial $H_i(\mathbf{x})$ are basis functions, and $b_i(\mathbf{x})$ are the coefficients of the approximation.

The coefficients $b_i(\mathbf{x})$ at any point \mathbf{x} are depending on the sampling points \mathbf{x}_I that are collected by a weighting function $w_a(\mathbf{x} - \mathbf{x}_I)$. This weighting function is defined to have a compact support measured by ' a ', i.e., the sub-domain over which it is nonzero is small relative to the rest of the domain. Each sub-domain $\Delta\Omega_I$ is associated with a node I . The most commonly used sub-domains are disks or balls. A typical numerical model is shown in [Figure 44.1](#).

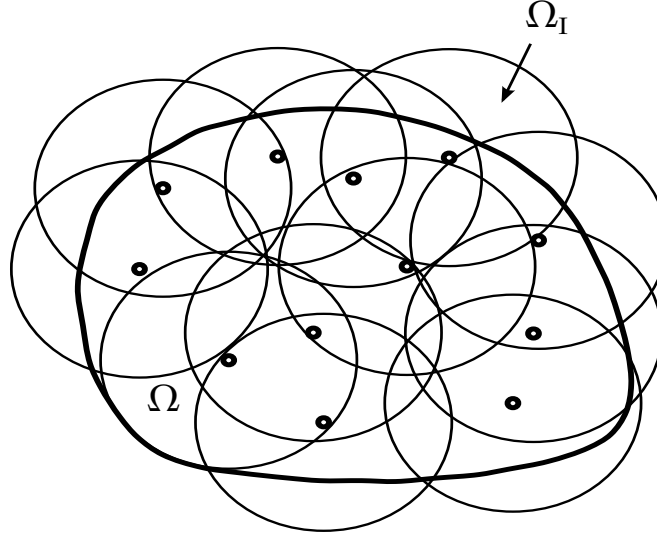


Figure 44.1. Graphical representation of mesh-free discretization

In this development, we employ the cubic B-spline kernel function as the weighting function:

$$w_a(\mathbf{x} - \mathbf{x}_I) = \begin{cases} \frac{2}{3} - 4 \left(\frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \right)^2 + 4 \left(\frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \right)^3 & \text{for } 0 \leq \frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \leq \frac{1}{2} \\ \frac{4}{3} - 4 \left(\frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \right) + 4 \left(\frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \right)^2 - \frac{4}{3} \left(\frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \right)^3 & \text{for } \frac{1}{2} < \frac{\|\mathbf{x} - \mathbf{x}_I\|}{a} \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (44.2)$$

The moving least-squares technique consists in minimizing the weighted L_2 -Norm

$$J = \sum_{I=1}^{NP} W_a(\mathbf{x}) (\mathbf{x} - \mathbf{x}_I) \left[\sum_{i=1}^n H_i(\mathbf{x}) b_i(\mathbf{x}) - u(\mathbf{x}_I) \right]^2, \quad (44.3)$$

where NP is the number of nodes within the support of \mathbf{x} for which the weighting function $w_a(\mathbf{x} - \mathbf{x}_I) \neq 0$.

Equation (39.3) can be written in the form

$$J = (\mathbf{H}\mathbf{b} - \mathbf{u})^T \mathbf{W}_a(\mathbf{x}) (\mathbf{H}\mathbf{b} - \mathbf{u}), \quad (44.4)$$

where

$$\mathbf{u}^T = (u_1, u_2, \dots, u_{NP}), \quad (44.5)$$

$$\mathbf{H} = \begin{bmatrix} \{\mathbf{H}(\mathbf{x}_1)\}^T \\ \dots \\ \{\mathbf{H}(\mathbf{x}_{NP})\}^T \end{bmatrix}, \quad (44.6)$$

$$\{\mathbf{H}(\mathbf{x}_i)\}^T = \{H_1(\mathbf{x}_i), \dots, H_n(\mathbf{x}_i)\}, \quad (44.7)$$

$$\mathbf{W}_a = \text{diag}[w_a(\mathbf{x} - \mathbf{x}_1), \dots, w_a(\mathbf{x} - \mathbf{x}_{NP})]. \quad (44.8)$$

To find the coefficients \mathbf{b} we obtain the extremum of J by

$$\frac{\partial J}{\partial \mathbf{b}} = \mathbf{M}^{[n]}(\mathbf{x})\mathbf{b}(\mathbf{x}) - \mathbf{B}(\mathbf{x})\mathbf{u} = 0, \quad (44.9)$$

where $\mathbf{M}^{[n]}(\mathbf{x})$ is called the moment matrix of $w_a(\mathbf{x} - \mathbf{x}_I)$ and is given by

$$\mathbf{M}^{[n]}(\mathbf{x}) = \mathbf{H}^T \mathbf{W}_a(\mathbf{x}) \mathbf{H}, \quad (44.10)$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{H}^T \mathbf{W}_a(\mathbf{x}). \quad (44.11)$$

So we have

$$\mathbf{b}(\mathbf{x}) = \mathbf{M}^{[n]-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}) \mathbf{u}. \quad (44.12)$$

For $\mathbf{M}^{[n]}(\mathbf{x})$ to be invertible, the support of $w_a(\mathbf{x} - \mathbf{x})$ needs to be greater than a minimum size that is related to the order of basis functions. Using the solution of [Equations \(44.1\), \(44.10\), \(44.11\) and \(44.12\)](#), the EFG approximation is obtained by

$$u^h(\mathbf{x}) = \sum_{I=1}^{NP} \Psi_I(\mathbf{x}) u_I, \quad (44.13)$$

where the EFG shape functions $\Psi_I(\mathbf{x})$ are given by

$$\Psi_I(\mathbf{x}) = \mathbf{H}^T(\mathbf{x}) \mathbf{M}^{[n]-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}), \quad (44.14)$$

and $\Psi_I(\mathbf{x})$ are n th-order complete, i.e.

$$\sum_{I=1}^{NP} \Psi_I(\mathbf{x}) x_{1I}^p x_{2I}^q = x_1^p x_2^q \text{ for } p + q = 0, \dots, n. \quad (44.15)$$

44.2 Integration constraint and strain smoothing

The convergence of the Galerkin method for a partial differential equation is determined by approximation for the unknowns and the numerical integration of the weak form. EFG shape functions with linear consistency can be obtained from MLS approximation with linear basis functions. The employment of linearly consistent mesh-free shape functions in the Galerkin approximation, however, does not guarantee a linear

exactness in the solution of the Galerkin method. It has been shown by Chen et al. [2001] that two integration constraints are required for the linear exactness solution in the Galerkin approximation.

$$\sum_{L=1}^{NIT} \nabla \Psi_I(\hat{x}_L) A_L = 0 \quad \text{for } \{I: \text{supp}(\Psi_I) \cap \Gamma = 0\}, \quad (44.16)$$

$$\sum_{L=1}^{NIT} \nabla \Psi_I(\hat{x}_L) A_L = \sum_{L=1}^{NITh} \mathbf{n} \Psi_I(\tilde{x}_L) s_L \quad \text{for } \{I: \text{supp}(\Psi_I) \cap \Gamma^h \neq 0\}. \quad (44.17)$$

where Γ^h is the natural boundary, Γ is the total boundary, \mathbf{n} is the surface normal on Γ^h , \hat{x}_L and A_L are the spatial co-ordinate and weight of the domain integration point, respectively, \tilde{x}_L and s_L are the spatial co-ordinate and weight of the domain of natural boundary integration point, respectively, NIT is the number of integration points for domain integration and NITh is the number of integration points for natural boundary integration.

A strain smoothing method proposed by Chen and Wu [1998] as a regularization for material instabilities in strain localization was extended in their nodal integration method [Chen et al, 2001] to meet the integration constraints. Here, we adopt the similar concept for the domain integration. It starts with a strain smoothing at the representative domain of a Gauss point by

$$\tilde{\nabla} u_i^h(\mathbf{x}_L) = \frac{1}{A_L} \int_{\Omega_L} \nabla u_i^h(\mathbf{x}_L) d\Omega, A_L = \int_{\Omega_L} d\Omega, \quad (44.18)$$

where Ω_L is a representative domain at each Gauss point and $\tilde{\nabla}$ is the smoothed gradient operator. By applying divergence theorem to Equation (44.18) to yield

$$\tilde{\nabla} u_i^h(\mathbf{x}_L) = \frac{1}{A_L} \int_{\Gamma_L} n u_i^h(\mathbf{x}_L) d\Gamma, \quad (44.19)$$

where Γ_L is the boundary of the representative domain of Gauss point L . Introducing EFG shape functions into Equation (26.24) yields

$$\tilde{\nabla} u_i^h(\mathbf{x}_L) = \sum_I \frac{1}{A_L} \int_{\Gamma_L} \Psi_I(\mathbf{x}) \mathbf{n} d\Gamma \cdot d_{iI} \equiv \sum_I \tilde{\nabla} \Psi_I(\mathbf{x}_L) \cdot d_{iI}. \quad (44.20)$$

It can be shown that the smoothed EFG shape function gradient $\tilde{\nabla} \Psi_I(\mathbf{x}_L)$ meets the integration constraints in Equations (44.16) and (44.17) regardless of the numerical integration employed.

44.3 Lagrangian strain smoothing for path-dependent problems

To avoid the tensile instability caused by the Eulerian kernel functions, the Lagrangian kernel functions are implemented in the current LS-DYNA.

To introduce the Lagrangian EFG shape function into the approximation of a path-dependent problem, the strain increment $\Delta u_{i,j}$ is computed by

$$\Delta u_{i,j} = \frac{\partial \Delta u_i}{\partial x_j} = \frac{\partial \Delta u_i}{\partial X_k} F_{kj}^{-1} = \Delta F_{ik} F_{kj}^{-1}. \quad (44.21)$$

The strain smoothing of $\Delta u_{i,j}$ at a material point x_L is computed by

$$\Delta \tilde{u}_{i,j}(x_L) = \Delta \tilde{F}_{ik}(x_L) \tilde{F}_{kj}^{-1}(x_L), \quad (44.22)$$

where $\tilde{F}_{ij}(x_L)$ is the Lagrangian strain smoothing of deformation gradient and is given by

$$\tilde{F}_{ij}(x_L) = \frac{1}{A_L} \int_{\Gamma_L} u_i^h N_j d\Gamma + \delta_{ij}. \quad (44.23)$$

44.4 Galerkin approximation for explicit dynamic computation

The strong form of the initial/boundary value problem for elasto-dynamics is as follows:

$$\rho \ddot{\mathbf{u}} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_b \text{ in } \Omega, \quad (44.24)$$

with the divergence operator ∇ , the body force \mathbf{f}_b , mass density ρ , and with the boundary conditions:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_0 \text{ on } \Gamma_u \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{h} \text{ on } \Gamma_h, \end{aligned} \quad (44.25)$$

and initial conditions

$$\begin{aligned} \mathbf{u}(\mathbf{X}, 0) &= \mathbf{u}^0(\mathbf{X}) \\ \dot{\mathbf{u}}(\mathbf{X}, 0) &= \dot{\mathbf{u}}^0(\mathbf{X}). \end{aligned} \quad (44.26)$$

To introduce the Lagrangian strain smoothing formulation into the Galerkin approximation, an assumed strain method is employed. The corresponding weak form becomes:

$$\int_{\Omega_x} \rho \delta \mathbf{u} \cdot \ddot{\mathbf{u}} d\Omega + \int_{\Omega_x} \delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} d\Omega = \int_{\Omega_x} \delta \mathbf{u} \cdot \mathbf{f}_b d\Omega + \int_{\Gamma_h} \delta \mathbf{u} \cdot \mathbf{h} d\Gamma. \quad (44.27)$$

Following the derivation for explicit time integration, the equations to be solved have the form

$$\delta \mathbf{u}^T \mathbf{M} \ddot{\mathbf{u}} = \delta \mathbf{u}^T \mathbf{R}, \quad (44.28)$$

where

$$\begin{aligned} \ddot{\mathbf{u}}_I &= [\ddot{d}_{1I}, \ddot{d}_{2I}, \ddot{d}_{3I}]^T \\ M_{IJ} &= \int_{\Omega_x} \rho \Psi_I(\mathbf{x}) \Psi_J(\mathbf{x}) d\Omega = \int_{\Omega_x} \rho^0 \Psi_I(\mathbf{X}) \Psi_J(\mathbf{X}) d\Omega \\ R_I &= \int_{\Omega_x} \tilde{\mathbf{B}}_I^T(\mathbf{x}) \cdot \boldsymbol{\sigma}(\tilde{\mathbf{F}}) d\Omega - [\Psi_I(\mathbf{x}) \mathbf{h}]|_{\Gamma_h} - \int_{\Omega_x} \Psi_I(\mathbf{x}) \mathbf{f}_b d\Omega, \end{aligned} \quad (44.29)$$

where $\tilde{\mathbf{B}}_I^T(\mathbf{x})$ is the smoothed gradient matrix obtained from Equation (44.22), d_{iI} is the coefficient of the approximation or the “generalized” displacement.

44.5 Imposition of essential boundary condition

In general, mesh-free shape functions Ψ_I do not possess Kronecker delta properties of the standard FEM shape functions, i.e.

$$\Psi_I(x_J) \neq \delta_{IJ}. \quad (44.30)$$

This is because, in general, the mesh-free shape functions are not interpolation functions. As a result, a special treatment is required to enforce essential boundary conditions. There are many techniques for mesh-free methods to impose the essential boundary condition. Here, we adopt the transformation method as originally proposed for the RKPM method by Chen et al. [1996].

Therefore, to impose the essential boundary conditions using kinematically admissible mesh-free shape functions by the transformation method, Equation (44.28) can be written as

$$\delta \hat{\mathbf{u}}^T \hat{\mathbf{M}} \ddot{\mathbf{u}} = \delta \hat{\mathbf{u}}^T \hat{\mathbf{F}}^{\text{int}}, \quad (44.31)$$

where

$$\hat{\mathbf{u}} = \mathbf{A} \mathbf{u}; A_{IJ} = \Psi_J(X_I). \quad (44.32)$$

or

$$\mathbf{u} = \mathbf{A}^{-1} \hat{\mathbf{u}}, \quad (44.33)$$

and

$$\hat{\mathbf{M}} = \mathbf{A}^{-T} \mathbf{M} \mathbf{A}^{-1}; \hat{\mathbf{F}}^{\text{int}} = \mathbf{A}^{-T} \mathbf{F}^{\text{int}}. \quad (44.34)$$

A mixed transformation method [Chen et al. 2000] is also considered as an alternative to impose the essential boundary conditions. The mixed transformation method

is an improved transformation method that the coordinate transformation is only applied for the degrees of freedom associated with the essential and contact boundaries.

The nodes are partitioned into three groups: a boundary group G^{B_1} which contains all the nodes subjected to kinematic constraints; group G^{B_2} which contains all the nodes whose kernel supports cover nodes in group G^{B_1} ; and internal group G^I which contains the rest of nodes. Nodes numbers are re-arranged in the following order in the generalized displacement vector:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}^{B_1} \\ \mathbf{u}^{B_2} \\ \mathbf{u}^I \end{bmatrix} \quad (44.35)$$

where \mathbf{u}^{B_1} , \mathbf{u}^{B_2} and \mathbf{u}^I are the generalized displacement vectors associated with groups G^{B_1} , G^{B_2} and G^I respectively. The transformation in Equation (44.32) is also re-arranged as

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}^B \\ \hat{\mathbf{u}}^I \end{bmatrix} \begin{bmatrix} \Lambda^{BB} & \Lambda^{BI} \\ \Lambda^{IB} & \Lambda^{II} \end{bmatrix} \begin{bmatrix} \mathbf{u}^B \\ \mathbf{u}^I \end{bmatrix} \equiv \hat{\Lambda} \mathbf{u}, \quad (44.36)$$

where

$$\begin{aligned} \hat{\mathbf{u}}^B &= \begin{bmatrix} \hat{\mathbf{u}}^{B_1} \\ \hat{\mathbf{u}}^{B_2} \end{bmatrix}; \mathbf{u}^B = \begin{bmatrix} \mathbf{u}^{B_1} \\ \mathbf{u}^{B_2} \end{bmatrix}; \Lambda^{BB} = \begin{bmatrix} \Lambda^{B_1 B_1} & \Lambda^{B_1 B_2} \\ \Lambda^{B_2 B_1} & \Lambda^{B_2 B_2} \end{bmatrix}; \Lambda^{BI} = \begin{bmatrix} 0 \\ \Lambda^{B_2 I} \end{bmatrix}; \Lambda^{IB} \\ &= \begin{bmatrix} \Lambda^{IB_1} & \Lambda^{IB_2} \end{bmatrix}. \end{aligned} \quad (44.37)$$

Here, we introduce a mixed displacement vector \mathbf{u}^* ,

$$\mathbf{u}^* = \begin{bmatrix} \hat{\mathbf{u}}^B \\ \mathbf{u}^I \end{bmatrix} \begin{bmatrix} \Lambda^{BB} & \Lambda^{BI} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}^B \\ \mathbf{u}^I \end{bmatrix} \equiv \Lambda^* \mathbf{u}, \quad (44.38)$$

and Λ^* and its inverse are:

$$\Lambda^* = \begin{bmatrix} \Lambda^{BB} & \Lambda^{BI} \\ 0 & \mathbf{I} \end{bmatrix}; \Lambda^{*-1} = \begin{bmatrix} \Lambda^{BB^{-1}} & -\Lambda^{BB^{-1}} \Lambda^{BI} \\ 0 & \mathbf{I} \end{bmatrix}. \quad (44.39)$$

Only the inversion of Λ^{BB} is required in Equation (23.83.15).

Using the mixed coordinates in Equation (44.38), the transformed discrete Equation (44.31) becomes

$$\delta \mathbf{u}^{*T} \mathbf{M}^* \ddot{\mathbf{u}}^* = \delta \mathbf{u}^{*T} \mathbf{R}^*, \quad (44.40)$$

where

$$\mathbf{M}^* = \mathbf{A}^{*-T} \mathbf{M} \mathbf{A}^{*-1}; \mathbf{R}^* = \mathbf{A}^{*-T} \mathbf{R}. \quad (44.41)$$

The computation in Equations (44.41) is much less intensive than that in Equation (44.31), especially when the number of boundary and contact nodes is much smaller than the number of interior nodes.

44.6 Mesh-free Shell

The extension of explicit mesh-free solid analysis to shell analysis is described in this section. Two projection methods are developed to generate the shell mid-surface using the moving-least-squares approximations. A co-rotational, updated Lagrangian procedure is adopted to handle arbitrarily large rotations with moderate strain responses of the shell structures. A local boundary integration method in conjunction with the selective reduced integration method is introduced to enforce the linear exactness and relieve shear locking.

44.6.1 Mesh-free Shell Surface Representation

Surface reconstruction from disorganized nodes is very challenging in three dimensions. The problem is ill posed, i.e., there is no unique solution. Lancaster et al. [1981] first proposed a fast surface reconstruction using moving least squares method. Their approach was then applied to the computational mechanics under the name ‘mesh-free method’. Implicitly, the mesh-free method uses a combination of smooth basis functions (primitives) to find a scalar function such that all data nodes are close to an iso-contour of that scalar function in a global sense. In reality, the shell surface construction using the 3D mesh-free method is inadequate. This is because the topology of the real surface can be very complicated in three dimensions. Without the information on the ordering or connectivity of nodes, the reconstructed surface will not be able to represent shell intersections, exterior boundaries and shape corners.

In our development of mesh-free shells, we assume that a shell surface is described by a finite element mesh. This can be easily accomplished by converting a part of shell finite elements into mesh-free zone. With the connectivity of nodes provided by the finite element mesh, a shell surface can be reconstructed with mesh-free interpolation from the nodal positions

$$\bar{\mathbf{x}} = \tilde{\Psi}_I(\mathbf{X})\mathbf{x}_I, \quad (44.42)$$

where \mathbf{x}_I is the position vector of the finite element node on the shell surface and $\tilde{\Psi}_I(\mathbf{X})$ is the mesh-free shape function. In the above surface representation, a 3D arbitrary shell surface needs to be projected to a 2D plane. Two approaches for the projection of mesh-free shell surface are used:

- Global parametric representation: The whole shell surface is projected to a parametric plane and the global parametric coordinates are obtained with a parameterization algorithm from the patch of finite elements.
- Local projection representation: A local area of the shell is projected to a plane based on the existing element where the evaluated point is located.

Global parametric approach

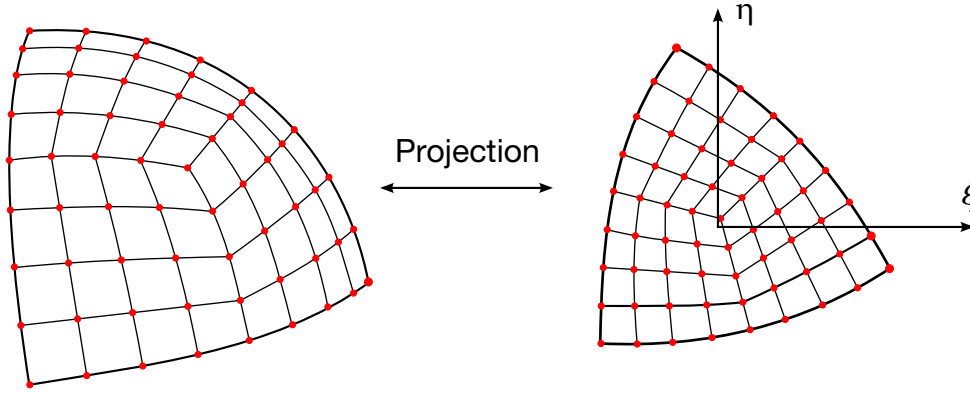


Figure 44.2. Mesh-free shell global approach

In the global approach, a mesh-free zone with a patch of finite elements is mapped onto a parametric plane with an angle-based triangular flattening algorithm [Sheffer and de Sturler 2001], (see [Figure 44.2](#)). The idea of this algorithm is to compute a projection that minimizes the distortion of the FE mesh angles. The mesh-free shape functions are defined in this parametric domain and given by

$$\tilde{\Psi}_I(\mathbf{X}) = \tilde{\Psi}_I(\zeta, \eta), \quad (44.43)$$

where (ζ, η) is the parametric coordinates corresponding to a point \mathbf{X} .

Local projection approach

Different from the parameterization algorithm that constructs the surface globally, we reconstruct the surface locally by projecting the surrounding nodes onto one element. In the local projection method, nodes in elements neighboring the element where the evaluated point is located (for example, the element i in [Figure 44.3](#)) are projected onto the plane which the element defines (the “M-plane” in [Figure 44.3](#)). In this figure, $(\hat{x}, \hat{y}, \hat{z})_i$ is a local system defined for each projected plane and $(\bar{x}, \bar{y}, \bar{z})_I$ is a nodal coordinate system defined for each node where \bar{z} is the initial averaged normal direction.

The mesh-free shape functions are then defined with those locally projected coordinates of the nodes

$$\Psi_I(\mathbf{X}) = \Psi_I(\hat{x}, \hat{y}). \quad (44.44)$$

However, the shape functions obtained directly above are non-conforming, i.e.

$$\Psi_I(\mathbf{X}_J)|_{\text{M-plane}} \neq \Psi_I(\mathbf{X}_J)|_{\text{N-plane}}. \quad (44.45)$$

When the shell structure degenerates to a plate, the constant stress condition cannot be recovered. To remedy this problem, an area-weighted smoothing across different projected planes is used to obtain the conforming shape functions that are given by

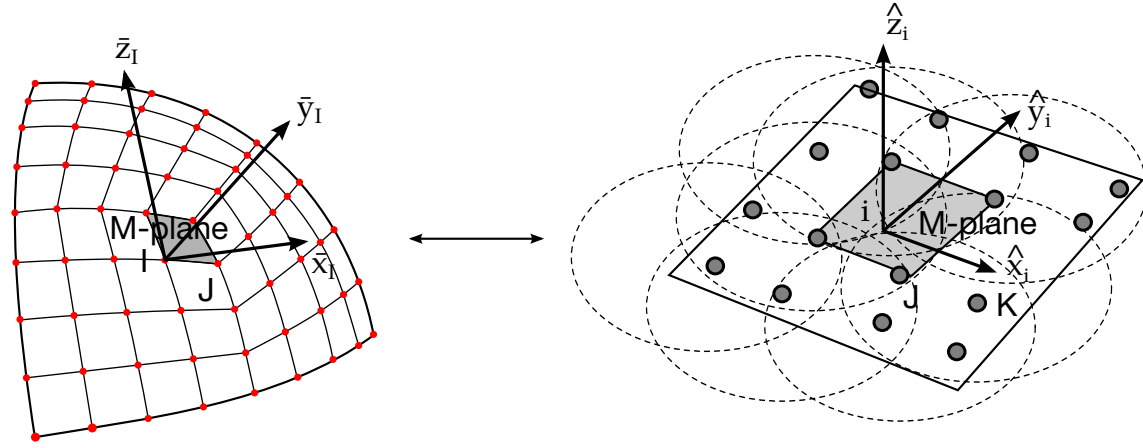


Figure 44.3. Mesh-free shell local projection

$$\tilde{\Psi}_I(\mathbf{X}) = \tilde{\Psi}_I(\hat{x}, \hat{y}) = \frac{\sum_{i=1}^{NIE} \Psi_I(\hat{x}_i, \hat{y}_i) A_i}{\sum_{i=1}^{NIE} A_i}. \quad (44.46)$$

where NIE is the number of surrounding projected planes that can be evaluated at point \mathbf{X} , A_i is the area of the element i , and (\hat{x}_i, \hat{y}_i) is the local coordinates of point \mathbf{X} in the projected plane i .

With this smoothing technique, we can prove that the modified shape functions satisfy at least the partition of unity property in the general shell problems. This property is important for the shell formulation to preserve the rigid-body translation.

When the shell degenerates to a plate, we can also prove that the shape functions obtained from this smoothing technique will meet the n -th order completeness condition as

$$\sum_{I=1}^{NP} \tilde{\Psi}_I(\mathbf{X}) X_{1I}^i X_{2I}^j X_{3I}^k = X_1^i X_2^j X_3^k, \quad i + j + k = n. \quad (44.47)$$

This is a necessary condition for the plate to pass the constant bending patch test.

44.6.2 Updated Lagrangian Formulation and Co-rotational Procedure

The mesh-free shell formulation is based on the Mindlin-Reissner plate theory, thus the geometry and kinematical fields of the shell can be described with the reference surface and fiber direction. The modified Mindlin-Reissner assumption requires that the motion and displacement of the shell are linear in the fiber direction. Assume that the reference surface is the mid-surface of the shell, the global coordinates and displacements at an arbitrary point within the shell body are given by

$$\mathbf{x} = \bar{\mathbf{x}} + \zeta \frac{h}{2} \mathbf{V}_3, \quad (44.48)$$

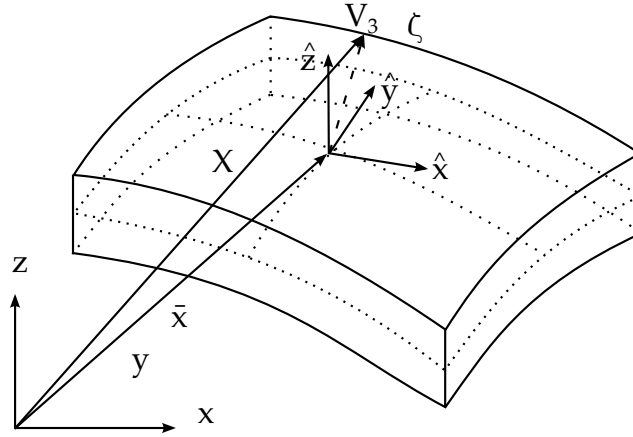


Figure 44.4. Geometry of a shell.

$$\mathbf{u} = \bar{\mathbf{u}} + \zeta \frac{h}{2} \mathbf{U}. \quad (44.49)$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are the position vector and displacement of the reference surface, respectively. \mathbf{V}_3 is the fiber director and \mathbf{U} is the displacement resulting from the fiber rotation (see Figures 44.4 and 44.5). h is the length of the fiber.

With the mesh-free approximation, the motion and displacements are given by

$$\mathbf{x}(\zeta, \eta, \zeta) = \bar{\mathbf{x}}(\zeta, \eta) + \mathbf{V}(\zeta, \eta, \zeta) \approx \sum_{I=1}^{NP} \tilde{\Psi}_I(\zeta, \eta) \mathbf{x}_I + \sum_{I=1}^{NP} \tilde{\Psi}_I(\zeta, \eta) \frac{\zeta h_I}{2} \mathbf{V}_{3I}, \quad (44.50)$$

$$\mathbf{u}(\zeta, \eta, \zeta) = \bar{\mathbf{u}}(\zeta, \eta) + \mathbf{U}(\zeta, \eta, \zeta) \approx \sum_{I=1}^{NP} \tilde{\Psi}_I(\zeta, \eta) \mathbf{u}_I + \sum_{I=1}^{NP} \tilde{\Psi}_I(\zeta, \eta) \frac{\zeta h_I}{2} [-\mathbf{V}_{2I} \quad \mathbf{V}_{1I}] \begin{Bmatrix} \alpha_I \\ \beta_I \end{Bmatrix}, \quad (44.51)$$

where \mathbf{x}_I and \mathbf{u}_I are the global coordinates and displacements at mesh-free node I , respectively. \mathbf{V}_{3I} is the unit vector of the fiber director and \mathbf{V}_{1I} , \mathbf{V}_{2I} are the base vectors of the nodal coordinate system at node I . α_I and β_I are the rotations of the director vector \mathbf{V}_{3I} about the \mathbf{V}_{1I} and \mathbf{V}_{2I} axes. h_I is the thickness. The variables with a superscripted bar

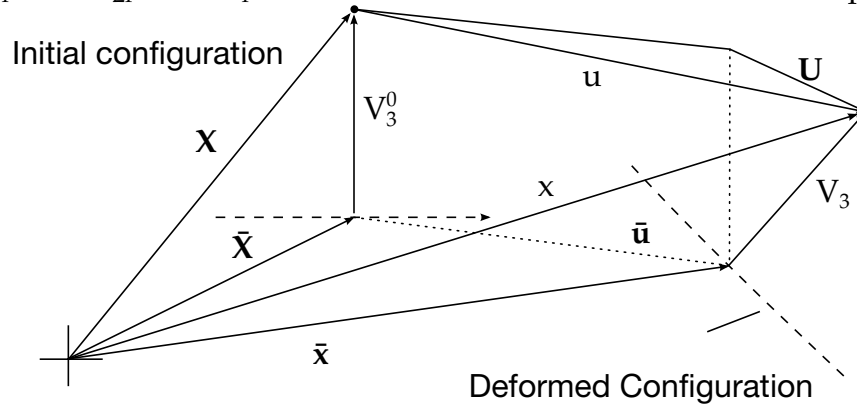


Figure 44.5. Deformation of a shell.

refer to the shell mid-surface. $\tilde{\Psi}_I$ is the 2D mesh-free shape functions constructed based

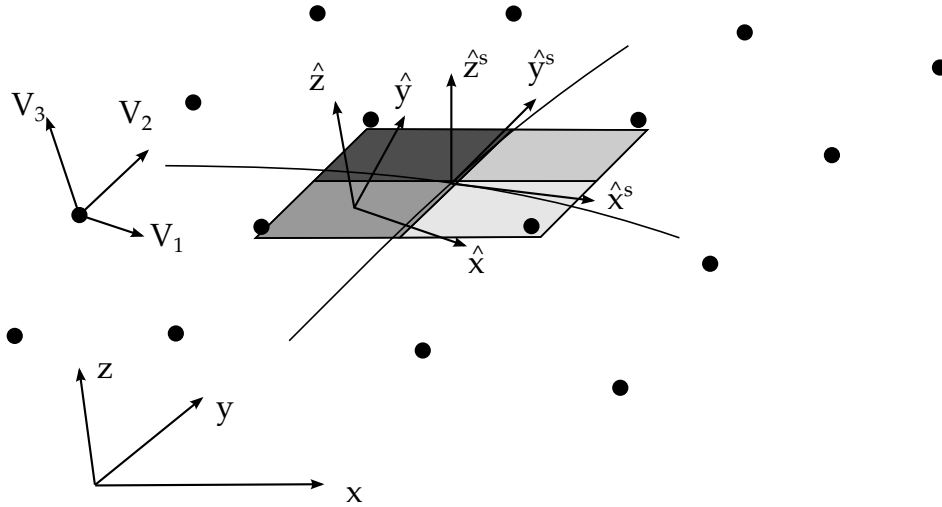


Figure 44.6. Local co-rotational and nodal coordinate systems.

on one of the two mesh-free surface representations described in the previous section, with (ξ, η) either the parametric coordinates or local coordinates of the evaluated point.

The local co-rotational coordinate system $(\hat{x}, \hat{y}, \hat{z})$ is defined at each integration point on the shell reference surface, with \hat{x} and \hat{y} tangent to the reference surface and \hat{z} in the thickness direction (see Figure 44.6). The base vectors are given as

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{x}_{,\xi}}{\|\mathbf{x}_{,\xi}\|}, \hat{\mathbf{e}}_3 = \frac{\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}}{\|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\|}, \hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1. \quad (44.52)$$

In order to describe the fiber rotations of a mesh-free node in a shell, we introduce a nodal coordinate system whose three base vectors are \mathbf{V}_1 , \mathbf{V}_2 and \mathbf{V}_3 , see Figure 44.6, where \mathbf{V}_3 is the fiber director at the node and \mathbf{V}_1 , \mathbf{V}_2 are defined as follows

$$\mathbf{V}_1 = \frac{\hat{\mathbf{x}} \times \mathbf{V}_3}{|\hat{\mathbf{x}} \times \mathbf{V}_3|}, \mathbf{V}_2 = \mathbf{V}_3 \times \mathbf{V}_1. \quad (44.53)$$

The rotation of the fiber director is then obtained from the global rotations:

$$\begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \Delta\theta, \quad \Delta\theta = [\Delta\theta_1 \quad \Delta\theta_2 \quad \Delta\theta_3]^T. \quad (44.54)$$

In the local co-rotational coordinate system, the motion and displacements are approximated by the mesh-free shape functions

$$\hat{x}_i = \sum_{I=1}^{NP} \tilde{\Psi}_I \hat{x}_{iI} + \zeta \sum_{I=1}^{NP} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{3iI}, \quad (44.55)$$

$$\hat{u}_i = \sum_{I=1}^{NP} \tilde{\Psi}_I \hat{u}_{iI} + \zeta \sum_{I=1}^{NP} \tilde{\Psi}_I \frac{h_I}{2} [-\hat{V}_{2iI} \quad \hat{V}_{1iI}] \begin{Bmatrix} \alpha_I \\ \beta_I \end{Bmatrix}. \quad (44.56)$$

The Lagrangian smoothed strains [Chen et al. 2001b] are given by

$$\tilde{\varepsilon}^m = \sum_I \tilde{\mathbf{B}}_I^m \hat{\mathbf{d}}_I, \tilde{\varepsilon}^b = \zeta \sum_I \tilde{\mathbf{B}}_I^b \hat{\mathbf{d}}_I, \tilde{\varepsilon}^s = \sum_I \tilde{\mathbf{B}}_I^s \hat{\mathbf{d}}_I, \quad (44.57)$$

where the smoothed strain operators are calculated by averaging the consistent strain operators over an area around the evaluated point

$$\tilde{\mathbf{B}}_I^m(\mathbf{x}_I) = \frac{1}{A_I} \int_{\Omega_I} \hat{\mathbf{B}}_I^m dA, \tilde{\mathbf{B}}_I^b(\mathbf{x}_I) = \frac{1}{A_I} \int_{\Omega_I} \hat{\mathbf{B}}_I^b dA, \tilde{\mathbf{B}}_I^s(\mathbf{x}_L) = \frac{1}{A_L} \int_{\Omega_L} \hat{\mathbf{B}}_I^s dA, \quad (44.58)$$

with

$$\hat{\mathbf{B}}_I^m = \begin{bmatrix} \tilde{\Psi}_{I,x} & 0 & 0 & -J_{13}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2xI} & J_{13}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1xI} \\ 0 & \tilde{\Psi}_{I,y} & 0 & -J_{23}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2yI} & J_{23}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1yI} \\ \tilde{\Psi}_{I,y} & \tilde{\Psi}_{I,x} & 0 & -J_{23}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2xI} - J_{13}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2yI} & J_{23}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1xI} + J_{13}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1yI} \end{bmatrix}, \quad (44.59)$$

$$\hat{\mathbf{B}}_I^b = \begin{bmatrix} 0 & 0 & 0 & -\tilde{\Psi}_{I,x} \frac{h_I}{2} \hat{V}_{2xI} & \tilde{\Psi}_{I,x} \frac{h_I}{2} \hat{V}_{1xI} \\ 0 & 0 & 0 & -\tilde{\Psi}_{I,y} \frac{h_I}{2} \hat{V}_{2yI} & \tilde{\Psi}_{I,y} \frac{h_I}{2} \hat{V}_{1yI} \\ 0 & 0 & 0 & -\tilde{\Psi}_{I,y} \frac{h_I}{2} \hat{V}_{2xI} - \tilde{\Psi}_{I,x} \frac{h_I}{2} \hat{V}_{2yI} & \tilde{\Psi}_{I,y} \frac{h_I}{2} \hat{V}_{1xI} + \tilde{\Psi}_{I,x} \frac{h_I}{2} \hat{V}_{1yI} \end{bmatrix}, \quad (44.60)$$

$$\hat{\mathbf{B}}_I^s = \begin{bmatrix} 0 & 0 & \tilde{\Psi}_{I,y} & -J_{33}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2yI} - J_{23}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2zI} & J_{33}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1yI} + J_{23}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1zI} \\ 0 & 0 & \tilde{\Psi}_{I,x} & -J_{33}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2xI} - J_{13}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{2zI} & J_{33}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1xI} + J_{13}^{-1} \tilde{\Psi}_I \frac{h_I}{2} \hat{V}_{1zI} \end{bmatrix}, \quad (44.61)$$

and \mathbf{J}^{-1} is the inverse of the Jacobian matrix at the integration point. The local degrees-of-freedom are

$$\hat{\mathbf{d}}_I = [\hat{u}_{xI} \quad \hat{u}_{yI} \quad \hat{u}_{zI} \quad \alpha_I \quad \beta_I]^T. \quad (44.62)$$

The internal nodal force vector is

$$\hat{\mathbf{F}}_I^{\text{int}} = \int_{\Omega} \tilde{\mathbf{B}}_I^m \hat{\sigma} d\Omega + \int_{\Omega} \zeta \tilde{\mathbf{B}}_I^b \hat{\sigma} d\Omega + \int_{\Omega} \tilde{\mathbf{B}}_I^s \hat{\sigma} d\Omega. \quad (44.63)$$

The above integrals are calculated with the local boundary integration method. Each background finite element is divided into four integration zones, shown as Ω_I in Figure 44.7. In order to avoid shear locking in the analysis of thin shells, the shear term (third term in Eq. (44.63)), should be under-integrated by using one integration zone in

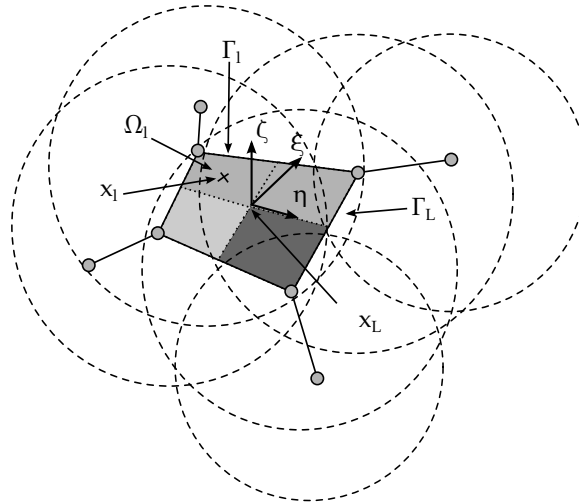


Figure 44.7. Integration scheme for mesh-free shells.

each background element (Ω_L in [Figure 44.7](#)). Accordingly, the co-rotational coordinate systems are defined separately at the center of each integration zone, as shown in [Figure 44.6](#).

The use of the updated Lagrangian formulation implies that the reference coordinate system is defined by the co-rotational system in the configuration at time t . Therefore, the local nodal force and displacement vectors referred to this coordinate system must be transformed to the global coordinate system prior to assemblage.

45

Linear shells

45.1 Shells for Linear Analysis

It is common to construct elements for linear analysis by the superimposition of a plate and a membrane element. If the base plates and membrane elements involve only three translational degrees-of-freedom and two in-plane rotational degrees-of-freedom, the resulting element then contains 5 degrees-of-freedom per node since there is an unconstrained rotational degree-of-freedom normal to the mid surface of the shell. This unconstrained mode can cause problems when linking the shell to other elements such as beam elements in three-dimensional space. For this reason, the linear elements in LS-DYNA are based on published formulations that include a drilling degree-of-freedom, which is added to the membrane part of the element to form a 24 degree-of-freedom shell element. These elements pass all patch tests, have 6 rigid body modes, and have no spurious mechanisms.

45.2 Wilson's Shell (element #20)

This quadrilateral element is constructed as described above and is discussed in more detail by Wilson [2000]. The triangular element, which is an 18 degree-of-freedom complement to the quadrilateral elements, follows the same procedure. In a linear analysis in LS-DYNA, automatic sorting is invoked if a mesh has both quadrilateral and triangular elements within a single part ID. This sorting ensures the proper treatment of triangles.

45.2.1 Plate Element

The 4 node quadrilateral plate element is based on the 8 node, quadratic quadrilateral plate element, which has 16 rotational degrees-of-freedom, i.e., two per nodal

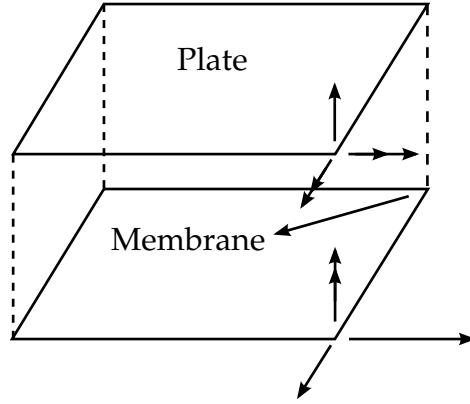


Figure 45.1. Shell as assembly of plate and membrane elements

point. The implementation in LS-DYNA directly follows the textbook by Wilson [2000] where the complete details of the element are provided. A condensed overview is given here. The shell theory makes the following assumptions:

- The fiber remains straight and inextensible
- The normal stress in the thickness direction is zero

The local x and y rotations of the shell are interpolated from the equations:

$$\begin{aligned}\theta_x(r, s) &= \sum_{i=1}^4 N_i(r, s) \theta_{xi} + \sum_{i=5}^8 N_i(r, s) \Delta \theta_{xi} \\ \theta_y(r, s) &= \sum_{i=1}^4 N_i(r, s) \theta_{yi} + \sum_{i=5}^8 N_i(r, s) \Delta \theta_{yi},\end{aligned}\tag{45.1}$$

where nodes 5-8 are at the mid side of the element. The interpolation functions are given by

$$\begin{aligned}N_1 &= \frac{1}{4}(1-r)(1-s) & N_5 &= \frac{1}{2}(1-r^2)(1-s) \\ N_2 &= \frac{1}{4}(1+r)(1-s) & N_6 &= \frac{1}{2}(1+r)(1-s^2) \\ N_3 &= \frac{1}{4}(1+r)(1+s) & N_7 &= \frac{1}{2}(1-r^2)(1+s) \\ N_4 &= \frac{1}{4}(1-r)(1+s) & N_8 &= \frac{1}{2}(1-r)(1-s^2).\end{aligned}\tag{45.2}$$

In his formulation, Wilson resolves the rotation of the mid side node into tangential and normal components relative to the shell edges. The tangential component is set to zero leaving the normal component as the unknown, which reduces the rotational degrees-of-freedom from 16 to 12, see [Figure 45.2](#).

$$\begin{aligned}\Delta \theta_x &= \sin \alpha_{ij} \Delta \theta_{ij} \\ \Delta \theta_y &= -\cos \alpha_{ij} \Delta \theta_{ij}\end{aligned}\tag{45.3}$$

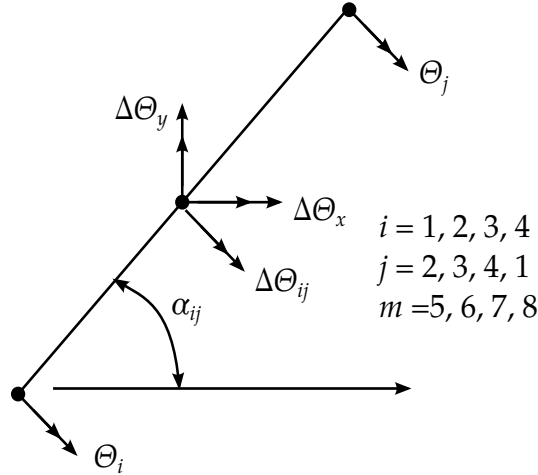


Figure 45.2. Element edge [Wilson, 2000]

$$\begin{aligned}\theta_x(r, s) &= \sum_{i=1}^4 N_i(r, s) \theta_{xi} + \sum_{i=5}^8 M_{xi}(r, s) \Delta \theta_i \\ \theta_y(r, s) &= \sum_{i=1}^4 N_i(r, s) \theta_{yi} + \sum_{i=5}^8 M_{yi}(r, s) \Delta \theta_i.\end{aligned}\tag{45.4}$$

Ultimately, the 4 mid side rotations are eliminated by using static condensation, a procedure that makes this shell very costly if used in explicit calculations.

The local x and y displacements relative to the mid surface are functions of the z -coordinate and rotations:

$$\begin{aligned}u_x(r, s) &= z \theta_y(r, s) \\ u_y(r, s) &= -z \theta_x(r, s).\end{aligned}\tag{45.5}$$

Wilson shows, where it is assumed that the normal displacement along each side is cubic, that the transverse shear strain along each side is given by,

$$\gamma_{ij} = \frac{1}{L} (u_{zj} - u_{zi}) - \frac{1}{2} (\theta_i + \theta_j) - \frac{2}{3} \Delta \theta_{ij},\tag{45.6}$$

which can be rewritten, referring to [Figure 45.3](#) as:

$$\gamma_{ij} = \frac{1}{L} (u_{zj} - u_{zi}) - \frac{\sin \alpha_{ij}}{2} (\theta_{xi} + \theta_{xj}) + \frac{\cos \alpha_{ij}}{2} (\theta_{yi} + \theta_{yj}) - \frac{2}{3} \Delta \theta_{ij},\tag{45.7}$$

The nodal shears are then written in terms of the side shears as

$$\begin{bmatrix} \gamma_{ij} \\ \gamma_{ki} \end{bmatrix} = \begin{bmatrix} \cos \alpha_{ij} & \sin \alpha_{ij} \\ \cos \alpha_{ki} & \sin \alpha_{ki} \end{bmatrix} \begin{bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{bmatrix},\tag{45.8}$$

which can be inverted to obtain the nodal shears:

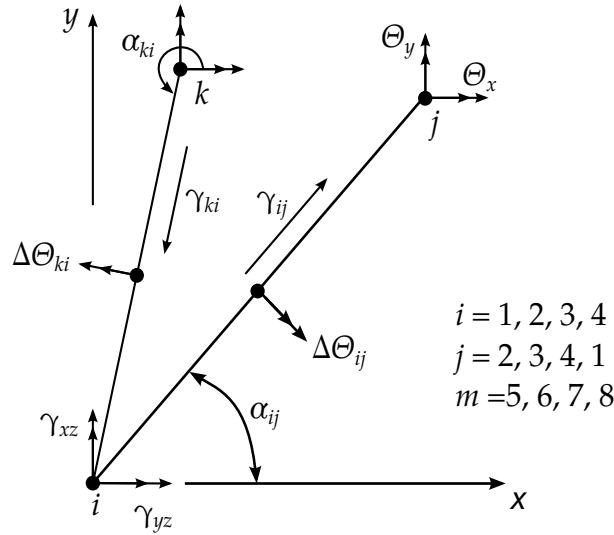


Figure 45.3. Nodal and edge shear strains [Wilson 2000].

$$\begin{bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} = \frac{1}{\cos\alpha_{ij}\sin\alpha_{ki} - \cos\alpha_{ki}\sin\alpha_{ij}} \begin{bmatrix} \sin\alpha_{ki} & -\cos\alpha_{ki} \\ -\sin\alpha_{ij} & \cos\alpha_{ij} \end{bmatrix} \begin{bmatrix} \gamma_{ij} \\ \gamma_{ki} \end{bmatrix}. \quad (45.9)$$

The standard bilinear basis functions are used to interpolate the nodal shears to the integration points.

45.2.2 Membrane Element

The membrane element, which is also coded from Wilson's textbook [2000], is based on the eight node isoparametric element, see [Figure 45.4](#).

The inplane displacement field for the 8 node membrane is interpolated, using the serendipity shape functions with the mid-side relative displacements, from:

$$\begin{aligned} u_x(r, s) &= \sum_{i=1}^4 N_i(r, s) u_{xi} + \sum_{i=5}^8 N_i(r, s) \Delta u_{xi} \\ u_y(r, s) &= \sum_{i=1}^4 N_i(r, s) u_{yi} + \sum_{i=5}^8 N_i(r, s) \Delta u_{yi}. \end{aligned} \quad (45.10)$$

It is desired to replace the mid side relative displacement by drilling rotations at the corner nodes. Consider [Figure 45.5](#): the mid-side normal displacements along the edge are parabolic, i.e.,

$$\Delta u_{ij} = \frac{L_{ij}}{8} (\Delta \theta_j - \Delta \theta_i), \quad (45.11)$$

while the mid-side tangential displacements are interpolated linearly from the end node displacements, thus,

$$\Delta u_x(r, s) = \cos \alpha_{ij} \Delta u_{ij} = \cos \alpha_{ij} \frac{L_{ij}}{8} (\Delta \theta_j - \Delta \theta_i) \quad (45.12)$$

$$\Delta u_y(r, s) = -\sin \alpha_{ij} \Delta u_{ij} = -\sin \alpha_{ij} \frac{L_{ij}}{8} (\Delta \theta_j - \Delta \theta_i),$$

$$\begin{aligned} u_x(r, s) &= \sum_{i=1}^4 N_i(r, s) u_{xi} + \sum_{i=5}^8 M_{xi}(r, s) \Delta \theta_i \\ u_y(r, s) &= \sum_{i=1}^4 N_i(r, s) u_{yi} + \sum_{i=5}^8 M_{yi}(r, s) \Delta \theta_i. \end{aligned} \quad (45.13)$$

This element has one singularity in the *drilling mode* of equal corner rotations, see Figure 45.6.

Ibrahimbegovic and Wilson [1991] added a penalty formulation to the potential energy of the element to eliminate the singularity. The following penalty term connects the averaged nodal rotation to the continuum mechanics rotation

$$\frac{1}{2} \left(\frac{\partial u_x}{\partial y} - \frac{\partial u_y}{\partial x} \right) - \omega \quad (45.14)$$

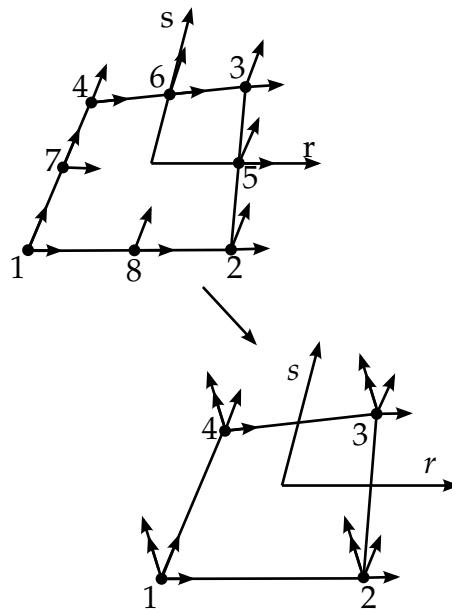


Figure 45.4. Eight node membrane element.

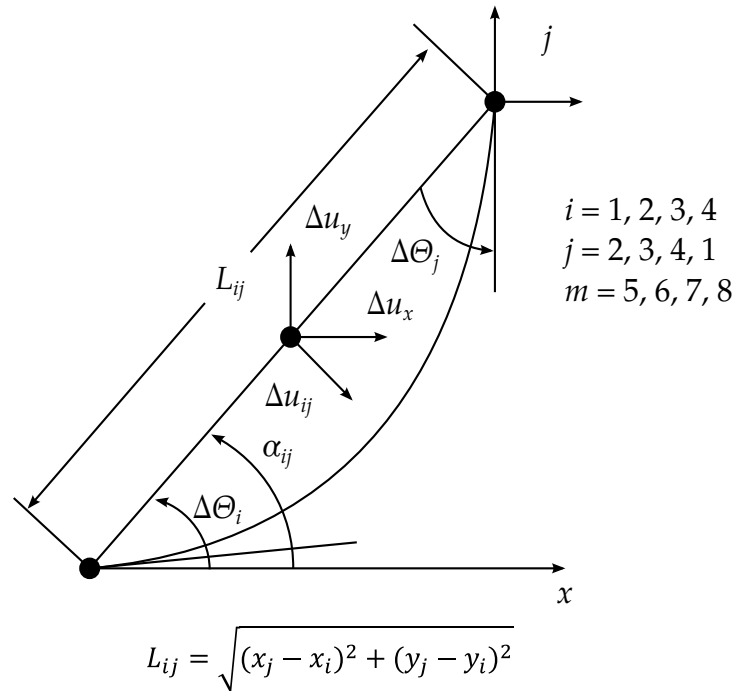


Figure 45.5. Corner node drilling rotations and mid side edge normal displacement [Wilson, 2000].

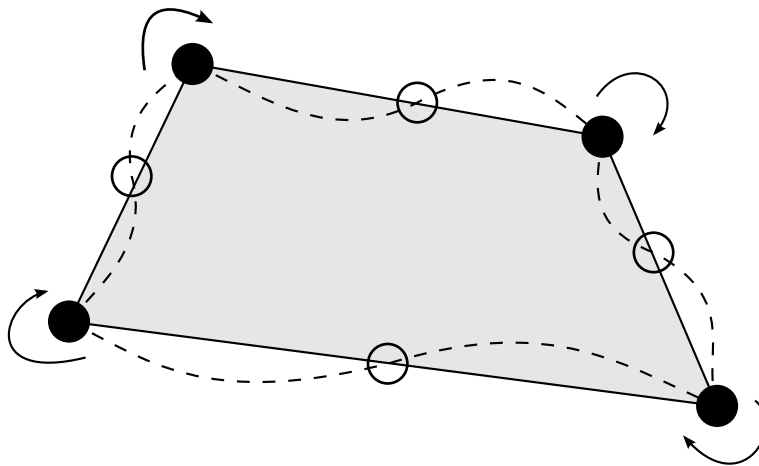


Figure 45.6. Zero energy mode

at the center of the element. The element performance is highly insensitive to the chosen value of the penalty factor and some fraction of the elastic moduli, G or E , is frequently used.

- 5, 8 or 9 point quadrature can be applied. The 5 and 8 point schemes induce a 'soft' first deformational mode, whereas the 9 point Gaussian quadrature results in a stiffer mode.
- A membrane locking correction (Taylor) is applied to (i) alleviate a membrane-bending interaction associated with the drilling degrees of freedom and (ii) allow

the standard application of the consistent nodal load at the edge. The correction has a slight stiffening effect (see e.g. Cook Cantilever).

- A warping correction is applied using the rigid link correction (see [Figure 45.7](#)).

45.3 Assumed Strain/Membrane with Drilling Degree-of-freedom (element #18)

45.3.1 Membrane Element

Formulation is the same as above for element type 20.

45.3.2 Plate Element

The Discrete Kirchhoff Quadrilateral element is an excellent thin shell element based on

- Rotational field is interpolated using the 8-node isoparametric parent element.
- Transverse displacement w assumed as cubic along the sides and collocated along the sides and at the nodes using the Kirchhoff condition that equates the fiber rotation to the slope. The Kirchhoff assumptions are satisfied along the entire boundary of the element.
- The rotational field about an axis parallel to the side is constrained linearly along the sides.

The warping correction is applied as above.

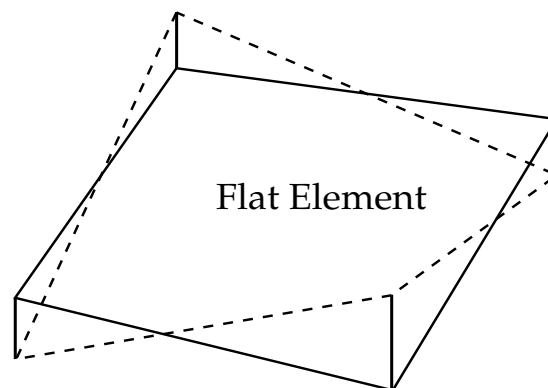


Figure 45.7. Flat element

45.4 Differences between Element Types 18 and 20.

The DKQ does not account for transverse shear because it locally enforces the Kirchhoff condition. Hence, element type 20 is better for layered composites and thick plates.

46

Random Geometrical Imperfections

46.1 Introduction to Random Geometrical Imperfections Using Karhunen-Loève Expansions

There are different methods of incorporating imperfections, depending on the availability of accurate imperfection data. The method implemented into LS-DYNA v971 uses a spectral decomposition of geometrical or thickness uncertainty, more specifically through using Karhunen-Loève expansions (see *PERTURBATION keyword). To specify the covariance of the random field of the geometrical imperfections or thickness variation, two methods are available. The first is to use available experimentally-measured imperfection fields as input for a principal component analysis based on pattern (face) recognition literature. This method reduces the cost of the resulting eigen-analysis. The second is to specify the covariance function analytically and to solve the resulting Friedholm integral equation of the second kind using a wavelet-Galerkin approach, also obtained from literature. Six different analytical covariance kernels (e.g., exponential and triangular) are available for selection.

46.2 Methodology

46.2.1 Generation of random fields using Karhunen-Loève expansion

The Karhunen-Loève expansion (e.g., Ghanem and Spanos [2003]) provides an attractive way of representing a random (stochastic) field (process) through a spectral decomposition, ω , as a function of x (e.g., two spatial variables):

$$\omega(x, \theta) = \bar{\omega}(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \zeta_i(\theta) f_i(x), \quad (46.1)$$

where the ζ_i are uncorrelated zero-mean random variables with unit variance, and $\bar{\omega}(x)$ is the average random field or mean of the process. The functions f_i are the eigenfunctions of the covariance kernel, C , with λ_i the associated eigenvalues, obtained from the spectral

decomposition of the covariance function via the solution on a domain D of the Fredholm integral equation of the second kind,

$$\int_D C(x_1, x_2) f_i(x_1) dx_1 = \lambda_i f_i(x_2). \quad (46.2)$$

The eigenfunctions form an orthogonal set

$$\int_D f_i(x) f_j(x) dx = \delta_{ij}. \quad (46.3)$$

Normally, a finite M number of terms are kept in the series expansion:

$$\omega(x, \theta) = \bar{\omega}(x) + \sum_{i=1}^M \sqrt{\lambda_i} \xi_i(\theta) f_i(x). \quad (46.4)$$

If ω is Gaussian, then ξ_i are also Gaussian. For non-Gaussian processes (with arbitrary but specified marginal distributions), ξ_i are unknown. Phoon et al ([2002a], [2005]) give an iterative procedure for obtaining ξ_i given a target marginal distribution.

46.2.2 Solution of Fredholm integral of the second kind for analytical covariance functions

The Wavelet-Galerkin method (Phoon et al [2002]) is used to perform the solution, and can be described as follows.

By defining a set of basis functions: $\varphi_1(x), \varphi_2(x), \dots, \varphi_N(x)$, each eigenfunction $f_i(x)$ can be approximated by the linear combination:

$$f_i(x) = \sum_{k=1}^N d_{ik} \varphi_k(x), \quad (46.5)$$

where the d_{ik} are constant coefficients. By substituting (46.5) into Fredholm equation (46.2) (using a scalar x (one-dimensional random process) as an example) and writing as an error:

$$\int_D C(x_1, x_2) \sum_{k=1}^N d_{ik} \varphi_k(x_1) dx_1 - \lambda_i \sum_{k=1}^N d_{ik} \varphi_k(x_1) = 0. \quad (46.6)$$

By making the error orthogonal to the basis functions:

$$\int_D \left[\int_D C(x_1, x_2) \sum_{k=1}^N d_{ik} \varphi_k(x_1) dx_1 - \lambda_i \sum_{k=1}^N d_{ik} \varphi_k(x_1) \right] \varphi_j(x_2) dx_2 = 0, \quad (46.7)$$

we get

$$\sum_{k=1}^N d_{ik} \left[\iint_D C(x_1, x_2) \varphi_k(x_1) \varphi_j(x_2) dx_1 dx_2 \right] - \lambda_i \sum_{k=1}^N d_{ik} \left[\int_D \varphi_k(x_2) \varphi_j(x_2) dx_2 \right] = 0, \quad (46.8)$$

or the eigensystem

$$\mathbf{AD} = \mathbf{\Lambda BD} \quad (46.9)$$

with $\Lambda_{ij} = \delta_{ij}\lambda_j$. Orthogonal wavelets $\psi(x)$ are used ($\int_0^1 \psi_j(x)\psi_k(x)dx = h_j\delta_{jk}$) as basis functions,

$$\mathbf{f}_i(x) = \sum_{k=1}^N d_{ik} \psi_k(x) = \boldsymbol{\psi}^T(x) \mathbf{D}^{(i)}, \quad (46.10)$$

so that the covariance function can be expressed as:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^N \sum_{k=1}^N \bar{A}_{jk} \psi_j(\mathbf{x}_1) \psi_k(\mathbf{x}_2) = \boldsymbol{\psi}^T(\mathbf{x}_1) \bar{\mathbf{A}} \boldsymbol{\psi}(\mathbf{x}_2), \quad (46.11)$$

where $\bar{\mathbf{A}}$ is the 2D wavelet transform of $C(\mathbf{x}_1, \mathbf{x}_2)$ given by

$$\bar{A}_{jk} = \frac{1}{h_j h_k} \int_0^1 \int_0^1 C(\mathbf{x}_1, \mathbf{x}_2) \psi_j(\mathbf{x}_1) \psi_k(\mathbf{x}_2) d\mathbf{x}_1 d\mathbf{x}_2. \quad (46.12)$$

Substituting (46.10) and (46.11) into (46.2), we again get an eigenvalue problem

$$\boldsymbol{\psi}^T(x) \bar{\mathbf{A}} \mathbf{H} \mathbf{D}^{(i)} = \lambda_i \boldsymbol{\psi}^T(x) \mathbf{D}^{(i)}. \quad (46.13)$$

Or, equating coefficients of $\boldsymbol{\psi}^T$ and using the transformation $\hat{\mathbf{D}}^{(i)} = \mathbf{H}^{1/2} \mathbf{D}^{(i)}$ and $\hat{\mathbf{A}} = \mathbf{H}^{1/2} \bar{\mathbf{A}} \mathbf{H}^{1/2}$, the eigensystem

$$\hat{\mathbf{A}} \hat{\mathbf{D}}^{(i)} = \lambda_i \hat{\mathbf{D}}^{(i)}. \quad (46.14)$$

The eigenvectors from (46.14) are transformed to the eigen functions (of (46.9)) by the equation:

$$\mathbf{f}_i(x) = \boldsymbol{\psi}^T(x) \mathbf{H}^{-1/2} \hat{\mathbf{D}}^{(i)}. \quad (46.15)$$

The double integral in (46.12) is constructed using two successive 1D discrete wavelet transforms in the form of Mallat's tree algorithm (Phoon et al [2002a]). Haar wavelets are used because of their simplicity and ability to capture the field characteristics. The eigenfunctions (46.15) and associated eigenvalues λ_i can be used to construct random fields (using (46.4)) with the same second-order statistics as the covariance model used.

The available covariance functions (Ghanem and Spanos [2003]), are:

- Exponential covariance function (First-order Markov process (autoregressive)),

$$C(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(\frac{-|\mathbf{x}_1 - \mathbf{x}_2|}{L_c}\right) \quad (46.16)$$

- Triangular covariance function

$$C(x_1, x_2) = 1 - \frac{|x_1 - x_2|}{L_c} \quad (46.17)$$

- Sine covariance function

$$C(x_1, x_2) = \frac{\sin L_c(x_1 - x_2)}{L_c(x_1 - x_2)} \quad (46.18)$$

- Squared exponential covariance function

$$C(x_1, x_2) = \exp\left(\frac{-|x_1 - x_2|^2}{L_c}\right) \quad (46.19)$$

- Wiener-Levy covariance function

$$C(x_1, x_2) = \min(x_1, x_2) \quad (46.20)$$

- Uniformly modulated nonstationary covariance function

$$C(x_1, x_2) = \exp(-(x_1 - x_2)) \exp \frac{-|x_1 - x_2|}{L_c} \quad (46.21)$$

with L_c the correlation length in the respective direction.

46.2.3 Generating eigenfunctions from experimentally measured fields

If experimentally measured random fields are available, then the eigenfunctions and eigenvalues in (46.1) or (46.4) can be determined from the second-order statistics of the measurements. Following the pattern recognition method proposed by Turk and Pentland [1991], the procedure described next is used.

Given a set of M field measurements, e.g., geometrical imperfections on an $N1 \times N2$ mesh, we can represent the measurements as 1-D vectors of length $N1 \times N2$, i.e., $\Gamma_1, \dots, \Gamma_M$. The average vector is defined by

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (46.22)$$

allowing us to define the deviation of each measured field from the average, also as a vector: $\Phi_i = \Gamma_i - \Psi$. Combining the deviation vectors into a covariance matrix, \mathbf{C} , we get

$$\mathbf{C} = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = \mathbf{A} \mathbf{A}^T, \quad (46.23)$$

where $\mathbf{A} = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$. The covariance matrix has a set of orthonormal eigenvectors, \mathbf{f}_i , and associated eigenvalues, λ_i obtained through a principal component analysis, i.e.,

$$\mathbf{A} \mathbf{A}^T \mathbf{f}_i = \lambda_i \mathbf{f}_i. \quad (46.24)$$

The eigenpairs are chosen such that

$$\lambda_i = \frac{1}{M} \sum_{n=1}^M (\mathbf{f}_i^T \Phi_n)^2, \quad (46.25)$$

is a maximum, subject to

$$\mathbf{f}_i^T \mathbf{f}_k = \delta_{ik}, \quad (46.26)$$

with δ the Kronecker delta. As the size of the covariance matrix is $(N1 \times N2)^2$, determining the eigenvectors and eigenvalues in (41.24) can be a time-consuming and memory-intensive task for large measurement meshes. This computation can be simplified if the number of measurement samples is less than the mesh count ($M < N1 \times N2$), as there are then only $M - 1$, rather than $N1 \times N2$, meaningful eigenvectors. This is done by considering the eigenvectors of another matrix $\mathbf{L} = \mathbf{A}^T \mathbf{A}$, as embodied in the eigensystem

$$\mathbf{A}^T \mathbf{A} \mathbf{v}_i = \mu_i \mathbf{v}_i. \quad (46.27)$$

Premultiplying both sides in (21.1.6) by \mathbf{A} , we get

$$\mathbf{A} \mathbf{A}^T \mathbf{A} \mathbf{v}_i = \mu_i \mathbf{A} \mathbf{v}_i, \quad (46.28)$$

which implies that the $M - 1$ eigenvectors, $\mathbf{A} \mathbf{v}_i$, are also eigenvectors of $\mathbf{A} \mathbf{A}^T$ or \mathbf{C} . The eigensystem in (21.1.6), however, is only size $(M \times M)$, as $\mathbf{L} = \mathbf{A}^T \mathbf{A}$, where $L_{nm} = \Phi_m^T \Phi_n$. Once the eigenvectors of \mathbf{L} are obtained, the required eigenfunctions, \mathbf{f}_i , are recovered through the linear combination

$$\mathbf{f}_i = \sum_{k=1}^M v_{ik} \Phi_k, \quad i = 1, \dots, M \quad (46.29)$$

The $M - 1$ eigenvalues of \mathbf{L} and \mathbf{C} are identical, i.e., $\lambda_i = \mu_i$. Finally \mathbf{f}_i and λ_i are used in (46.4) to construct the required Karhunen-Loève expansion of the random fields.

47

Frequency Domain

47.1 Frequency Response Functions

Frequency response function (FRF) is a characteristic of a system that has a measured or computed response resulting from a known applied input. Mathematically it is a transfer function and expresses the structural response to an applied force as a function of frequency. The response can be given in terms of displacement, velocity, or acceleration. Frequency response functions are complex functions, with real and imaginary components. They can also be written in terms of magnitude and phase pairs.

47.1.1 FRF Computations

FRF is computed using mode superposition method, in frequency domain. When damping is included, the dynamic response of a system is governed by

$$\mathbf{m}\ddot{\mathbf{u}} + \mathbf{c}\dot{\mathbf{u}} + \mathbf{k}\mathbf{u} = \mathbf{p}(t), \quad (47.1.1)$$

where \mathbf{m} , \mathbf{c} and \mathbf{k} are the mass, damping and stiffness matrices, $\mathbf{p}(t)$ is the external force.

Using the mode superposition method, the displacement response can be expressed by

$$\mathbf{u} = \sum_{n=1}^N \phi_n q_n(t) = \Phi \mathbf{q}, \quad (47.1.2)$$

where ϕ_n is the n -th mode shape and $q_n(t)$ is the n -th modal coordinates.

With the substitution of Equation (47.1.2) into Equation (47.1.1), the governing equation can be rewritten as

$$\mathbf{m}\Phi\ddot{\mathbf{q}} + \mathbf{c}\Phi\dot{\mathbf{q}} + \mathbf{k}\Phi\mathbf{q} = \mathbf{p}(t). \quad (47.1.3)$$

Pre-multiplying by Φ^T gives

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{p}(t), \quad (47.1.4)$$

The orthogonality of natural modes implies that the following square matrices are diagonal:

$$\mathbf{M} \equiv \Phi^T \mathbf{m} \Phi, \quad \mathbf{K} \equiv \Phi^T \mathbf{k} \Phi, \quad (47.1.5)$$

where the diagonal elements are

$$\mathbf{M}_n = \phi_n^T \mathbf{m} \phi_n, \quad \mathbf{K}_n = \phi_n^T \mathbf{k} \phi_n \quad (47.1.6)$$

Since \mathbf{m} and \mathbf{k} are positive definite, the diagonal elements of \mathbf{M} and \mathbf{K} are positive. They are related by

$$\mathbf{K}_n = \omega_n^2 \mathbf{M}_n. \quad (47.1.7)$$

The square matrix \mathbf{C} is obtained similarly as follows

$$\mathbf{C} = \Phi^T \mathbf{c} \Phi. \quad (47.1.8)$$

\mathbf{C} may or may not be diagonal, depending on the distribution of damping in the system. If \mathbf{C} is diagonal (the diagonal elements are $C_n = \phi_n^T c \phi_n$), [Equation \(47.1.4\)](#) represents N uncoupled differential equations in modal coordinates q_n , and the system is said to have classical damping and the systems possess the same natural modes as those of the undamped system. Only the classical damping is considered in this approach.

The right hand side vector (generalized force) $\mathbf{P}(t)$ is

$$\mathbf{P}(t) = \Phi^T p(t) \quad (47.1.9)$$

For an N -DOF system with classical damping, each of the N differential equations in modal coordinates is

$$M_n \ddot{q}_n + C_n \dot{q}_n + K_n q_n = P_n(t) \quad (47.1.10)$$

or,

$$\ddot{q}_n + 2\zeta_n \omega_n \dot{q}_n + \omega_n^2 q_n = \frac{P_n(t)}{M_n} \quad (47.1.11)$$

where the modal damping coefficient ζ_n is defined as

$$\zeta_n = \frac{C_n}{2M_n \omega_n} \quad (47.1.12)$$

Applying Fourier transform to both sides of [Equation \(47.1.11\)](#), one obtains

$$(-\omega^2 + 2i\zeta_n \omega_n \omega + \omega_n^2) q_n(\omega) = \frac{P_n(\omega)}{M_n} \quad (47.1.13)$$

The structural displacement response in frequency domain can be represented as

$$\mathbf{u}(\omega) = \sum_{n=1}^N \frac{\phi_n}{(-\omega^2 + 2i\zeta_n \omega_n \omega + \omega_n^2)} \frac{P_n(\omega)}{M_n} \quad (47.1.14)$$

Thus the displacement frequency response function (Compliance) can be expressed as (suppose that the excitation is applied at node j and the response is evaluated for node k)

$$\mathbf{FRF}_u(x_j, x_k, \omega) = \sum_{n=1}^N \frac{\phi_n(x_k)}{(-\omega^2 + 2i\zeta_n\omega_n\omega + \omega_n^2)} \frac{\tilde{P}_n(x_j)}{M_n} \quad (47.1.15)$$

The velocity frequency response function (Mobility) can be expressed as

$$\mathbf{FRF}_v(x_j, x_k, \omega) = \omega i \sum_{n=1}^N \frac{\phi_n(x_k)}{(-\omega^2 + 2i\zeta_n\omega_n\omega + \omega_n^2)} \frac{\tilde{P}_n(x_j)}{M_n} \quad (47.1.16)$$

The acceleration frequency response function (Accelerance) can be expressed as

$$\mathbf{FRF}_a(x_j, x_k, \omega) = -\omega^2 \sum_{n=1}^N \frac{\phi_n(x_k)}{(-\omega^2 + 2i\zeta_n\omega_n\omega + \omega_n^2)} \frac{\tilde{P}_n(x_j)}{M_n} \quad (47.1.17)$$

where $\tilde{P}_n(x_j)$ is obtained as

$$\tilde{P}_n(x_j) = \phi_n^T \tilde{p}(x_j) \quad (47.1.18)$$

and $\tilde{p}(x_j)$ is the space distribution of the harmonic force excitation (in the case of point force excitation, $\tilde{p}(x_j) = 1$ at node j in specified direction of excitation and 0 elsewhere).

47.1.2 About the damping

Damping can be given in several forms (see Keyword user manual in Appendix). A very common type of damping used in the nonlinear analysis of structure is to assume that the damping matrix is proportional to the mass and stiffness matrices, or

$$\mathbf{c} = \alpha \mathbf{m} + \beta \mathbf{k} \quad (47.1.19)$$

This type of damping is normally referred to as Rayleigh damping. For classically damped system,

$$2\omega_n\zeta_n = \phi_n^T \mathbf{c} \phi_n \quad (47.1.20)$$

Due to the orthogonality of the mass and stiffness matrices, it can be rewritten as

$$2\omega_n\zeta_n = \alpha + \beta\omega_n^2 \quad (47.1.21)$$

or,

$$\zeta_n = \frac{\alpha}{2\omega_n} + \frac{\omega_n}{2} \beta \quad (47.1.22)$$

47.2 ACOUSTIC FEM

A frequency domain acoustic finite element method has been implemented in LS-DYNA, to model the acoustic behavior of a confined acoustic fluid volume. This method is based on nodal velocity/pressure formulation. Three types of elements are available. They are hexahedron, pentahedron, and tetrahedron elements.

47.2.1 Theory basis

The governing equation for the acoustic problem is the Helmholtz equation.

$$\nabla^2 p + k^2 p = 0 \quad (47.2.1)$$

where p is the acoustic pressure; $k = \omega/c$ is called the wave number; $\omega = 2\pi f$ is the circular frequency of the acoustic wave; and c is the wave speed.

For vibro-acoustic problems, the boundary condition is given as follows,

$$\frac{\partial p}{\partial n} = -i\rho\omega v_n \quad (47.2.2)$$

where n is the normal vector pointing outside from the acoustic volume; $i = \sqrt{-1}$ is the imaginary unit; ρ is the acoustic fluid density and v_n is the normal velocity.

Using the weighted residue technique and taking the shape function N_i as the weighting function, the governing equation can be written as

$$\int_V \nabla^2 p N_i dV + \int_V k^2 p N_i dV = 0 \quad (47.2.3)$$

Using the Green's theorem, Equation (2.3) can be written as

$$-\int_V \nabla p \nabla N_i dV + k^2 \int_V p N_i dV = -\int_{\Gamma} \frac{\partial p}{\partial n} N_i d\Gamma \quad (47.2.4)$$

With the substitution of the boundary condition (47.2.2) into Equation (47.2.4), and taking the nodal pressure as the unknown variables, a linear equation system can be established and solved in frequency domain. Since there is only one variable on each node, this method is very fast.

48

Rotor Dynamics

48.1 Introduction

Rotor dynamics is a specialized branch of engineering science concerned with the behavior and diagnosis of rotating structures. It is a study of vibration of rotating parts found in a wide range of equipment including engine, turbine, aircraft, hard disk drive and more. The analysis of the rotator dynamics involves two coordinate systems: rotating and fixed coordinate systems. The equations of motion in the two coordinate systems are both introduced.

48.2 Two Coordinate systems

The interpretation of rotational phenomena requires the introduction of a rotating coordinate system in relation to the fixed coordinate system. Figure 1.1 depicts the relationship between the two coordinate systems. OXYZ is the fixed coordinate system and oxyz is the rotating coordinate system. \mathbf{R} is the location vector of the disk center; \mathbf{r} is the location vector of a point P with respect to the rotating coordinate system. Let the velocity of rotation be defined as Ω .

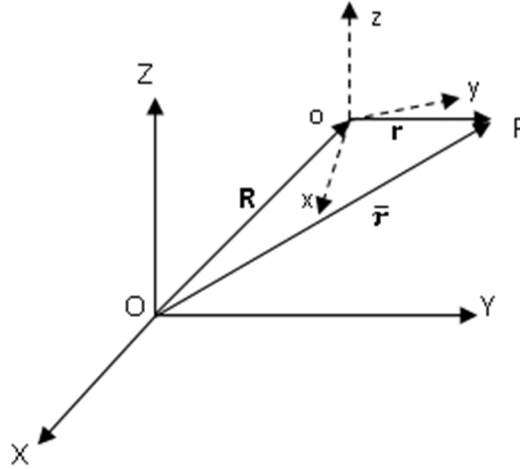


Figure 1.1 A rotating disk in two coordinate systems.

The position of the point P with respect to OXYZ is \bar{r} , so

$$\bar{r} = R + r. \quad (48.1)$$

The velocity of point P is:

$$\bar{v} = \frac{d\bar{r}}{dt} = \frac{dR}{dt} + \frac{dr}{dt} = V + v + \Omega \times r, \quad (48.2)$$

where $V (= \frac{dR}{dt})$ is the velocity of the origin o; v is the velocity of point P in the rotating coordinate system.

The acceleration of point P can be calculated as:

$$\bar{a} = \frac{d\bar{v}}{dt} = \frac{dV}{dt} + \frac{dv}{dt} + \frac{d(\Omega \times r)}{dt} = A + a + 2\Omega \times v + \Omega \times (\Omega \times r) + \frac{d\Omega}{dt} \times r, \quad (48.1)$$

where, A is the acceleration of the origin o, a is the acceleration of point P in rotating coordinate system.

We assume that the origin of the rotating coordinate system is fixed, so that:

$$V = A = 0. \quad (48.2)$$

By substituting (1.4) to (1.3):

$$\bar{a} = a + 2\Omega \times v + \Omega \times (\Omega \times r) + \frac{d\Omega}{dt} \times r. \quad (48.3)$$

48.3 Forces in the Rotating Coordinate System

We now place a particle with mass m into the position of the point P we were following. From (1.5), we can express the force of the particle in the rotating system as:

$$\mathbf{F} = m\mathbf{a} = m\bar{\mathbf{a}} - 2m\boldsymbol{\Omega} \times \mathbf{v} - m\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}) - m \frac{d\boldsymbol{\Omega}}{dt} \times \mathbf{r}. \quad (48.4)$$

The first term $m\bar{\mathbf{a}}$ is the force in the fixed coordinate system. All other terms on the right hand side are inertia forces arising in the rotating system. The Coriolis force is the following quantity:

$$\mathbf{F}_C = -2m\boldsymbol{\Omega} \times \mathbf{v}. \quad (48.5)$$

The third term produces the familiar centrifugal force:

$$\mathbf{F}_{Cf} = -m\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}). \quad (48.6)$$

The last term introduces the Euler fore when there is a nonzero rate of change in the magnitude of the rotation vector:

$$\mathbf{F}_E = -m \frac{d\boldsymbol{\Omega}}{dt} \times \mathbf{r}. \quad (48.7)$$

48.4 Transformation between Coordinate Systems

Let's assume that the rotation axis coincides with one of the axis of the rotating coordinate system, specifically to the z axis. An arbitrary rotation axis will be discussed later. In this case, the rotational velocity becomes:

$$\boldsymbol{\Omega} = 0 \bullet \mathbf{i} + 0 \bullet \mathbf{j} + \Omega \bullet \mathbf{k}. \quad (48.8)$$

We further restrict our computations to constant rotational velocity; hence the Euler force will not appear in the formulations (Euler force is easy to add to our equations if the rotational velocity is not constant though).

We write the location of a particle in the rotating coordinate system as:

$$\mathbf{r} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}. \quad (48.9)$$

It is transformed to the location in the fixed system as follows:

$$\bar{\mathbf{r}} = \begin{Bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{Bmatrix} = \begin{bmatrix} \cos\Omega t & -\sin\Omega t & 0 \\ \sin\Omega t & \cos\Omega t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \mathbf{H} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}. \quad (48.10)$$

\mathbf{H} is the transformation matrix. It is easy to get that $\mathbf{H}^T \mathbf{H} = \mathbf{I}$, where \mathbf{I} is the identity matrix. Other matrices that may be used later are also given here:

$$\dot{\mathbf{H}} = \Omega \begin{bmatrix} -\sin\Omega t & -\cos\Omega t & 0 \\ \cos\Omega t & -\sin\Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} = \Omega \bar{\mathbf{H}}, \quad (48.11)$$

$$\dot{H} = \Omega^2 \begin{bmatrix} -\cos\Omega t & \sin\Omega t & 0 \\ -\sin\Omega t & -\cos\Omega t & 0 \\ 0 & 0 & 0 \end{bmatrix} = \Omega^2 \bar{\bar{H}}, \quad (48.12)$$

$$\bar{H}^T H = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = P, \quad (48.13)$$

$$H^T \bar{H} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = P^T = -P, \quad (48.14)$$

$$\bar{H}^T \bar{H}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = J. \quad (48.15)$$

48.5 Equation of Motion in Rotating Coordinate System

When the particle undergoes a nodal translation in the rotating coordinate system as shown in Figure 1.2, it can be defined as:

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}. \quad (48.16)$$

The location vector in the fixed coordinate is:

$$\bar{\mathbf{r}} = H(\mathbf{r} + \mathbf{u}) \quad (48.17)$$

After calculate the velocity in the fixed coordinate system, we can get the kinetic energy due to translation displacement as:

$$T = \frac{m\Omega^2}{2} (\mathbf{r}^T J \mathbf{r} + 2\mathbf{r}^T J \mathbf{u} + \mathbf{u}^T J \mathbf{u}) + \frac{m}{2} (2\Omega \dot{\mathbf{u}}^T P^T \mathbf{r} + 2\Omega \mathbf{u}^T P \dot{\mathbf{u}} + \dot{\mathbf{u}}^T \dot{\mathbf{u}}). \quad (48.18)$$

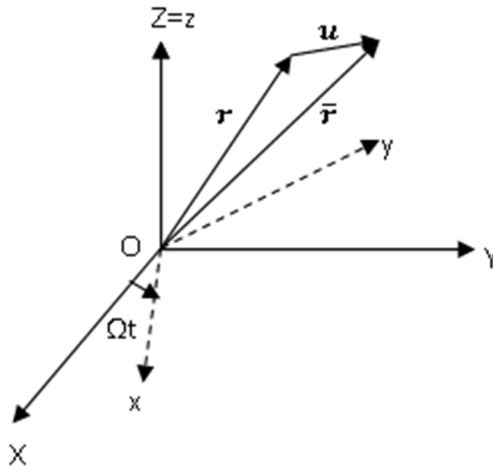


Figure 1.2 Translation nodal displacement.

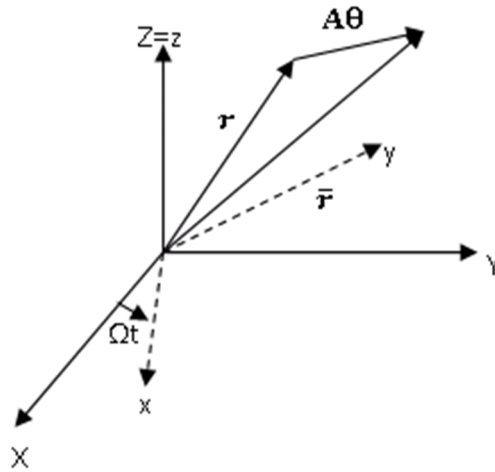


Figure 1.3 Rotational nodal displacement.

The nodal displacement of the point can also be rotational as shown in Figure 1.3. We assume small rotation of the rotating point:

$$\theta = \begin{Bmatrix} \varphi \\ \phi \\ \theta \end{Bmatrix}. \quad (48.19)$$

The nodal rotation requires the consideration of mass and inertia. The center of the mass point is coincident with the node. The inertia moment can be given with the concentrated mass input of commercial finite element codes, or they can be defined in connection with the surrounding mass point, or even with a simplified model by attaching six submasses to the node, as shown in figure 1.4.

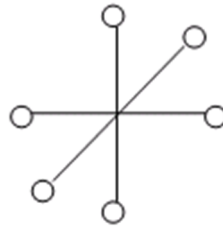


Figure 1.4 Node with six masses located at offset x' , y' and z' from the node center.

With this, the location vector in the fixed coordinate is of the form if only considers rotational displacement:

$$\bar{r} = H(r + u) = H(r + A\theta), \quad (48.20)$$

where

$$A = \begin{bmatrix} 0 & z' & -y' \\ -z' & 0 & x' \\ y' & -x' & 0 \end{bmatrix}. \quad (48.21)$$

Then we can get the kinetic energy due to rotational displacement as:

$$T = \frac{m\Omega^2}{2} (r^T J r + 2r^T J A \theta + \theta^T A^T J A \theta) + \frac{m}{2} (2\Omega r^T P A \dot{\theta} + 2\Omega \theta^T A^T P A \dot{\theta} + \dot{\theta}^T A^T A \dot{\theta}). \quad (48.22)$$

Applying Lagrange's equation, then we can obtain the final equation of motion as follows:

$$\begin{bmatrix} M_u & 0 \\ 0 & M_\theta \end{bmatrix} \begin{Bmatrix} \ddot{u} \\ \ddot{\theta} \end{Bmatrix} + 2\Omega \begin{bmatrix} C_u & 0 \\ 0 & C_\theta \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{\theta} \end{Bmatrix} - \Omega^2 \begin{bmatrix} Z_u & 0 \\ 0 & Z_\theta \end{bmatrix} \begin{Bmatrix} u \\ \theta \end{Bmatrix} = \begin{Bmatrix} F_{cu} \\ F_{c\theta} \end{Bmatrix}, \quad (48.23)$$

where M_u and M_θ are the mass and inertia matrices; C_u and C_θ are the gyroscopic matrices; Z_u and Z_θ are the centrifugal softening matrices; F_{cu} and $F_{c\theta}$ are centrifugal force. Note that we don't consider the other system damping and external force terms here, but they can be added to (1.25) accordingly.

48.6 Equation of Motion in Fixed Coordinate System

The location vector in the fixed coordinate is:

$$\bar{r} = H(r + u) = Hr + \bar{u}. \quad (48.24)$$

Here the \bar{u} represents the displacement of the point in the fixed coordinate system due to the nodal translation displacement. Similar analysis as in section 1.5 can be done for the equation of motion in fixed coordinate system. We only give the final equation here:

$$\begin{bmatrix} M_{\bar{u}} & 0 \\ 0 & M_{\bar{\theta}} \end{bmatrix} \begin{Bmatrix} \ddot{\bar{u}} \\ \ddot{\bar{\theta}} \end{Bmatrix} + \Omega \begin{bmatrix} 0 & 0 \\ 0 & C_{\bar{\theta}} \end{bmatrix} \begin{Bmatrix} \dot{\bar{u}} \\ \dot{\bar{\theta}} \end{Bmatrix} = \begin{Bmatrix} F_{c\bar{u}} \\ 0 \end{Bmatrix}, \quad (48.25)$$

where $M_{\bar{u}}$ and $M_{\bar{\theta}}$ are the mass and inertia matrices; $C_{\bar{\theta}}$ is the gyroscopic matrix; $F_{c\bar{u}}$ is the centrifugal force. All of them are written in the fixed coordinate system. Note that only nodal rotations contributed to the gyroscopic matrix. Same as before, we don't consider the other system damping and external force terms here, but they can be added to (1.27) accordingly.

48.7 Arbitrary Rotation Axis

All the above analysis is based on the assumption that the rotation axis is coincide with the z-axis. Here we will give a way to transform all variables back to global if the rotation axis is not coincide with the z-axis (Figure 1.5). The transformation matrix from z axis to rotation axis is denoted as T and it is easy to get $T^T T = I$.

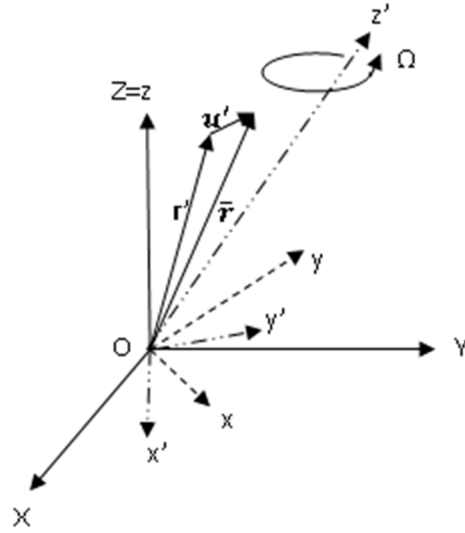


Figure 1.5 Rotation axis not coincide with z axis.

The location vector in the fixed coordinate becomes:

$$\bar{r} = T^T H(r' + u'). \quad (48.26)$$

where r' and u' are the location vector and displacement in the coordinate system $Ox'y'z'$, in which the z axis is transformed to the rotation axis z' by rotation matrix T . At the same time:

$$r' = Tr, \quad (48.27)$$

$$u' = Tu. \quad (48.28)$$

where, r and u are the location vector and displacement in the rotating coordinate system $Oxyz$.

The equation of motion in (1.25) and (1.27) can be simplified written as:

$$M_0 a + C_0 v + K_0 u = F_0, \quad (48.29)$$

And the force term can be written as

$$F_0 = f_0 r, \quad (48.30)$$

By substituting (1.28), (1.29) and (1.30) to the equation of motion, we can get new mass, damping, stiffness matrices and force vector as follows:

$$M = T^T M_0 T, \quad (48.31)$$

$$C = T^T C_0 T, \quad (48.32)$$

$$K = T^T K_0 T, \quad (48.33)$$

$$F = T^T f_0 Tr. \quad (48.34)$$

So the equation of motion becomes:

$$Ma + Cv + Ku = F. \quad (48.35)$$

After the equation of motion is obtained, it can then be solved using the implicit solver. Especially, the damping and stiffness matrices are related to the rotational velocity, so the eigen-frequencies might change with the change of rotational velocity. A diagram to represent this relationship is called Campbell diagram. An example is given in Figure 1.6.

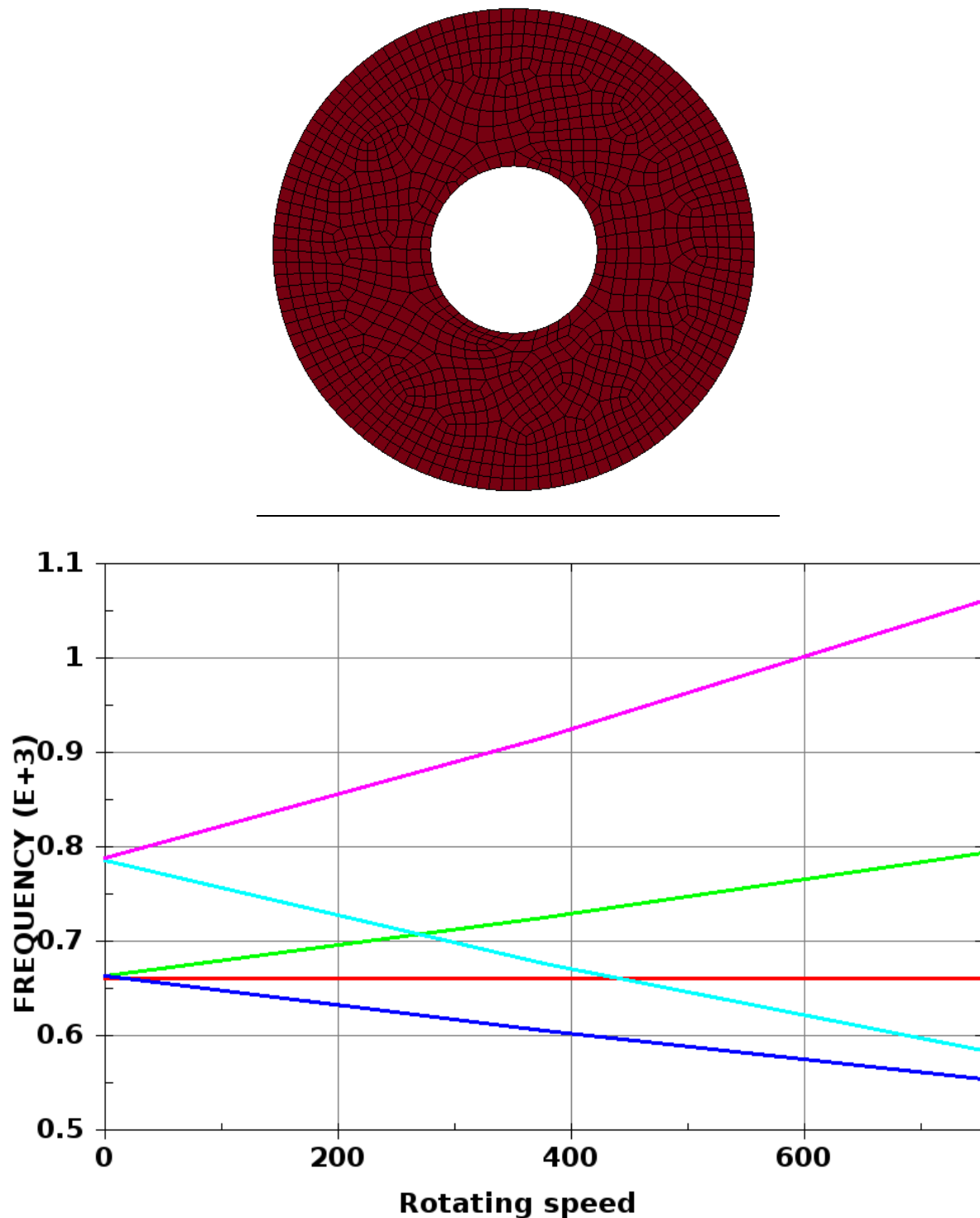


Figure 1.6 A disk is spinning with the center axis, the mode frequencies change with the increase of rotating speed.

49

References

Abbo A.J., S.W. Sloan, "A Smooth Hyperbolic Approximation to the Mohr-Coulomb Yield Criterion," *Computers and Structures*, Vol 54, No 1, 1995.

Addessio, F.L., D.E. Carroll, J.K. Dukowicz, F.H. Harlow, J.N. Johnson, B.A. Kashiwa, M.E. Maltrud, H.M. Ruppel, "CAVEAT: A Computer Code for Fluid Dynamics Problems with Large Distortion and Internal Slip," Report LA-10613-MS, UC-32, Los Alamos National Laboratory (1986).

Ahmad, S., Irons, B.M. and Zienkiewicz, O.C., "Analysis of Thick and Thin Shell Structures by Curved Finite Elements," *Int. J. Numer. Meths. Eng.*, **2** (1970).

Amsden, A. A., and Hirt, C. W., "YAQUI: An Arbitrary Lagrangian-Eulerian Computer Program for Fluid Flow at All Speeds," Los Alamos Scientific Laboratory, LA-5100 (1973).

Amsden, A., A., Ruppel, H. M., and Hirt, C. W., "SALE: A Simplified ALE Computer Program for Fluid Flow at All Speeds," Los Alamos Scientific Laboratory (1980).

Argyris, J.H., Kelsey, S., and Kamel, H., "Matrix Methods of Structural Analysis: A Precis of recent Developments," *Matrix Methods of Structural Analysis*, Pergamon Press (1964).

Arruda, E., and M. Boyce, "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials," published in the *Journal of the Mechanics and Physics of Solids*, Vol. **41**, No. 2, 389-412 (1993).

Auricchio, F., Taylor, R.L. and Lubliner J., "Shape-memory alloys: macromodelling and numerical simulations of the superelastic behavior", *Computer Methods in Applied Mechanics and Engineering* **146**, 281-312 (1997).

- Auricchio, F. and Taylor, R.L., "Shape-memory alloys: modeling and numerical simulations of the finite-strain superelastic behavior", *Computer Methods in Applied Mechanics and Engineering* **143**, 175-194 (1997).
- Back, S.Y., and Will, K.M., "A Shear-flexible Element with Warping for Thin-Walled Open Sections," *International Journal for Numerical Methods in Engineering*, 43, 1173-1191 (1998).
- Bahler AS: The series elastic element of mammalian skeletal muscle. *Am J Physiol* 213:1560-1564, 1967.
- Bammann, D. J., and E.C. Aifantis, "A Model for Finite-Deformation Plasticity," *Acta Mechanica*, **70**, 1-13 (1987).
- Bammann, D. J., "Modeling the Temperature and Strain Rate Dependent Large Deformation of Metals," *Proceedings of the 11th US National Congress of Applied Mechanics*, Tucson, AZ (1989).
- Bammann, D. J., and Johnson, G., "On the Kinematics of Finite-Deformation Plasticity," *Acta Mechanica* **69**, 97-117 (1987).
- Bammann, D.J., Chiesa, M.L., McDonald, A., Kawahara, W.A., Dike, J.J. and Revelli, V.D., "Predictions of Ductile Failure in Metal Structures," in *AMD-Vol. 107, Failure Criteria and Analysis in Dynamic Response*, edited by. H.E. Lindberg, 7-12 (1990).
- Bandak, F.A., private communications, U.S. Dept. of Trans., Division of Biomechanics Research, 400 7th St., S.W. Washington, DC 20590 (1991).
- Barlat and Lian, J., "Plastic Behavior and Stretchability of Sheet Metals, Part I: A Yield Function for Orthotropic Sheets Under Plane Stress Conditions," *International Journal of Plasticity*, **5**, 51-66 (1989).
- Barlat, F., Lege, D.J., and Brem, J.C., "A Six-Component Yield Function for Anisotropic Materials," *International Journal of Plasticity*, **7**, 693-712 (1991).
- Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall (1982).
- Bathe, K. J., and Wilson, E.L., *Numerical Methods in Finite Element Analysis*, Prentice-Hall (1976).
- Bathe, K.J., and Dvorkin, E.N., "A Continuum Mechanics Based Four Node Shell Element for General Nonlinear Analysis," *Int. J. Computer-Aided Eng. and Software*, Vol. **1**, 77-88 (1984).

- Bathe, K.-J. and Dvorkin, E.N. A four node plate bending element based on Mindlin-Reissner plate theory and a mixed interpolation, *Int. J. Num. Meth. Eng.*, **21**, 367-383, 1985.
- Batoz, J.L. and Ben Tahar, M. Evaluation of a new quadrilateral thin plate bending element, *Int. J. Num. Meth. Eng.*, **18**, 1644-1677, 1982.
- Battini, J., and Pacoste C., "Co-rotational Beam Elements with Warping Effects in Instability Problems," *Computational Methods in Applied Mechanical Engineering*, 191, 1755-1789, (2002).
- Bazeley, G.P., Cheung, W.K., Irons, B.M. and Zienkiewicz, O.C., "Triangular Elements in Plate Bending—Conforming and Nonconforming Solutions in Matrix Methods and Structural Mechanics," *Proc. Conf. on Matrix Methods in Structural Analysis*, Rept. AFFDL-R-66-80, Wright Patterson AFB, 547-576 (1965).
- Belytschko, T., "Transient Analysis," *Structural Mechanics Computer Programs*, edited by W. Pilkey, et. al., University Press of Virginia, 255-276 (1974).
- Belytschko, T., and Bindeman, L. P., "Assumed Strain Stabilization of the Eight Node Hexahedral Element," *Comp. Meth. Appl. Mech. Eng.* **105**, 225-260 (1993).
- Belytschko, T., and Hsieh, B.J., "Nonlinear Transient Finite Element Analysis with Convected Coordinates," *Int. J. Num. Meths. Engrg.*, **7**, 255–271 (1973).
- Belytschko, T., and Leviathan, I., "Projection schemes for one-point quadrature shell elements," *Comp. Meth. Appl. Mech. Eng.*, **115**, 277-286 (1994).
- Belytschko, T., and Lin, J., "A New Interaction Algorithm with Erosion for EPIC-3", Contract Report BRL-CR-540, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, Maryland (1985).
- Belytschko, T., Lin, J., and Tsay, C.S., "Explicit Algorithms for Nonlinear Dynamics of Shells," *Comp. Meth. Appl. Mech. Eng.* **42**, 225-251 (1984) [a].
- Belytschko, T., Ong, J. S.-J., Liu, W.K. and Kennedy, J.M., "Hourglass Control in Linear and Nonlinear Problems", *Comput. Meths. Appl. Mech. Engrg.*, **43**, 251–276 (1984b).
- Belytschko, T., Stolarski, H., and Carpenter, N., "A C0 Triangular Plate Element with One-Point Quadrature," *International Journal for Numerical Methods in Engineering*, **20**, 787-802 (1984) [b].

- Belytschko, T., Schwer, L., and Klein, M. J., "Large Displacement Transient Analysis of Space Frames," *International Journal for Numerical and Analytical Methods in Engineering*, **11**, 65-84 (1977).
- Belytschko, T., and Tsay, C.S., "Explicit Algorithms for Nonlinear Dynamics of Shells," *AMD*, **48**, ASME, 209-231 (1981).
- Belytschko, T., and Tsay, C.S., "WHAMSE: A Program for Three-Dimensional Nonlinear Structural Dynamics," Report EPRI NP-2250, Project 1065-3, Electric Power Research Institute, Palo Alto, CA (1982).
- Belytschko, T., and Tsay, C. S., "A Stabilization Procedure for the Quadrilateral Plate Element with One-Point Quadrature," *Int. J. Num. Method. Eng.* **19**, 405-419 (1983).
- Belytschko, T., Yen, H. R., and Mullen R., "Mixed Methods for Time Integration," *Computer Methods in Applied Mechanics and Engineering*, **17**, 259-175 (1979).
- Belytschko, T., "Partitioned and Adaptive Algorithms for Explicit Time Integration," in *Nonlinear Finite Element Analysis in Structural Mechanics*, ed. by Wunderlich, W. Stein, E, and Bathe, J. J., 572-584 (1980).
- Belytschko, T., Wong, B.L., and Chiang, H.Y., "Improvements in Low-Order Shell Elements for Explicit Transient Analysis," *Analytical and Computational Models of Shells*, A.K. Noor, T. Belytschko, and J. Simo, editors, ASME, CED, **3**, 383-398 (1989).
- Belytschko, T., Wong, B.L., and Chiang, H.Y., "Advances in One-Point Quadrature Shell Elements," *Comp. Meths. Appl. Mech. Eng.*, **96**, 93-107 (1992).
- Belytschko, T., Wong, B. L., Plaskacz, E. J., "Fission - Fusion Adaptivity in Finite Elements for Nonlinear Dynamics of Shells," **Computers and Structures**, Vol. 33, 1307-1323 (1989).
- Belytschko, T., Lu, Y.Y. and Gu, L., "Element-free Galerkin Methods," *Int. J. Numer. Methods Engrg.* **37**, 229-256 (1994).
- Benson, D.J., "Vectorizing the Right-Hand Side Assembly in an Explicit Finite Element Program," *Comp. Meths. Appl. Mech. Eng.*, **73**, 147-152 (1989).
- Benson, D. J., and Hallquist, J.O., "A Simple Rigid Body Algorithm for Structural Dynamics Program," *Int. J. Numer. Meth. Eng.*, **22** (1986).
- Benson, D.J., and Hallquist J.O., "A Single Surface Contact Algorithm for the Postbuckling Analysis of Shell Structures," *Comp. Meths. Appl. Mech. Eng.*, **78**, 141-163 (1990).

- Benson, D. J., "Momentum Advection on a Staggered Mesh," *Journal of Computational Physics*, **100**, No. 1, (1992).
- Benson, D. J., "Vectorization Techniques for Explicit Arbitrary Lagrangian Eulerian Calculations," *Computer Methods in Applied Mechanics and Engineering* (1992).
- Berstad, T., "Material Modelling of Aluminium for Crashworthiness Analysis", Dr.Ing. Dissertation, Department of Structural Engineering, Norwegian University of Science and Technology, Trondheim, Norway (1996).
- Berstad T., Hopperstad, O.S., and Langseth, M., "Elasto-Viscoplastic Constitutive Models in the Explicit Finite Element Code LS-DYNA," *Proceedings of the Second International LS-DYNA Conference*, San Francisco, CA (1994).
- Bischoff M. and Ramm E., "Shear deformable shell elements for large strains and rotations," *Int. J. Numer. Methods*, **40**, 4427-4449 (1997).
- Blatz, P.J., and Ko, W.L., "Application of Finite Element Theory to the Deformation of Rubbery Materials," *Trans. Soc. of Rheology*, **6**, 223-251 (1962).
- Bodig, Jozsef and Benjamin A. Jayne, *Mechanics of Wood and Wood Composites*, Krieger Publishing Company, Malabar, FL (1993).
- Boeing, Boeing Extreme Mathematical Library BCSLIB-EXT User's Guide, The Boeing Company, Document Number 20462-0520-R4 (2000).
- Borrvall, T., Development and implementation of material tangent stiffnesses for material model 76 in LS-DYNA, ERAB-02:46, Engineering Research Nordic AB, Linköping (2002).
- Borrvall, T., Revision of the implementation of material 36 for shell elements in LS-DYNA, ERAB Report E0307, Engineering Research Nordic AB, Linköping (2003).
- Björklund O., "Ductile Failure in High Strength Steel Sheets," *Linköping Studies in Science and Technology. Dissertations No. 1579*, ISSN 0345-7524 (2014).
- Brekelmans, W.A.M., Scheurs, P.J.G., and de Vree, J.H.P., "Continuum damage mechanics for softening of brittle materials", *Acta Mechanica*, **93**, 133-143, (1991).
- Brooks, A. N., and Hughes, T. J. R., "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering*, **32**, 199-259, (1982).

Burton, D.E., et. al., "Physics and Numerics of the TENSOR Code," Lawrence Livermore National Laboratory, Internal Document UCID-19428 (July 1982).

Cardoso, R.P.R. and Yoon J-W., "One point quadrature shell element with through-thickness stretch, Computer Methods in Applied Mechanics and Engineering," 194(9), 1161-1199 (2005).

Chang, F.K., and Chang, K.Y., "Post-Failure Analysis of Bolted Composite Joints in Tension or Shear-Out Mode Failure," J. of Composite Materials, **21**, 809-833 (1987).[a]

Chang, F.K., and Chang, K.Y., "A Progressive Damage Model for Laminated Composites Containing Stress Concentration," J. of Composite Materials, **21**, 834-855 (1987).[b]

Chang, F.S., "Constitutive Equation Development of Foam Materials," Ph.D. Dissertation, submitted to the Graduate School, Wayne State University, Detroit, Michigan (1995).

Chang, F.S., J.O. Hallquist, D. X. Lu, B. K. Shahidi, C. M. Kudelko, and J.P. Tekelly, "Finite Element Analysis of Low Density High-Hysteresis Foam Materials and the Application in the Automotive Industry," SAE Technical Paper 940908, in Safety Technology (SP-1041), International Congress and Exposition, Detroit, Michigan (1994).

Chen, J.S., Pan, C., Wu, C.T. and Liu, W.K., "Reproducing Kernel Particle Methods for Large Deformation Analysis of Nonlinear Structures," Comput. Methods Appl. Mech. Engrg. **139**, 195-227 (1996).

Chen, J.S. and Wu, C.T., "Generalized Nonlocal Meshfree Method in Strain Localization," *Proceeding of International Conference on Computational Engineering Science*, Atlanta, Georgia, 6-9 October (1998).

Chen, J.S. and Wang, H.P. "New Boundary Condition Treatments in Meshfree Computation of Contact Problems," Computer Methods in Applied Mechanics and Engineering, **187**, 441-468, (2001a).

Chen, J.S., Wu, C.T., Yoon, S. and You, Y. "A Stabilized Conforming Nodal Integration for Galerkin Meshfree Methods," Int. J. Numer. Methods Engrg. **50**, 435-466 (2001b).

Chen, W.F., and Baladi, G.Y., *Soil Plasticity: Theory and Implementation*, Elsevier, New York, (1985).

Christensen, R.M. "A Nonlinear Theory of Viscoelasticity for Application to Elastomers," *Journal of Applied Mechanics*, **47**, American Society of Mechanical Engineers, 762-768 (December 1980).

Chung, K., and K. Shah, "Finite Element Simulation of Sheet Metal Forming for Planar Anisotropic Metals," *Int. J. of Plasticity*, **8**, 453-476 (1992).

Cochran, S.G., and Chan, J., "Shock Initiation and Detonation Models in One and Two Dimensions," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18024 (1979).

Cohen, M., and Jennings, P.C., "Silent Boundary Methods for Transient Analysis", in *Computational Methods for Transient Analysis*, T. Belytschko and T.J.R. Hughes, editors, North-Holland, New York, 301-360 (1983).

Cook, R. D., *Concepts and Applications of Finite Element Analysis*, John Wiley and Sons, Inc. (1974).

Couch, R., Albright, E. and Alexander, "The JOY Computer Code," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19688 (1983).

Cray Research Inc., "CF77 Compiling System, Volume 4: Parallel Processing Guide," SG-3074 4.0, Mendota Heights, MN (1990).

Crisfield M.A., *Non-linear Finite Element Analysis of Solids and Structures, Volume 2, Advanced Topics*, John Wiley, New York (1997).

DeBar, R. B., "Fundamentals of the KRAKEN Code," Lawrence Livermore Laboratory, UCIR-760 (1974).

Deshpande, V.S. and N.A. Fleck, "Isotropic Models for Metallic Foams," *Journal of the Mechanics and Physics of Solids*, Vol. 48, 1253-1283, (2000).

Dobratz, B.M., "LLNL Explosives Handbook, Properties of Chemical Explosives and Explosive Simulants," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52997 (1981).

Duff, I. S., and Reid, J. K., "The Multifrontal Solution of Indefinite Sparse Symmetric Linear Equations," *ACM Transactions of Mathematical Software*, **9**, 302-325 (1983).

Dvorkin, E.N. and Bathe, K.J. "A continuum mechanics based four-node shell element for general nonlinear analysis," *International Journal for Computer-Aided Engineering and Software*, **1**, 77-88 (1984).

Englemann, B.E. and Whirley, R.G., "A New Explicit Shell Element Formulation for Impact Analysis," In *Computational Aspects of Contact Impact and Penetration*, Kulak, R.F., and Schwer, L.E., Editors, Elmeppress International, Lausanne, Switzerland, 51-90 (1991).

Englemann, B.E., Whirley, R.G., and Goudreau, G.L., "A Simple Shell Element Formulation for Large-Scale Elastoplastic Analysis," In Analytical and Computational Models of Shells, Noor, A.K., Belytschko, T., and Simo, J.C., Eds., CED-Vol. 3, ASME, New York, New York (1989).

Farhoomand, I., and Wilson, E.L., "A Nonlinear Finite Element Code for Analyzing the Blast Response of Underground Structures," U.S. Army Waterways Experiment Station, Contract Rept. N-70-1 (1970).

Feng, W.W., Private communication, Livermore, CA (1993).

Flanagan, D.P. and Belytschko, T., "A Uniform Strain Hexahedron and Quadrilateral and Orthogonal Hourglass Control," Int. J. Numer. Meths. Eng. **17**, 679-706 (1981)

Fleck, J.T., "Validation of the Crash Victim Simulator," **I - IV**, Report No. DOT-HS-806 279 (1981).

Fleischer M., Borrvall T., and Bletzinger K-U., "Experience from using recently implemented enhancements for Material 36 in LS-DYNA 971 performing a virtual tensile test", 6th European LS-DYNA Users Conference, Gothenburg, Sweden, 2007.

"Fundamental Study of Crack Initiation and Propagation," Author unknown, LLNL report, document received from LSTC (2003).

Galbraith, P.C., M.J. Finn, S.R. MacEwen, et.al., "Evaluation of an LS-DYNA3D Model for Deep-Drawing of Aluminum Sheet", FE-Simulation of 3-D Sheet Metal Forming Processes in Automotive Industry, VDI Berichte, **894**, 441-466 (1991).

Ghanem RG, Spanos PD. Stochastic Finite Elements – A Spectral Approach, Springer-Verlag, 1991, Revised Edition, Dover, 2003.

Ginsberg, M. and Johnson, J.P., "Benchmarking the Performance of Physical Impact Simulation Software on Vector and Parallel Computers," Proc. of the Supercomputing 88: Vol. II, Science and Applications, Computer Society Press (1988).

Gingold, R.A., and Monaghan, J.J., "Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars," Mon. Not. R. Astron. Soc. **181**, 375-389 (1977).

Ginsberg, M., and Katnik, R.B., "Improving Vectorization of a Crashworthiness Code," SAE Technical Paper No. 891985, Passenger Car Meeting and Exposition, Dearborn, MI (1989).

Giroux, E.D., "HEMP User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).

- Govindjee, S., Kay G.J., and Simo, J.C., "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," Report Number UCB/SEM M-94/18, University of California at Berkeley, Department of Civil Engineering (1994).
- Govindjee, S., Kay G.J., and Simo, J.C., "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," *International Journal for Numerical Methods in Engineering*, Volume 38, pages 3611-3633 (1995).
- Green, A.E. and Naghdi, P.M., "A General Theory of Elastic-Plastic Continuum," *Archive for Rational Mechanics and Analysis*, **18**, 251 (1965).
- Groenwold, A.A. and Stander, N. An efficient 4-node 24 d.o.f. thick shell finite element with 5-point quadrature. *Eng. Comput.* **12**(8), 723-747, 1995.
- Grimes, Roger, Lewis, John G., and Simon, Horst D., "A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems," *SIAM Journal of Matrix Analysis and Applications*, **15**, 228-272 (1994).
- Guccione, J., A. McCulloch, and L. Waldman, "Passive Material Properties of Intact Ventricular Myocardium Determined from a Cylindrical Model," published in the *ASME Journal of Biomechanical Engineering*, **113**, 42-55 (1991).
- Hallquist, J.O., "Preliminary User's Manuals for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimension)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-17268 (1976) and Rev. 1 (1979).[a]
- Hallquist, J.O., "A Procedure for the Solution of Finite Deformation Contact-Impact Problems by the Finite Element Method," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52066 (1976).
- Hallquist, J.O., "A Numerical Procedure for Three-Dimensional Impact Problems," American Society of Civil Engineering, Preprint 2956 (1977).
- Hallquist, J.O., "A Numerical Treatment of Sliding Interfaces and Impact," in: K.C. Park and D.K. Gartling (eds.) *Computational Techniques for Interface Problems*, AMD, **30**, ASME, New York (1978).
- Hallquist, J.O., "NIKE2D: An Implicit, Finite-Element Code for Analyzing the Static and Dynamic Response of Two-Dimensional Solids," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52678 (1979).[b]
- Hallquist, J.O. (1998), LS-DYNA Theoretical Manual, May (1998).

Hallquist, J.O., "User's Manual for DYNA2D— An Explicit Two-Dimensional Hydrodynamic Finite Element Code with Interactive Rezoning," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18756 (1980).

Hallquist, J.O., "User's Manual for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1981).[a]

Hallquist, J. O., "NIKE3D: An Implicit, Finite-Deformation, Finite-Element Code for Analyzing the Static and Dynamic Response of Three-Dimensional Solids," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18822 (1981).[b]

Hallquist, J.O., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1982; Rev. 1: 1984; Rev. 2: 1986).

Hallquist, J.O., "Theoretical Manual for DYNA3D," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19401 (March, 1983).

Hallquist, J.O., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1988, Rev. 4).

Hallquist, J.O., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," Livermore Software Technology Corporation, Rept. 1007 (1990).

Hallquist, J.O., LS-DYNA Keyword User's Manual, version 970, Livermore Software Technology Corporation (April, 2003).

Hallquist, J.O., Benson, D.J., and Goudreau, G.L., "Implementation of a Modified Hughes-Liu Shell into a Fully Vectorized Explicit Finite Element Code," Proceedings of the International Symposium on Finite Element Methods for Nonlinear Problems, University of Trondheim, Trondheim, Norway (1985).

Hallquist, J.O., and Benson, D.J., "A Comparison of an Implicit and Explicit Implementation of the Hughes-Liu Shell," Finite Element Methods for Plate and Shell Structures, T.J.R. Hughes and E. Hinton, Editors, 394-431, Pineridge Press Int., Swansea, U.K. (1986).

Hallquist, J.O., and Benson, D.J., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1986, Rev. 2).

Hallquist, J.O. and Benson, D.J., "DYNA3D User's Manual (Nonlinear Dynamic Analysis of Solids in Three Dimensions)," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19156 (1987, Rev. 3).

Hallquist, J.O., Stillman, D.W., Hughes, T.J.R., and Tarver, C., "Modeling of Airbags Using MVMA/DYNA3D," LSTC Report (1990).

Harten, A., "ENO Schemes with Subcell Resolution," *J. of Computational Physics*, **83**, 148-184, (1989).

Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, **47**, 329 (1980).

Herrmann, L.R. and Peterson, F.E., "A Numerical Procedure for Viscoelastic Stress Analysis," Seventh Meeting of ICRPG Mechanical Behavior Working Group, Orlando, FL, CPIA Publication No. 177 (1968).

Hill A.V., "The heat of shortening and the dynamic constants of muscle," *Proc Roy Soc B126*:136-195, (1938).

Hill, R., "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," *Proceedings of the Royal Society of London, Series A.*, **193**, 281 (1948).

Hill, R. "Constitutive Modeling of Orthotropic Plasticity in Sheet Metals," *Journal of Mechanics and Physics of Solids*, **38**, Number 3, 405-417 (1999).

Holmquist, T.J., G.R. Johnson, and W.H. Cook, "A Computational Constitutive Model for Concrete Subjected to Large Strains, High Strain Rates, and High Pressures, pp. 591-600, (1993).

Holtz R.D., W. D. Kovacs, "An Introduction to Geotechnical Engineering", Prentice Hall, Inc. New Jersey, 1981

Hopperstad, O.S. and Remseth, S., "A Return Mapping Algorithm for a Class of Cyclic Plasticity Models," *International Journal for Numerical Methods in Engineering*, **38**, 549-564 (1995).

Hughes, T.J.R., *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall Inc., Englewood Cliffs, NJ (1987).

Hughes, T.J.R., "Generalization of Selective Integration Procedures to Anisotropic and Nonlinear Media," *Int. J. Numer. Meth. Eng.* **15**, 9 (1980).

- Hughes, T.J.R. and Carnoy, E., "Nonlinear Finite Element Shell Formulation Accounting for Large Membrane Strains," AMD-Vol.48, ASME, 193-208 (1981).
- Hughes, T.J.R. and Liu, W.K., "Nonlinear Finite Element Analysis of Shells: Part I. Two-Dimensional Shells." *Comp. Meths. Appl. Mechs.* **27**, 167-181 (1981).
- Hughes, T.J.R. and Liu, W.K., "Nonlinear Finite Element Analysis of Shells: Part II. Three-Dimensional Shells." *Comp. Meths. Appl. Mechs.* **27**, 331-362 (1981).
- Hughes, T.J.R., Liu, W.K., and Levit, I., "Nonlinear Dynamics Finite Element Analysis of Shells." *Nonlinear Finite Element Analysis in Struct. Mech.*, Eds. W. Wunderlich, E. Stein, and K.J. Bathe, Springer-Verlag, Berlin, 151-168 (1981).
- Hughes, T.J.R., Taylor, R. L., Sackman, J. L., Curnier, A.C., and Kanoknukulchai, W., "A Finite Element Method for a Class of Contact-Impact Problems," *J. Comp. Meths. Appl. Mechs. Eng.* **8**, 249-276 (1976).
- Hughes, T.J.R., and Winget, J., "Finite Rotation Effects in Numerical Integration of Rate Constitutive Equations Arising in Large-Deformation Analysis," *Int. J. Numer. Meths. Eng.*, **15**, 1862-1867, (1980).
- Hulbert, G.M. and Hughes, T.J.R., "Numerical Evaluation and Comparison of Subcycling Algorithms for Structural Dynamics," Technical Report, DNA-TR-88-8, Defense Nuclear Agency, Washington, DC (1988).
- Ibrahimbegovic, A. and Wilson, E.L. A unified formulation for triangular and quadrilateral flat shell finite elements with six nodal degrees of freedom, *Comm. Applied Num. Meth*, **7**, 1-9, 1991.
- Isenberg, J., D.K. Vaughn, and I. Sandler, "Nonlinear Soil-Structure Interaction," EPRI Report MP-945, Weidlinger Associates, December (1978).
- Jabareen, M., and Rubin, M.B., A Generalized Cosserat Point Element (CPE) for Isotropic Nonlinear Elastic Materials including Irregular 3-D Brick and Thin Structures, *J. Mech. Mat. And Struct.*, Vol 3-8, 1465-1498 (2008).
- Jabareen, M., Hanukah, E. and Rubin, M.B., A Ten Node Tetrahedral Cosserat Point Element (CPE) for Nonlinear Isotropic Elastic Materials, *J. Comput. Mech.* **52**, 257-285 (2013).
- Jaumann, G. "Geschlossenes System Physikalischer und Chemischer Differentialdesefz, Sitz Zer. Akad. Wiss. Wein, (IIa), **120**, 385 (1911).

Johnson, G.C. and Bammann D.J., "A Discussion of Stress Rates in Finite Deformation Problems" *International Journal of Solids and Structures*, **20**, 725-737 (1984).

Johnson, G.R. and Cook, W. H., "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures," presented at the Seventh International Symposium on Ballistics, The Hague, The Netherlands, April (1983).

Johnson, G.R. and T.J. Holmquist, "An Improved Computational Model for Brittle Materials" in High-Pressure Science and Technology - 1993 American Institute of Physics Conference Proceedings 309 (c 1994) pp.981-984 ISBN 1-56396-219-5.

Johnson, C., Navert, U., and Pitkaranta, J., "Finite Element Methods for Linear Hyperbolic Problems," *Computer Methods in Applied Mechanics and Engineering*, **45**, 285-312, (1984).

Jones, N., "Structural Aspects of Ship Collisions," Chapter 11, in *Structural Crashworthiness*, Eds. N. Jones and T Wierzbicki, Butterworths, London, 308-337 (1983).

Jones, R.M., *Mechanics of Composite Materials*, Hemisphere Publishing Co., New York (1975).

Ju J.W., "Energy-Based Coupled Elastoplastic Damage Models at Finite Strains", *J. of Eng Mech.*, Vol 115, No 11, Nov 1989.

Karypis, G., and Kumar V., "METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices," Department of Computer Science, University of Minnesota (1998).

Katz, J., and Maskew, B., "Unsteady Low-Speed Aerodynamic Model for Complete Aircraft Configurations," *Journal of Aircraft* **25**, 4 (1988).

Kenchington, G. J., "A Non-Linear Elastic Material Model for DYNA3D," Proceedings of the DYNA3D Users Group Conference, September 1988, published by Boeing Computer Services (Europe) Limited.

Kennedy, J. M., Belytschko, T., and Lin, J. I., "Recent Developments in Explicit Finite Element Techniques and their Applications to Reactor Structures," *Nuclear Engineering and Design* **97**, 1-24 (1986).

Kim, N., Seo, K., and Kim, M., "Free Vibration and Spatial Stability of Non-symmetric Thin-Walled Curved Beams with Variable Curvatures," *International Journal of Solids and Structures*, **40**, 3107-3128, (2003).

- Klisinski M., "Degradation and Plastic Deformation of Concrete", Ph.D. thesis, Polish Academy of Sciences, 1985, IFTR report 38.
- Kosloff, D. and Frazier, G.A., "Treatment of Hourglass Patterns in Low Order Finite Element Codes," Int. J. Numer. Anal. Meth. Geomech. **2**, 57-72 (1978)
- Kretschmann, D. E., and David W. Green, "Modeling Moisture Content-Mechanical Property Relationships for Clear Southern Pine," Wood and Fiber Science, 28(3), pp. 320-337 (1996).
- Kreyszig, E., Advanced Engineering Mathematics, John Wiley and Sons, New York, New York (1972).
- Krieg, R.D., "A Simple Constitutive Description for Cellular Concrete," Sandia National Laboratories, Albuquerque, NM, Rept. SC-DR-72-0883 (1972).
- Krieg, R.D. and Key, S.W., Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs, Vol. 20 of Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects (American Society of Mechanical Engineers, New York, N.Y., 1976), 125-137.
- Kumar, P., Nukala, V. V., and White, D. W., "A Mixed Finite Element for Three-dimensional Nonlinear Analysis of Steel Frames," *Computational Methods in Applied Mechanical Engineering*, 193, 2507-2545, (2004).
- Lancaster, P. and Salkauskas, K., "Surfaces generated by moving least squares methods", Math. Comput. **37**, 141-158, (1981).
- Landshoff, R., "A Numerical Method for Treating Fluid Flow in the Presence of Shocks," Los Alamos Scientific Laboratory, Rept. LA-1930 (1955).
- Lee, E.L. and Tarver, C.M., "Phenomenological Model of Shock Initiation in Heterogeneous Explosives," Phys. of Fluids, **23**, 2362 (1980).
- Lemaitre, J., A Course on Damage Mechanics. Springer-Verlag, (1992).
- Lemmen, P.P.M and Meijer, G.J., "Failure Prediction Tool Theory and User Manual," TNO Building and Construction Research, The Netherlands, Rept. 2000-CMC-R0018, (2001).

Lewis B.A., "Developing and Implementing a Road Side Safety Soil Model into LS-DYNA", FHWA Research and Development Turner-Fairbank Highway Research Center. Oct, 1999.

Librescu, L. Qin, Z., and Ambur D.R., "Implications of Warping Restraint on Statics and Dynamics of Elastically Tailored Thin-Walled Composite Beams," *International Journal of Mechanical Sciences*, 45, 1247-1267 (2003).

Liu, Joseph W. H., "Modification of the Minimum-Degree Algorithm by Multiple Elimination," *ACM Transactions on Mathematical Software*, 11, 141-153 (1985).

Liu, W.K. and Belytschko, T., "Efficient Linear and Nonlinear Heat Conduction with a Quadrilateral Element," *Int. J. Num. Meths. Engrg.*, 20, 931-948 (1984).

Liu, W.K., Belytschko, T., Ong, J.S.J. and Law, E., "Use of Stabilization Matrices in Nonlinear Finite Element Analysis," *Engineering Computations*, 2, 47-55 (1985).

Liu, W.K., *Lecture Notes on Nonlinear Finite Element Methods*, Northwestern University, (1983, revised in 1992).

Liu, W.K., Guo, Y., Tang, S. and Belytschko T., "A Multiple-Quadrature Eight-node Hexahedral Finite Element for Large Deformation Elastoplastic Analysis," *Comput. Meths. Appl. Mech. Engrg.*, 154, 69-132 (1998).

Liu, W.K., Hu, Y.K., and Belytschko, T., "Multiple Quadrature Underintegrated Finite Elements," *Int. J. Num. Meths. Engrg.*, 37, 3263-3289 (1994).

Liu, W. K., Chang, H., and Belytschko, T., "Arbitrary Lagrangian-Eulerian Petrov-Galerkin Finite Elements for Nonlinear Continua," *Computer Methods in Applied Mechanics and Engineering*, to be published.

Liu, W.K., Jun, S. and Zhang, Y.F., "Reproducing Kernel Particle Method," *Int. J. Numer. Methods Fluids* 20, 1081-1106 (1995).

Lucy, L.B., "Numerical Approach to Testing the Fission Hypothesis," *Astron. J.* 82, 1013-1024 (1977).

Lysmer, J. and Kuhlemeyer, R.L., "Finite Dynamic Model for Infinite Media", *J. Eng. Mech. Div. ASCE*, 859-877 (1969).

MacNeal R.H. and Harder R.L., "A Proposed Standard Set of Problems to Test Finite Element Accuracy," *Finite Elements Anal. Des.*, (1985, 3-20 (1985).

- Maenchen, G. and Sack, S., "The Tensor Code," Meth. Comp. Phys. **3**, (Academic Press), 181-263 (1964).
- Maffeo, R., "TRANCITS Program User's Manual," General Electric Company, Cincinnati, OH, NASA Report CR-174891, (1985).
- Maffeo, R., "Burner Liner Thermal/Structural Load Modeling," General Electric Company, Cincinnati, OH, NASA Report CR-174892, (1984).
- Maker, B. N., Private communication, Lawrence Livermore National Laboratory. Dr. Maker programmed and implemented the compressible Mooney-Rivlin rubber model.
- Marchertas, A. H., and Belytschko, T. B., "Nonlinear Finite Element Formulation for Transient Analysis of Three Dimensional Thin Structures," Report ANL-8104, LMFBR Safety, Argonne National Laboratory, Argonne, IL (1974).
- Margolin, L. G., personal communication to D. Benson (1989).
- Maskew, B., "Program VSAERO Theory Document," NASA CR-4023 (1987).
- Matthies, H., and G. Strang, The Solution of Nonlinear Finite Element Equations, Int. Journal for Numerical Methods in Engineering, **14**, No. 11, 1613-1626.
- Matzenmiller, A., and Schweizerhof, K., "Crashworthiness Simulations of Composite Structures - A First Step with Explicit Time Integration," Nonlinear Computational mechanics - A State of the Art, edited by P.W. Wriggers, et. al., Springer-Verlag, (1991).
- Mauldin, P.J., R.F. Davidson, and R.J. Henninger, "Implementation and Assessment of the Mechanical-Threshold-Stress Model Using the EPIC2 and PINON Computer Codes," Report LA-11895-MS, Los Alamos National Laboratory (1990).
- McGlaun, personal communication to D. Benson, Sandia National Laboratories, (1990).
- McGlaun, J. M., Thompson, S. L., and Elrick, M. G., "CTH: A Three-Dimensional Shock Wave Physics Code," Proc. of the 1989 Hypervelocity Impact Symposium (1989).
- Mindlin, R.D., "Influence of Rotary Inertia and Shear on Flexural Motions of Isotropic, Elastic Plates," J. Appl. Mech. **18**, 31-38 (1951).
- Mizukami, A., and Hughes, T. J. R., "A Petrov-Galerkin Finite Element Method for Convection Dominated Flows: An Accurate Upwinding Technique for Satisfying the Maximum Principle," Computer Methods in Applied Mechanics and Engineering, Vol. 50, pp. 181-193 (1985).

- Monaghan, J.J., and Gingold, R.A., "Shock Simulation by the Particle Method of SPH," *Journal of Computational Physics*, **52**, 374-381 (1983).
- Murray Y.D., "Modeling Rate Effects in Rock and Concrete", Proceedings of the 8th International Symposium on Interaction of the Effects of Munitions with Structures, Defense Special Weapons Agency, McLean VA, USA, (1997).
- Murray, Y. D., "Modeling Rate Effects in Rock and Concrete," Proceedings of the 8th International Symposium on Interaction of the Effects of Munitions with Structures, Defense Special Weapons Agency, McLean, VA, USA, (1997).
- Murray, Y.D., Users Manual for Transversely Isotropic Wood Model, aptek technical report to fhwa (to be published), (2002).
- Murray, Y.D., and J. Reid, Evaluation of Wood Model for Roadside Safety Applications, aptek technical report to fhwa (to be published), (2002).
- Nagtegaal, J.C., Parks, D.M., and Rice J.R., "On Numerically Accurate Finite Element Solution in the Fully Plastic Range", *Computer Methods in Applied Mechanics and Engineering*, **4**, 153 (1974).
- Newman, W.M., and Sproull, R.F., *Principles of Interactive Computer Graphics*, McGraw-Hill, New York (1979).
- Newmark, N., "A Method of Computation for Structural Dynamics," *Journal of the Engineering Mechanics Division*, Proceedings of the American Society of Civil Engineers, 67-94 (1959).
- Neilsen, M.K., Morgan, H.S., and Krieg, R.D., "A Phenomenological Constitutive Model for Low Density Polyurethane Foams," Rept. SAND86-2927, Sandia National Laboratories, Albuquerque, NM (1987).
- Noh, W.F., *Meth. Comp. Phys.* **3**, (Academic Press) (1964).
- Noh, W.F., "Numerical Methods in Hydrodynamic Calculations," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52112 (1976).
- Oden, J.T, P. Devloo, and T. Strouboulis, "Adaptive Finite Element Methods for the Analysis of Inviscid Compressible Flow: Part I. Fast Refinement/Unrefinement and Moving Mesh Methods for Unstructured Meshes," *Computer Methods in Applied Mechanics and Engineering*, **59**, 327-362 (1986).
- Ogden, R.W., *Non-Linear Elastic Deformations*, Ellis Horwood Ltd., Chichester, Great Britain (1984).

- Okuda T., Yamamae Y. and Yasuki T., Request for MAT126 Modification, Microsoft Power Point presentation, Toyota Communication Systems and Toyota Motor Corporation, 2003.
- Oliver, J., "A Consistent Characteristic Length of Smeared Cracking Models," *International Journal for Numerical Methods in Engineering*, **28**, 461-474 (1989).
- Rajendran, A.M., "Material Failure Models", in Material Behavior at High Strain Rates, Course Notes, Computational Mechanics Associates, Monterey, CA, 424 (1989).
- Papadrakakis, M., "A Method for the Automated Evaluation of the Dynamic Relaxation Parameters," *Comp. Meth. Appl. Mech. Eng.* **25**, 35-48 (1981).
- Phoon KK, Huang SP, Quek ST. Implementation of Karhunen–Loeve expansion for simulation using a wavelet-Galerkin scheme. *Probabilist Eng Mech* 2002;17(3):293–303. (2002a)
- Phoon KK, Huang SP, Quek ST. Simulation of second-order processes using Karhunen–Loeve expansion. *Comput Struct* 2002; 80(12):1049–60. (2002b)
- Phoon KK, Huang HW, Quek ST. Simulation of strongly non-Gaussian processes using Karhunen–Loeve expansion, *Probabilistic Engineering Mechanics* 20 (2005) 188–198.
- Pian, T.H.H., and Sumihara, K., "Rational Approach for Assumed Stress Elements," *Int. J. Num. Meth. Eng.*, **20**, 1685-1695 (1985).
- Prokic, A., "New Warping Function for Thin-Walled Beams. I: Theory," *Journal of Structural Engineering*, 122, 1437-1442, (1994).
- Przemieniecki, J. S., *Theory of Matrix Structural Analysis*, McGraw-Hill Book Company, New York, NY (1986).
- Reid, S.R. and C. Peng, "Dynamic Uniaxial Crushing of Wood," *Int. J. Impact Engineering*, Vol. 19, No. 5-6, pp. 531-570, (1997).
- Reyes, A., O.S. Hopperstad, T. Berstad, and M. Langseth, "Implementation of a Material Model for Aluminium Foam in LS-DYNA", Report R-01-02, Restricted, Department of Structural Engineering, Norwegian University of Science and Technology, (2002).
- Richards, G.T., "Derivation of a Generalized von Neumann Pseudo-Viscosity with Directional Properties," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-14244 (1965).

Richtmyer, R.D., and Morton, K.W., *Difference Equations for Initial-Value Problems*, Interscience Publishers, New York (1967).

Sandler, I.S., and D. Rubin, "An Algorithm and a modular subroutine for the cap model," *Int. J. Numer. Anal. Meth. Geomech.*, **3**, 173-186 (1979).

Schwer, L.E., Cheva, W., and Hallquist, J.O., "A Simple Viscoelastic Model for Energy Absorbers Used in Vehicle-Barrier Impacts," In *Computational Aspects of Contact Impact and Penetration*, Kulak, R.F., and Schwer, L.E., Editors, Elmepress International, Lausanne, Switzerland, 99-117 (1991).

Shapiro, A.B., "TOPAZ3D - A Three Dimensional Finite Element Heat Transfer Code," University of California, Lawrence Livermore National Laboratory, Report UCID-20481 (1985).

Shapiro, A.B., "REMAP: a Computer Code That Transfers Node Information Between Dissimilar Grids," Lawrence Livermore National Laboratory, UCRL-ID-104090, (1990).

Shkolnikov, M.B. Private Communication (1990-1991).

Simo, J.C. and Govindjee, S., "Exact Closed-Form Solution of the Return Mapping Algorithm in Plane Stress Elasto-Viscoplasticity," *Engineering Computations*, **5**, 254-258 (1988).

Simo, J.C. and Hughes, T.J.R. "On the variational foundations of assumed strain methods," *Journal of Applied Mechanics*, **53**, 1685-1695 (1986).

Simo, J.C., Ju, J.W., "Stress and Strain Based Continuum Damage Models", Parts I & II, *Int. J. of Solids and Structures*, Vol 23, No 7, (1987).

Simo, J.C., Ju, J.W., Pister, K.S., and Taylor, R.L. "An Assessment of the Cap Model: Consistent Return Algorithms and Rate-Dependent Extension," *J. Eng. Mech.*, **114**, No. 2, 191-218 (1988a).

Simo, J.C., Ju, J.W., Pister, K.S., and Taylor, R.L. "Softening Response, Completeness Condition, and Numerical Algorithms for the Cap Model," *Int. J. Numer. Anal. Meth. Eng.* (1990).

Simo, J.C. and Taylor, R.L., "A Return Mapping Algorithm for Plane Stress Elastoplasticity," *International Journal for Numerical Methods in Engineering*, **22**, 649-670 (1986).

Steinberg, D.J. and Guinan, M.W., "A High-Strain-Rate Constitutive Model for Metals," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-80465 (1978).

Stillman, D. W., and Hallquist, J. O., "LS-INGRID: A Pre-Processor and Three-Dimensional Mesh Generator for the Programs LS-DYNA, LS-NIKE3D, and TOPAZ3D," Version 3.1, Livermore Software Technology Corporation (1992).

Stojko, S., privated communication, NNC Limited, Engineering Development Center (1990).

Stolarski, H., and Belytschko, T., "Shear and Membrane Locking in Curved Elements," *Comput. Meths. Appl. Mech. Engrg.*, **41**, 279–296 (1983).

Stone, C.M., Krieg, R.D., and Beisinger, Z.E., "Sancho, A Finite Element Computer Program for the Quasistatic, Large Deformation, Inelastic Response of Two-Dimensional Solids," Sandia Report, SAND 84-2618, UC-32, Albuquerque, NM (1985).

Storakers, B., "On Material Representation and Constitutive Branching in Finite Compressible Elasticity", Royal Institute of Technology, Stockholm Sweden, (1985).

Sturt, R.M.V., and B.D. Walker, J.C. Miles, A. Giles, and N. Grew, "Modelling the Occupant in a Vehicle Context-An Integrated Approach," 13th International ESV Conference, Paris, November 4-7 (1991).

Sussman, T. and Bathe, K.J., "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *Computers & Structures*, **26**, Number 1/2, 357-409 (1987).

Tarver, C.M., and Hallquist, J.O., "Modeling of Two-Dimensional Shock Initiation and Detonation Wave Phenomena in PBX 9404 and LX-17," University of California, Lawrence Livermore National Laboratory, Rept. UCID84990 (1981).

Taylor, R.L. and Simo, J.C. Bending and membrane elements for the analysis of thick and thin shells, *Proc. of NUMETA Conference*, Swansea, 1985.

Taylor, R.L., Finite element anlysis of linear shell problems, in Whiteman, J.R. (ed.), *Proc. of the Mathematics in Finite Elements and Applications*, Academic Press, New York, 191-203 (1987).

Taylor, L.M., and Flanagan, D.P., "PRONTO3D A Three-Dimensional Transient Solid Dynamics Program," Sandia Report: SAND87-1912, UC-32 (1989).

Thompson, S. L., "CSQ -- A Two Dimensional Hydrodynamic Program with Energy Flow and Material Strength," Sandia Laboratories, SAND74-0122 (1975).

- Thompson, R., L., and Maffeo, R. L., "A Computer Analysis Program for Interfacing Thermal and Structural Codes," NASA Lewis Research Center, Report NASA-TM-87021 (1985).
- Trefethen, L.N., "Group Velocity in Finite Difference Schemes," SIAM Review, **24**, No. 2 (1982).
- Tsai, S.W., and Wu, E.M., "A General Theory of Strength for Anisotropic Materials," Journal of Composite Materials, 58-80 (1971).
- Tuler, F.R. and B.M. Butcher, "A Criterion for the Time Dependence of Dynamic Fracture," *The International Journal of Fracture Mechanics*, **4**, No. 4 (1968).
- Turk M, Pentland A. Eigenfaces for Recognition, Journal of Cognitive Neuroscience, 1991; 3(1):71-86.
- Turkalj, G. Brnic, J., and Prpic-Orsic J., "Large Rotation Analysis of Elastic Thin-Walled Beam-Type Structures Using ESA Approach," *Computers & Structures*, **81**, 1851-1864, (2003).
- Underwood, P., "Dynamic Relaxation," Computational Method for Transient Analysis, Belytschko, T., and Hughes, T.J.R., Eds., **1**, 245-263, (1986).
- Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection," Journal of Computational Physics, **23**, 276-299 (1977).
- Vawter, D., "A Finite Element Model for Macroscopic Deformation of the Lung," published in the Journal of Biomechanical Engineering, **102**, 1-7, (1980).
- von Neumann, J., and Richtmyer, R.D., "A Method for the Numerical Calculation of Hydrodynamical Shocks," J. Appl. Phys., **21**, 232 (1950).
- Walker, B.D., and P.R.B. Dallard, "An integrated Approach to Vehicle Crashworthiness and Occupant Protection Systems", SAE International Congress and Exposition, Detroit, Michigan, (910148), February 25-March 1 (1991).
- Walker, H.F., Numerical Solution of Nonlinear Equations, University of California, Lawrence Livermore National Laboratory, Rept. UCID-18285 (1979).
- Wang, J. T., and O. J. Nefske, "A New CAL3D Airbag Inflation Model," SAE paper 880654, (1988).
- Wang, J. T., private communication (1992).

- Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", Publication R&D 8332, General Motors Development Center, Warren, MI (1995).
- Wang, J.T., "An Analytical Model for an Airbag with a Hybrid Inflator", AMD-Vol. 210, BED 30, ASME, 467-497 (1995).
- Warsi, Z. U. A., "Basic Differential Models for Coordinate Generation," in Symposium on the Numerical Generation of Curvilinear Coordinate Systems, Nashville, Tenn. (1982).
- Whirley, R. G., Hallquist, J. O., and Goudreau, G. L., "An Assessment of Numerical Algorithms for Plane Stress and Shell Elastoplasticity on Supercomputers," *Engineering Computations*, **6**, 116-126 (June, 1989).
- Wilkins, M.L., "Calculations of Elastic Plastic Flow," *Meth. Comp. Phys.*, **3**, (Academic Press), 211-263 (1964).
- Wilkins, M. L., "Calculation of Elastic-Plastic Flow," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-7322, Rev. I (1969).
- Wilkins, M.L., "Use of Artificial Viscosity in Multidimensional Fluid Dynamic Calculations," *J. Comp. Phys.* **36**, 281 (1980).
- Wilkins, M.L., Blum, R.E., Cronshagen, E., and Grantham, P., "A Method for Computer Simulation of Problems in Solid Mechanics and Gas Dynamics in Three Dimensions and Time," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51574 (1974).
- Wilson, E.L., Three Dimensional Static and Dynamic Analysis of Structures, A publication of Computers and Structures, Inc., Berkeley, California, [1996-2000].
- Winslow, A. M., "Equipotential Zoning of Two-Dimensional Meshes," Lawrence Radiation Laboratory, UCRL-7312 (1963).
- Winslow, A. M., "Equipotential Zoning of The Interior of a Three-Dimensional Mesh," Report to LSTC (1990).
- Winters, J.M., "Hill-based muscle models: A systems engineering perspective," In Multiple Muscle Systems: Biomechanics and Movement Organization, JM Winters and SLY Woo eds, Springer-Verlag (1990).
- Winters J.M. and Stark L., "Estimated mechanical properties of synergistic muscles involved in movements of a variety of human joints," *J Biomechanics* 21:1027-1042, (1988).

Woodruff, J.P., "KOVEC User's Manual," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-51079 (1973).

Yen, C.F. and Caiazzo, A, "Innovative Processing of Multifunctional Composite Armor for Ground Vehicles," ARL-CR-484, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, (2001).

Yunus, S.M., Pawlak, T.P., and Cook, R.D., " Solid Elements with Rotational Degrees of Freedom, Part I-Hexahedron Elements," To be published, (1989).

Zajac F.E., "Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control, "CRC Critical Reviews in Biomedical Engineering 17(4):359-411, (1989).

