# R15.0.2 Release Notes

March 27, 2024

# 1 Release notes revision

This file constitutes revision 0 of the release notes for LS-DYNA version R15.0.2.

# 2 License

The string "REVISION 15" must appear in the LS-DYNA license file to run version R15.0.2. Please contact your LS-DYNA distributor or your Ansys sales representative if you need to have your license updated.

# 3 Documentation

The R15.0 User's Manual provides documentation for R15.0.2. To download these manuals, go to https://lsdyna.ansys.com/manuals/.

# 4 Notes

The remainder of this file briefly describes new features, enhancements, and bug fixes in versions R15.0.2, R15.0.1, and R15.0.0. Other recent releases may also include some of these updates. We have included all these release notes together in one file for convenience.

We arranged the items by category. Understand that, in many cases, a particular item may pertain to more than one category. In the interest of brevity, we list each item only once, under a single category.

# 5 R15.0.2 Release Notes

## 5.1 Bug Fixes

### 5.1.1 Miscellaneous

1. Fixed licensing compatibility issue impacting a small number of users.

# 6 R15.0.1 Release Notes

## 6.1 Bug Fixes

### 6.1.1 Forming Analysis

1. Fixed bug in Windows MPP LS-DYNA for metal forming problems that involved adaptivity. The bug caused the run to fail due to an i/o issue.

# 7 R15.0.0 Release Notes

## 7.1 New

### 7.1.1 Acoustics, NVH, and Frequency Domain

1. Added METHOD = 311 (collocation direct BEM with fast matrix assembly) to *FREQUENCY_DOMAIN_ACOUSTIC_BEM. This method is a new version of collocation direct BEM that uses skeletonized interpolation to speed up matrix assembly. It is up to five times faster than METHOD = 3 for large problems.

2. Added new keyword *LOAD_SSD_SEGMENT. When using this new keyword in combination with *CONTROL_IMPLICIT_SSD_DIRECT, this instruction defines frequency-dependent pressure segments with both phase and magnitude components. *LOAD_SEGMENT is conversely limited to specifying the magnitude and assumes a zero-phase angle.

3. Added new material *MAT_090_POROUS_JCA / *MAT_ACOUSTIC_POROUOS_JCA. It is a Johnson-Champoux-Allard equivalent acoustic fluid material model. It is to be used with acoustic isoparametric and spectral volume elements in *CONTROL_IMPLICIT_SSD_DIRECT solutions.

4. Added new keywords *BOUNDARY_FLUIDM, *BOUNDARY_FLUIDM_INTERIOR, *BOUNDARY_-FLUIDM_BOTTOM, and *BOUNDARY_FLUIDM_FREE_SURFACE. These keywords are for adding boundaries of fluid mass to structural surfaces for structural vibration solutions of *CONTROL_IM-PLICIT_EIGENVALUE.

5. Added the SUBCASE keyword option to *FREQUENCY_DOMAIN_FRF to include multiple load cases with one keyword.

6. Implemented ATV (acoustic transfer vector) computation for collocation BEM (METHOD = 3 on *FRE-QUENCY_DOMAIN_ACOUSTIC_BEM with the ATV keyword option). Added d3atv output for this method.

7. Implemented missing mass correction method for response spectrum analysis with the MISSING_MASS_CORRECTION keyword option on *FREQUENCY_DOMAIN_RESPONSE_SPEC-TRUM.

### 7.1.2 Airbags

1. Added *DEFINE_CPM_SWITCH_REGION to define a region where user input parameters can control particle-to-particle collisions.

2. Support CPM in the dynamic relaxation phase. CPM particles can move randomly, but the interaction forces between particles and the structure are disabled.

### 7.1.3 ALE

1. Support mass scaling with S-ALE.

2. Support dynamic relaxation with S-ALE.

3. Support S-ALE mesh trimming in 2D.

4. Support S-ALE to work with *ALE_UP_SWITCH. This switch turns off airbags coupling with S-ALE gas and turns on Uniform Pressure airbags.

5. Support S-ALE mesh motion for merged/refined meshes. The algorithm now groups into a single "mesh group" connected S-ALE meshes or S-ALE meshes autogenerated from the same parent mesh. The mesh motion is calculated on all the mesh control points in that mesh group and then updated accordingly. Thus, mesh motion applied to one mesh in the group applies to all meshes in the group.

6. Added keyword option REGION to *ALE_STRUCTURED_MESH_REFINE to restrict refinement of an S-ALE mesh to a region.

7. Added *ALE_STRUCTURED_MESH_JOIN to join ALE structured meshes.

8. Support *ALE_MESH_INTERFACE for 2D. In 2D, it meshes ALE materials with triangles. Then, the subsequent Lagrangian run can be initialized on the triangular mesh with *INITIAL_LAG_MAPPING. It also supports mapping velocities for initialization in a Lagrangian run.

9. Added *CONTROL_MAPPING_SHIFT_BY_DT to shift the initial time of the second run.

### 7.1.4 Chemistry Cards

1. Enabled *CHEMISTRY_CONTROL_INFLATOR and *CHEMISTRY_INFLATOR_PROPERTIES. With this, the 0-D inflator modeling capabilities are now available. These keywords have been in Volume III of the manual for some time, but the keywords were only enabled with this release.

### 7.1.5 Contact

1. Added new MPP specific field, IGTOL, to Card MPP 2 of *CONTACT_OPTION. This field takes a real number to specify a scale factor for determining the contact exclusion distance. It works for AUTOMATIC_GENERAL and AUTOMATIC_SINGLE_SURFACE contacts only for SOFT = 0 or 1. Setting IGTOL > 0.0 causes the generation of a list of nodes excluded from contact for each segment. Each list contains nodes within IGTOL*(segment thickness + node thickness) as measured by traveling along the edges of the contact segment. This option will improve the behavior for cases when SOFT = 0 or 1, SSTHK = 1 on *CONTROL_CONTACT, and the mesh size is equal to or smaller than the thickness. Without it, spurious stresses/deformation might be observed depending on the geometry.

### 7.1.6 Discrete Element Method

1. Added keywords *DATABASE_DEBOND and *DATABASE_HISTORY_DES_(SET). *DATABASE_-DEBOND produces the debond ASCII or binout file to provide the mode of bond breakage between

discrete element spheres. *DATABASE_HISTORY_DES_(SET) limits the bonds monitored for output to debond to all bonds associated with specific node IDs instead of all DES bonds in the model.

2. *DEFINE_DE_MESH_SPHERE generates embedded DES in target solid parts for coupling ICFD and porous structures.

3. Added keyword *DEFINE_DE_MESH_VOLUME to generate embedded DES particles inside SOLID element for SOLID/ICFD (porous medium) coupling.

4. Added *DEFINE_DE_TEMP to enable a temperature effect on the radius of DEM particles. The radius of a DEM particle increases or decreases with the temperature change.

5. Added *DATABASE_DEFRAGMENT to output the fragment volume, radius, and maximum size at a specified time interval for the bonded DEM.

## 7.1.7 Dual CESE Compressible Fluid Solver

1. Added phase-change model solvers, including versions with FSI capabilities. That is, 2D, axisymmetric, and 3D fluid-only phase-change solvers have been added, as well as 2D, axisymmetric, and 3D versions with immersed-boundary method (IBM) FSI capabilities. In this model, three materials (liquid, vapor, and air) can be handled, and the generalized 'Nobel-Abel Stiffened Gas (NASG)' type EOS is adopted. Its main applications include cavitation, evaporation, etc.

2. For these dual CESE phase-change solvers, added appropriate d3plot output. Also, time history output for the phase-change state variables has been included. These go out to the binout file. LS-PrePost now supports these time history variables via its binout interface.

3. Added a new point-source capability for injecting a gas flow into the interior of the dual CESE mesh. This capability is especially useful for injecting the inflator's output in folded-bag deployment simulations. In addition, it provides a more flexible way to specify a folded-bag inlet boundary condition. The available options are:

   a. a mass flow rate plus other appropriate variables
   b. flow variables, such as those used for prescribed boundary conditions.

   This capability is supported in the single-fluid and two-phase multiphase flow FSI-IBM solvers and the pure single-fluid and two-phase flow solvers (that is, the solver versions without FSI). The keywords associated with this capability are *DUALCESE_POINT_SOURCE, *DUALCESE_POINT_SOURCE_TWO-PHASE, and *DUALCESE_POINT_SOURCE_STRUCTNODE_SET.

4. Another significant addition to the R15 dual CESE solver is the ability to output time histories of drag, lift, and related quantities to the binout file using *DUALCESE_DATABASE_HISTORY_DRAG_LIFT. This feature is particularly designed to work in IBM-FSI calculations where structural parts break apart and form new pieces. To do this, the dual CESE solver automatically computes the topologically separate pieces of all structural parts active in the FSI calculation. Due to material erosion during IBM-FSI simulations, each of the structural pieces for which drag and lift are being computed can have different starting and stopping times for the binout output of the drag-related variables. Here are the possible reasons:

   a. A piece can break off from another structural piece at any given time during the simulation.
   b. A structural piece can completely erode away (disappear) from the problem.

   Plotting these drag-related variables is now possible with the current version of LS-PrePost 4.11, using its binout interface.

5. Introduced several new boundary conditions for inlets and outlets appropriate for gas turbines and similar applications. The keywords associated with these new boundary conditions are *DUALCESE_BOUNDARY_PRESCRIBED_INLET_MACHNUM, *DUALCESE_BOUNDARY_PRESCRIBED_INLET_PRESSURE, *DUALCESE_BOUNDARY_PRESCRIBED_INLET_MASSFLRATE, *DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_FARFIELD, *DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_PRESSURE, and *DUALCESE_BOUNDARY_PRESCRIBED_OUTLET_EXMFRATE. These keywords provide the ability to specify the Mach number, mass flow rate, etc., at the problem inlet and to set the Mach number, mass flow rate, or pressure at a problem outlet. These new boundary conditions enable the use of more natural values for some application areas like mass flow rate, Mach number, total pressure, total temperature, and static pressure. In addition, they allow specifying the flow direction and artificial walls at the boundary.

### 7.1.8 EM (Electromagnetic Solver)

1. Introduced a new frequency-based Eddy current solver (see EMSOL = 4 in *EM_CONTROL). This new solver supports sources specified with *EM_CIRCUIT, *EM_CIRCUIT_SOURCE, and *EM_EXTERNAL_FIELD. It also enables defining source terms by an amplitude and a phase shift.

2. Introduced analytical laws in nonlinear magnetic materials. See *EM_EOS_PERMEABILITY.

3. Added keyword *EM_OUTPUT_FORCES to enable outputting the EM forces at each node to a binary or ASCII file. When the ASCII file option is selected, the output format corresponds to a *LOAD_NODE keyword format, allowing for subsequent solid mechanics analyses using the results from a previous EM run. Vibroacoustics models are an example of where this feature may be useful.

4. Added capabilities for radiofrequency (RF) heating. It is an extension of the resistive heating solver that uses complex numbers. Dielectric material properties are defined in *EM_MAT_001 on Card 2.

5. Added coupling method for the scalar potential between the solid mechanics and ICFD solvers in FSI cases. See FSIB on *EM_CONTROL_COUPLING.

6. Added keyword *EM_DATABASE_CIRCUITSOURCE to calculate total magnetic flux per Circuit Source ID given. It is useful for retrieving quantities like inductance or resistance variations. Frequently, it is used in crack detection problems.

7. Added keyword *EM_EP_EIKONAL to set up eikonal (EMSOL = 14 on *EM_CONTROL) or reaction eikonal (EMSOL = 15 on *EM_CONTROL) models.

8. Added isochrones with depolarization and repolarization times. See *EM_EP_ISOCH.

### 7.1.9 Fatigue Cards

1. Added keyword *DATABASE_FATIGUE_STRESS_CYCLE to output the number of stress cycles and fatigue damage versus stress range xyplot file for random vibration fatigue analysis.

2. Added keyword option SCALE_FACTOR to *INITAL_FATIGUE_DAMAGE_RATIO. This keyword option enables defining a scale factor for the initial damage ratio before adding it to the new fatigue damage ratio and writing the results to d3ftg.

### 7.1.10 Forming Analysis

1. Added keyword option GENERAL to *INCLUDE_STAMPED_PART_SOLID_TO_SOLID. The default mapping uses the top surface to find the matching elements in each layer. In this case, it requires the solid model to be extruded from a shell part and the part to have the same number of layers in each location. Advantageously, this method can find the corresponding layer for mapping. When the solid part is not extruded, the default algorithm does not work. The keyword option GENERAL invokes an algorithm that finds the closest element in the source and maps the information. However, this approach might not find the right layer in the mapping.

### 7.1.11 ICFD (Incompressible Fluid Solver)

1. Added keyword *ICFD_DEFINE_TRANSFORM. This keyword enables applying translations, rotations, and scaling on the initial surface mesh before generating the volume mesh. This feature enables quick changes (such as when switching between unit systems) on the geometry without having to edit the case manually.

2. It is now possible to include the ICFD solver in the dynamic relaxation phase (mesh movement is solved only). This capability enables initial dynamic relaxation on the structure followed by an FSI analysis in a single simulation. See *ICFD_CONTROL_GENERAL. Additional options added to *ICFD_CONTROL_-ADAPT, *ICFD_CONTROL_TIME, *ICFD_CONTROL_IMPOSED_MOVE, *ICFD_BOUNDARY_PRE-SCRIBED_MOVEMESH, and *MESH_BL.

3. Introduced a 2D axisymmetric solver. See *ICFD_CONTROL_GENERAL.

4. Improved the support and integration of Twin Builder. We added support for both static and dynamic ROMs and for the selection of the variables used as excitations and responses. See new keyword *ICFD_DATABASE_TWINBUILDER.

For more details, check the Twin Builder manual for dynamic and static ROMs. Users can add custom responses or excitations by augmenting the results provided by ICFD. This can be done through scripting if necessary.

5. Added keyword *ICFD_CONTROL_ADVECTION which enables a new advection technique for different parts of the solver.

6. Introduced a 2D axisymmetric solver. See SOLCL on *ICFD_CONTROL_GENERAL.

### 7.1.12 Implicit (Mechanical) Solver

1. The Block Low-Rank approximation (BLR) allows LS-DYNA's multifrontal solvers (options 0, 7, and 30) to reduce the storage and operations needed to factor a sparse linear system. The trade-off is that they reduce the accuracy of the factors and turn them into high-quality preconditioners for a subsequent iterative solver, such as Preconditioned Conjugate Gradients (PCG). Users are encouraged to experiment with BLR and report back to Ansys whether it improves LS-DYNA run time or reduces resource requirements. As described below, *CONTROL_IMPLICIT_SOLVER controls BLR.

    For solver option 7 (or 0), RPARM4 (Card 2, field seven) is the nominal maximum dimension of an off-diagonal block that will be subject to BLR. In our experience, 240 is a good setting. For both multifrontal codes, options 7 or 30, RPARM5 (Card 2, field 8) is the tolerance below which BLR eliminates rows and columns from off-diagonal blocks of the factors. Setting RPARM5 to 1.0E-7 is roughly equivalent to reducing the accuracy to single precision. If RPARM5 is greater than 1.0E-7, users choosing solver 7 are also encouraged to try setting ISINGLE (Card 3, field four) to two. This should further reduce storage and runtime.

    As stated above, BLR turns a direct solver into a high-quality preconditioner for an iterative solver. As with solver options 22-26, the absolute and relative tolerances for convergence are RPARM1 and RPARM2 (Card 2, fields four and five). Errors introduced by BLR can result in negative diagonal entries in the factors, even if the matrix is symmetric and positive definite. If a user is confident that their problem is not indefinite, they can set the ISPD flag (Card 3, field seven) to ensure that the Conjugate Gradient method is performed.

    For the default multifrontal code (solver options 0 or 7), IBLROPT (Card 3, field five) controls the aggressiveness with which BLR is applied. IBLROPT of 0 (default) disables BLR. IBLROPT of 1 instructs the solver to factor in full precision, then use BLR to compress the storage afterward, reducing the overall memory required. IBLROPT of 2 instructs the solver to eliminate fully assembled (FA) equations in full precision and to apply low-rank updates to their parents in the supernodal elimination tree. IBLROPT of 3 performs low-rank updates with the FA equations. IBLROPT of 4 applies BLR to compress the real stack. For problems amenable to BLR, IBLROPT of three should minimize runtime.

2. Implemented a Jacobian-Free Newton-Krylov (JFNK) GMRES method as an alternative to the BFGS in implicit. Setting NSOLVR = 13 on *CONTROL_IMPLICIT_SOLUTION activates this method.

3. Added new eigensolver option (EIGMTH = 103 on *CONTROL_IMPLICIT_EIGENVALUE) called Fast Lanczos for computing thousands of approximate eigenmodes targeting the NVH application.

### 7.1.13 Isogeometric Analysis (IGA)

1. Added coupling between the electrophysiology (EP) solver in EM with the IGA solid mechanics solver for untrimmed and unstructured solids. This new feature enables solving coupled electrophysiology and solid mechanics problems using different meshes for the different physics.

2. Added binary input for all *IGA keywords using the *INCLUDE keyword.

3. Added keywords *IGA_2D_BASIS_TRANSFORM_XYZ and *IGA_3D_BASIS_TRANSFORM_XYZ for defining a physical bivariate and trivariate spline, respectively, expressed as a basis transform. The keywords *IGA_EDGE_UVW, *IGA_FACE_UVW, *IGA_FACE_XYZ, *IGA_POINT_UVW, and *IGA_VOLUME_XYZ can use these basis transform keywords when using the BASIS_TRANSFORM keyword option.

4. Added *IGA_MASS to add mass to IGA points, edges, and faces.

5. Added *LOAD_EDGE_UVW to apply a uniformly distributed load on IGA edges.

6. Added keywords *IGA_INTEGRATION_SHELL_REDUCE and *IGA_INTEGRATION_SOLID_RE-

DUCE to apply a reduced Generalized Gaussian Quadrature rule to IGA shells and IGA solids, respectively. If there is IGA element failure, usually defined in *DEFINE_ELEMENT_EROSION, *MAT_ADD_EROSION, or *MAT_ADD_DAMAGE_DIEM/GISSMO, the Generalized Gaussian Quadrature rule is turned off and set to a reduced Gauss-Legendre integration rule.

## 7.1.14 Materials and Equations-of-State

1. Implemented a Displacement-Pressure (*U-P*) mixed finite element method for incompressible hypermaterials when using implicit. It is only available for solids with ELFORM 2/-2, 10, 15, and 16 for materials *MAT_027/*MAT_MOONEY-RIVLIN_RUBBER and *MAT_077_H/*MAT_HYPERELAS-TIC_RUBBER. To enable this method, set the Poisson's ratio, PR, to 0.5.

2. Developed *MAT_199_27P as an extension to *MAT_199. The yield criterion for this model involves up to 27 material parameters.

## 7.1.15 Mesh Cards

1. Added *MESH_JOIN to join together two distinct *MESH_VOLUMEs, provided they have matching surface part geometries and meshes. This feature enables having a single continuous mesh while applying different mesh criteria on different regions (mesh size limit, adaptivity criteria, etc).

## 7.1.16 MPP

1. LS-DYNA creates the following scripts at the beginning of the run to be used after the job starts to help track MPP-related bugs remotely. The user can execute the first script and provide the screen output to the developer for cluster debugging.

```
$JOBID.exec-gdb.sh
$JOBID.exec-gdb-$HOSTNAME1.sh
$JOBID.exec-gdb-$HOSTNAME2.sh
```

2. Added keyword *CONTROL_MPP_IO_NOTIEDIO to invoke using memory instead IO for storing the tied_nodes temp files to avoid problems on distributed file systems. The equivalent pfile command is `general { notiedio }`.

## 7.1.17 Output

1. Added sphhtc branch to the binout file when *DATABASE_ISPHHTC keyword exists in the input file. Every time an isphhtc file is created, we also compute mass-averaged HTC values per part and fill the sphhtc branch.

## 7.1.18 Sensors

1. Added output file snsrout with *DATABASE_SNSROUT to trace the values of *SENSOR.

## 7.1.19 SPG (Smooth Particle Galerkin)

1. Implemented a new ISPG solver. This new solver is not compatible with the old ISPG solver that uses *SECTION_FPD and *MAT_IFPD. The following keywords are related to this solver:

   - *ISPG_ASCII_CPL
   - *ISPG_ASCII_SUMFORC
   - *ISPG_BOUNDARY_SYMMETRIC
   - *ISPG_CONTROL_ADAPTIVITY
   - *ISPG_CONTROL_IMPLICIT
   - *ISPG_CONTROL_MPP
   - *ISPG_DAMPING_SURF
   - *ISPG_LOAD_GRAVITY
   - *DEFINE_ISPG_TO_SURFACE_COUPLING
   - *INCLUDE_ISPG
   - *MAT_ISPG_CARREAU
   - *MAT_ISPG_CROSSMODEL
   - *MAT_ISPG_ISO_NEWTONIAN

- *SECTION_ISPG

All of the above keywords, except *INCLUDE_ISPG and *DEFINE_ISPG_TO_SURFACE_COUPLING cannot be in the normal LS-DYNA input deck but must be in an input file included with *INCLUDE_-ISPG.

This ISPG solver supports nodal-cloud-based adaptivity (*ISPG_CONTROL_ADAPTIVITY). This ISPG adaptivity is in-core. The node insertion and deletion are automatically done based on the nodal distribution. A surface cleaning algorithm reconstructs the geometry accurately after remeshing. The nodal-cloud-based adaptivity improves the accuracy of ISPG and enables the accurate alignment of the ISPG fluid surface with the solid surfaces. Moreover, ISPG can handle fluid fusion, such as solder merging.

*DEFINE_ISPG_TO_SURFACE_COUPLING enables FSI coupling between the adaptive ISPG fluid and structures. Wall adhesive forces can be defined with a contact angle. The effect of the ISPG fluid edge velocity on the contact angle can be modeled by defining the static, advancing, and receding contact angles.

This adaptive ISPG solver also includes a customized MPP algorithm for large-scale solder reflow simulation using adaptive ISPG. The MPP algorithm features automatic distribution of the solder balls into different CPU cores based on the total number of the solder parts (no need to use PFILE to define the decomposition) and takes care of the decomposition when the simulation includes coupling between ISPG fluids and solid structures. *ISPG_CONTROL_MPP enables controlling the MPI decomposition for the ISPG fluid. Currently, this keyword is valid for the decomposition of the ISPG fluid parts onto different CPU cores. During the decomposition, the nodes and elements of one fluid part are decomposed onto the same CPU core to minimize the MPI communication.

The ISPG solver supports several options for the definition of the fluid's viscosity:

- Newtonian viscosity (i.e., constant viscosity)
- Non-Newtonian viscosity (Carreau and Cross models)
- Temperature-dependent viscosity

Temperature dependence requires including *LOAD_THERMAL_LOAD_CURVE in the LS-DYNA input deck to assign the nodal temperatures to the ISPG parts.

For post-processing and further structural analysis, the solver outputs the final deformed state of the ISPG part to an ASCII file called ispg_export.k. In addition, *ISPG_ASCII_CPL outputs contact information between the coupled ISPG parts and structures to ASCII file ispg_cpl. *ISPG_ASCII_SUM-FORC causes the output of ASCII files ispg_sumforc_above and ispg_sumforc_below. These files contain the total reaction forces to the structure from the ISPG fluid above and below the cutting plane, respectively.

2. Added new keyword *DATABASE_SPGCPL to control the output of SPG coupling forces. Currently, only the ASCII format is written and can be read by LS-PrePost using the rcforc option.

## 7.1.20 SPH

1. Added *DEFINE_SPH_INJECTION_SIMPLIFIED for simple injections where we want the centroid, radius, and normal vector. It simplifies the setup for problems with many injection nozzles.

2. Added support for ICPL = 1 in *DEFINE_SPH_AMBIENT_DRAG. We import results from a prior ICFD analysis that were printed to a CSV file defining air velocities at ICFD mesh points after reaching steady-state analysis.

3. Added *BOUNDARY_SPH_PERIODIC to impose a periodic boundary condition for explicit SPH particles.

4. Added support for interparticle heat transfer for ISPH. Users can, for example, simulate the heat transfer between a boundary with known temperature and a fluid by setting the temperature of ISPH structure particles using *BOUNDARY_TEMPERATURE_SET.

5. Temperature-dependent viscosity can now be specified for ISPH in the *MAT_SPH_INCOMPRESS-IBLE_FLUID keyword with MU < 0.0.

## 7.1.21 Thermal Solver

1. Added SOLVER = 90 to *CONTROL_THERMAL_SOLVER to call a user-supplied linear equation solver.

## 7.2 Improvements and Enhancements

### 7.2.1 Acoustics, NVH, and Frequency Domain

1. Fully multithreaded METHOD = 3 (collocation direct BEM) of *FREQUENCY_DOMAIN_ACOUSTIC_-BEM, leading to significant performance improvements when using SMP or HYBRID executables and ncpu > 1.

2. Part-by-part structural damping invoked with *DAMPING_PART_STRUCTURAL is now supported in *CONTROL_IMPLICIT_SSD_DIRECT solutions.

3. For *FREQUENCY_DOMAIN_FRF added load type VAD1 = 12 for torsion.

4. Added support for structural damping, defined by *DAMPING_STRUCTURAL to *FREQUENCY_DO-MAIN_SSD_DIRECT.

5. For *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE,

   - Added MFTG = -1 for Steinberg's method to use peak crossing frequency in the computation. The old version of Steinberg's method, activated by MFTG = 1, uses zero-crossing frequency with a positive slope.
   - Added counted cycles output to d3ftg for random vibration fatigue analysis.

6. For *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE:

   - Added a new parameter (STRSF in card 7) to define a stress range factor for random vibration fatigue analysis. In the past, a fixed value of 10 was used for the stress range factor.
   - Added support to use *MAT_ADD_FATIGUE in random vibration fatigue analysis, with SN curves defined by equations.

7. For *DATABASE_FREQUENCY_ASCII_NODFOR_SSD, enabled NODFOR_SSD output for the SUBCASE keyword option.

8. For *DATABASE_FREQUENCY_BINARY_D3SSD:

   - Implemented PSETID to define the set of parts for D3SSD output.
   - Added frequency spacing options FSPACE = 4 and 5 for D3SSD. FSPACE = 4 enables biased spacing between eigenfrequencies for the output frequencies. FSPACE = 5 causes biased spacing in frequency ranges, which spread around the eigenfrequencies for the output frequencies.
   - Added bias parameter (BIAS) and eigenfrequency spread ratio (SPREAD) for frequency spacing definition.

9. For *DATABASE_FREQUENCY_ASCII_OPTION with keyword options NODOUT_SSD, ELOUT_SSD, and NODFOR_SSD:

   - Added frequency spacing options FSPACE = 4 and 5 for databases NODFOR_SSD, NODOUT_SSD, and ELOUT_SSD. FSPACE = 4 enables biased spacing between eigenfrequencies for the output frequencies. FSPACE = 5 causes biased spacing in frequency ranges, which spread around the eigenfrequencies for the output frequencies.
   - Added bias parameter (BIAS) and eigenfrequency spread ratio (SPREAD) for frequency spacing definition.

10. Removed the limit on the number of D3PSD databases for summation (Card 1b on *DATABASE_FRE-QUENCY_BINARY_D3SPD_SUMMATION).

11. Removed the limit on the number of D3RMS databases for summation (Card 1b on *DATABASE_FRE-QUENCY_BINARY_D3RMS_SUMMATION).

12. Update in accessing bevel and bevelf user-data files with *FREQUENCY_DOMAIN_ACOUSTIC_BEM to support CASE control.

13. Implemented output of nodfor_ssd for MPP.

14. Added option RESTRT = 30 to *FREQUENCY_DOMAIN_ACOUSTIC_BEM to use user-defined velocity or acceleration time history data from bevel.lsda.

15. Added option RESTRT = -30 to *FREQUENCY_DOMAIN_ACOUSTIC_BEM to use user-defined velocity or acceleration spectra from bevelf.lsda.

16. Support running FRF (*FREQUENCY_DOMAIN_FRF) with single precision executables.

17. Added output of nodfor_spcm in MPP.

18. Support running response spectrum analysis (*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM) with single precision executables.

19. Added option BINARY = 3 to *DATABASE_FREQUENCY_BINARY_D3RMS to add static results to RMS results for writing d3rms.

20. For *FREQUENCY_DOMAIN_RANDOM_VIBRATION, fixed a bug in accessing the d3eigv family files when MDMIN is larger than 1.

21. Support running random vibration (*FREQUENCY_DOMAIN_RANDOM_VIBRATION) with single precision executables.

22. Added new database secforc_ssd for output of the cross-section force in SSD. See *DATABASE_FREQUENCY_ASCII_SECFORC_SSD.

23. Support running SSD (*FREQEUNCY_DOMAIN_SSD) with single precision executables.

24. Added flag PREF to *FREQUENCY_DOMAIN_SEA_SUBSYSTEM to define reference sound pressure based on a different unit system.

## 7.2.2 Adaptivity

1. Added damage accumulation for the adaptive part, which serves as a failure indicator for forged products. The damage mechanism is uncoupled from the material response. It is implemented for material types 24, 106, 107, and 224. ADPDAM > 0 on Card 2b of *CONTROL_ADAPTIVE invokes this feature.

2. Added flag ADPD3P to Card 1b of *CONTROL_ADAPTIVE to control output with 3D $r$-adaptivity.

3. Added strain-rate scaling to 3D $r$-adaptivity with ADPSCL > 0.0 to obtain the correct stress response in forging simulations. In forging simulations, the tool speed is increased to speed up the simulation. The strain rate needs to be scaled down for rate-dependent materials to obtain correct stresses. This feature is implemented for material types 24, 106, 107, and 224.

4. Implemented a time-dependent minimum adaptive size by using a load curve (see RMIN < 0.0 on *CONTROL_REMESHING).

5. For in-core adaptivity (INMEMORY = 1 on *CONTROL_ADAPTIVE), made improvements that resulted in the adaptive mesh in MPP being more consistent with that produced in SMP for metal forming applications.

6. Added remeshing shells in a region bounded by a specified curve *CONTROL_2D_REMESHING_REGION. Setting LTYP = 10 invokes this feature.

7. Added LTYP = 8 to *CONTROL_2D_REMESHING_REGION to specify a remeshing region with 2D segments. A shell is in the region if nominally all the segment normal vectors point towards it.

8. Added LTYP = 9 to *CONTROL_2D_REMESHING_REGION.

## 7.2.3 Airbags

1. Added field DPWR to *AIRBAG_PARTICLE for IAIR = 4. This field controls the pressure effect of resident air in the unfolded region against ambient pressure. A larger DPWR removes the effect faster.

2. Added option IAIRSTAT to *AIRBAG_PARTICLE for IAIR = 4. This field controls whether initial air particles can pass through vents and pores.

3. Added CDEXT to *DEFINE_CPM_NPDATA to give a part-based exterior drag coefficient for CPM

airbags.

4. LCPC23 on *AIRBAG_PARTICLE can be a *DEFINE_CURVE_FUNCTION or *DEFINE_FUNCTION.

5. Improved CPM-thermal coupling by using the CPM particle temperature as the gas temperature instead of the lumped gas temperature. This change gives a better local heat transfer effect.

6. For IAIR = 4 on *AIRBAG_PARTICLE, added field VNDAMP, which is a damping coefficient, to suppress the motion in the fabric's normal direction.

7. Added VD$i$ = -3 and -4 on *AIRBAG_PARTICLE to enable an offset from the shell's center of the inflator position for shell-based inflators. This option can give consistent results between inflators defined by shells and nodes.

8. Added IOUT to *AIRBAG_REFERENCE_GEOMETRY and *AIRBAG_SHELL_REFERENCE_GEOMETRY to enable outputting the current reference node coordinates to a keyword file.

### 7.2.4 ALE

1. Update the mass scaling algorithm for ALE to account for all materials in the element instead of a single material.

2. Expanded *ALE_STRUCTURED_MESH_TRIM to support more features, namely:

   - Options SALECPT and SALEFAC in *SET_SOLID_GENERAL
   - *BOUNDARY_AMBIENT

3. For 2D S-ALE, add support for glstat. It required special treatment for nodal mass.

4. Improved *ALE_FSI_SWITCH_MMG to better capture and switch fluid flowing past the structure.

5. For 2D S-ALE with FSI (*ALE_STRUCTURED_FSI):

   - Use the segment's normal vector instead of the nodal normal at edge segments.
   - Disable FSI along the symmetric axis (y-axis) for axisymmetric.

6. For S-ALE with FSI (*ALE_STRUCTURED_FSI):

   - Improved stability by excluding opposing structural segments (almost aligned segments with normal vectors in opposite directions) from the coupling segment list.
   - Enhanced the algorithm to find edge nodes (nodes connected to two or more segments with orthogonal normals to each other).

7. Added new logic to ensure velocity monotonicity during advection for S-ALE.

8. Reorganized the MPP communication for S-ALE to be a "non-blocking" pattern to avoid bottlenecks.

9. S-ALE 2D *BOUNDARY_NON_REFLECTING, Improved the bulk modulus and shear modulus calculation in mixed cells for 2D S-ALE with a non-reflecting boundary (*BOUNDARY_NON_RE-FLECTING).

10. Support an S-ALE mesh merge in which the merged mesh contains holes.

11. Made an improvement to S-ALE volume filling for GEOM = PARTSET and SEGSET on *ALE_STRUC-TURED_MESH_VOLUME_FILLING. The enhanced logic finds cells far away from the structures and marks them as inside or outside. This improvement solves the "ghost" material fillings.

12. Made improvements to using S-ALE with *ALE_FSI_SWITCH_MMG to better capture material flow through holes.

13. Automatically add reference pressure (defined in *ALE_STRUCTURED_MULTI-MATERIAL_-GROUP) for *EOS_JWL/JWLB. This change is to balance high explosive pressure with the surrounding air and to model the underpressure phase.

14. Improved algorithm of S-ALE initial volume filling. This update improves filling by setting up boxes around Lagrange geometry to help identify inside/outside.

15. Added to *BOUNDARY_ALE_MAPPING outputting the mapping file for S-ALE.

16. For *ALE_MESH_INTERFACE, added the following:

    - Output triangular shells of length EDGMIN to represent 2D areas of selected ALE groups (in alemeshmatvol.k).
    - Output beams of length EDGMIN to represent 2D material interfaces as contours of selected ALE groups (in aleremeshmatint.k).
    - Output beams to represent 2D material interfaces as contours of selected ALE groups (in alemeshmatint.k).

17. Enabled *DATABASE_TRACER_ALE to add tracers in ALE 2D.

18. Support S-ALE for *ALE_MAPPING.

19. Support S-ALE for *BOUNDARY_ALE_MAPPING.

20. For S-ALE, support *SET_NODE_GENERAL, *SET_SEGMENT_GENERAL, and *SET_SOLID_GEN-ERAL with OPTION:

    - BOX
    - DBOX
    - PART
    - DPART
    - VOL (for *SET_NODE_GENERAL)

21. For 2D S-ALE, support *SET_SHELL_GENERAL with OPTION:

    - BOX
    - DBOX
    - PART
    - DPART

## 7.2.5 Battery Cards (Electrochemistry Solver)

1. Added keyword count output for *BATTERY keywords.

2. Made the various solvers more robust.

## 7.2.6 Boundary Cards

1. Support start and ending times for *BOUNDARY_AMBIENT when AMBTYP = 4 when using this boundary condition with S-ALE. The starting and ending times are automatically taken from the load curves, LCID1 and LCID2.

## 7.2.7 CESE Compressible Fluid Solver

1. Made the *CESE chemically-reacting flow solver more reliable by replacing the internal nonlinear solvers with more robust versions. These new methods correct many cases where errors such as the following were encountered:

```
Error 150209 (CHEM+209)
   Chemistry EOS nonlinear temperature solver failed to converge.
   Error code = 50 Last iterate temperature = 0.96106E+03
```

2. Updated the nonlinear solvers used by the 0-D inflator solvers for cases where they were encountering nonlinear temperature solver errors similar to the ones encountered when using the chemically-reacting flow solver.

3. Improved the robustness of the CESE solvers in general.

## 7.2.8 Chemistry Cards

1. Added keyword count output for *CHEMISTRY_BATTERY and *CHEMISTRY_MODEL.

## 7.2.9 Constraint Cards

1. Added support for *CONSTRAINED_NODE_INTERPOLATION in MPP. The MPP version follows the generic definition of nodal displacement interpolation constraints, where the displacements of interpolation nodes depend on the nodes controlling the interpolation. This new development does not apply to IGA or generalized elements. Image parts in RVE analysis (see *RVE_ANALYSIS_FEM) automatically and internally use this keyword.

2. Added frictional resistance during unload/reload of perpendicular post-hole behavior to *CON-STRAINED_SOIL_PILE. See new Card 7a for *CONSTRAINED_SOIL_PILE_CONSTANTS and new input fields HLCFD and FRATIO on Card 6b for *CONSTRAINED_SOIL_PILE_CURVES.

3. Added support for SPR3 (*CONSTRAINED_INTERPOLATION_SPOTWELD) to implicit analysis. It is currently restricted to elastic response.

4. Add new option to *CONSTRAINED_INTERPOLATION_SPOTWELD (SPR3): Scale factors SCARN/SCARS can now refer to curves by setting them to values less than 0. These curves allow the scale factors to be functions of the state of the neighboring SPR3. Thus, the scale factors can reduce the strength as the SPR3 approaches failure.

5. Added option SPRSMD to *CONTROL_CONSTRAINED. This flag sets the "shear moment distribution" for SPR3 with:

   - SPRSMD = 0: Distributed as force pairs (new default), or
   - SPRSMD = 1: Distributed as nodal moments (old behavior).

   This flag acts as a bug fix with a fallback option.

6. Added new interpolation method for *CONSTRAINED_SPR2 and *CONSTRAINED_INTERPOLA-TION_SPOTWELD (SPR3). INTP = 3 applies quadratic weighting, meaning the scale factor increases with the squared distance from the center to achieve higher stresses at the connector periphery.

7. Enhanced the performance of *CONSTRAINED_INTERPOLATION_SPOTWELD (SPR3) and *CON-STRAINED_SPR2 by replacing brute force search by a bucket sort approach. This change should reduce initial CPU time for large models with a huge number of such connector definitions.

8. Added optional force-based failure criterion (see Card 7) to *CONSTRAINED_INTERPOLATION_-SPOTWELD (SPR3). The force-based criterion simplifies the transition from *MAT_100 as an alternative to damage driven by plastic displacements. The same update applies to *MAT_265.

9. Made changes for using functions in SPR3 (*CONSTRAINED_INTERPOLATION_SPOTWELD with MODEL > 10):

   - Added five new arguments, namely, initial yield stresses of both partners, resultant velocity, and two failure terms (normal and shear).
   - Support more materials: 3, 36, 81, 133, 187, 224, 243, 251, and 324.
   - Enabled more parameters to be defined as functions with MODEL > 30. The same change applies to *MAT_265 (*MAT_CONSTRAINED_SPR3).

10. Added new search method for SPR2 and SPR3 invoked by flag SPRSRCH on *CONTROL_CON-STRAINED. SPRSRCH = 1 invokes including not only the nodes inside the search radius but also all nodes from elements inside the search radius.

11. Added the TITLE keyword option to *CONSTRAINED_INTERPOLATION_SPOTWELD and *CON-STRAINED_SPR2. This title is used as the part title of the visualization beams.

### 7.2.10 Contact

1. Decreased the bucketsort search distance for MPP beam-beam contact, which improves performance.

2. Made MPP contact processing order independent of the input order when contact IDs are present. This helps minimize model behavior when the input is rearranged (such as when include file order changes).

3. Improved how the exterior surface of a solid part or part set is computed for contact. The former code produced errors in cases with more than 39.7 million solids.

4. Changed the behavior of the *CONTROL_MPP_CONTACT_GROUPABLE GRP flag for SOFT = 2 contacts. SOFT = 2 contacts are now excluded from the normal meanings of bit values 1 and 4. Setting the bit value to 16 turns on groupable for SOFT = 2.

5. Allow GRPABLE on the MPP2 card of *CONTACT to be set to -1, which turns off groupable for this contact, overriding any value input on *CONTROL_MPP_CONTACT_GROUPABLE.

6. Made speed improvements to the MPP beam-beam contact bucket sort, which improved the speed and robustness of AUTOMATIC_GENERAL contact.

7. Improved friction handling for draw beads (*CONTACT_DRAWBEAD) during the in-core adaptive step.

8. Reworked the Augmented Lagrangian (ALM) contact formulation for *RIGIDWALL and *CONTACT_- ENTITY in implicit. This improvement increased robustness and accuracy.

9. Added calculating analytic properties (mass and inertia) for the rigid body underlying a contact entity. These values enable correctly calculating the energy and other quantities.

10. When solid cohesive elements are connected (as SURFA) using tied contacts, setting COHTIEM = 1 on *CONTROL_CONTACT causes the inclusion of the mass of the SURFB side to the time step estimation of the cohesive elements. This feature enables using low (or zero) cohesive density and decreases the mass scaling of the cohesive elements. Currently, it is not supported for groupable contacts.

11. IGACTC in *CONTROL_CONTACT can work with *IGA keywords.

12. Modified segment-to-segment contact (SOFT = 2) with DEPTH = 25, 33, 35, and 55 to prevent possible energy growth from the contact calculation that can occur prior to any load being applied to the model. This change will subtly change some results with shell segment contact.

13. Added negative birth time and death time options for *CONTACT_2D_AUTOMATIC. Like 3D, a negative birth time applies only to the dynamic relaxation phase. A negative death time makes the contact inactive during the dynamic relaxation phase. Setting TDEATH = -9999 causes TBIRTH to be used as a curve ID for a curve where each point is a birth time and death time.

14. Output the initial penetration of contact nodes to the intfor file under the following conditions:

    - SOFT is set to 0 or 1 on the contact data cards,
    - NPEN = 1 on *DATABASE_EXTENT_INTFOR, and
    - SAPR = 1 or SBPR = 1 on the contact data cards.

    Previously, the initial penetration was only nonzero in the intfor file for Mortar contacts. Now it applies to all contacts unless

    - SOFT = 2, or
    - The contact is a "TIED" or "TIEBREAK" contact.

    For these cases, the initial penetrations are zero in the intfor file. Additionally, the output does not appear in the reference configuration ($t = 0$) but in the subsequent one, indicating that penetration information is not reported for the first configuration.

15. Activate output of pressure in Mortar tiebreak contacts. Previously, the output was all zeros.

16. Implemented directional ties, activated by DIR_TIE on *CONTROL_CONTACT. This feature will tie solid element nodes in the direction of the outward normal rather than to the closest segments where this is possible.

17. Added DIR_TIE on *CONTROL_CONTACT to enable a "directional tie" when a solid element is tied to its surroundings. With this option, solid nodes are preferentially tied in the direction of the outward normal to avoid a collapse of the solid elements to zero volume.

18. Added option SSFTYP = 2 for *CONTACT_… with SOFT = 2 to invoke using the smaller of the SSF values from *PART_CONTACT. Already existing SSFTYP = 1 takes the larger value.

19. Added option SHLOFF to *CONTACT_… on optional Card G to consider shell offsets for individual contacts. SHLOFF is a contact-specific version of the global option CNTCO on *CONTROL_SHELL.

## 7.2.11 Control Cards

1. Improved the keyword processing speed for *CONTROL_SPOTWELD_BEAM when only a subset of the beam spot welds is converted to hex elements.

2. Initial implementation of a "consolidated error file" for MPP, which optionally echos some of the diagnostic output (from all processors) to a single file named error.log. There are three fields in *CONTROL_OUTPUT (columns 1-3 on optional Card 6). Each field corresponds to one "processing phase": keyword input, initialization, and solution. Each field determines what information is output to error.log for the corresponding phase.

3. Extended IACC = 1 on *CONTROL_ACCURACY by including ITORM = 2 on *SECTION_BEAM for improved representation of torsional and rotational modes in eigenvalue analysis with beams.

4. Added DTDYNV to *CONTROL_TIMESTEP to consider the effect of dynamic viscosity on the critical time step.

5. Add support for using groupable tied contacts and contact type *CONTACT_TIED_SHELL_EDGE_-TO_SURFACE in Selective Mass Scaling (IMSCL on *CONTROL_TIMESTEP).

6. When using Selective Mass Scaling (IMSCL on *CONTROL_TIMESTEP), print a second list of 100 smallest time steps, where parts included in SMS are excluded, to the messag files.

7. IGACTC in *CONTROL_CONTACT now can work with IGA shells defined by *IGA keywords.

8. Added CNTCO = 3 and 4 to *CONTROL_SHELL. CNTCO = 3 is similar to CNTCO = 1, but it has improved behavior at corners or folds in the mesh. The existing method has two flaws. One is it uses area-weighted averaging at the nodes to determine a direction for the surface movement. That moves the node in the wrong direction. This is corrected by using thickness-weighted averaging, which moves nodes in the direction the node should be moved. A second fix is to scale the movement so that at corners, all surfaces are moved the correct amount. Now, the corners are set to where the offset surfaces would intersect. CNTCO = 4 is similar to CNTCO = 2 but has the same two corrections previously mentioned.

9. It is now possible to fringe plot kinetic energy density with output to d3plot and d3eigv. Parameter KINENG on *CONTROL_OUTPUT activates this output.

10. Disabled the NOCOPY option on *CONTROL_SOLUTION since it results in slightly more simulation time when not being used. NOCOPY = 0 is currently always active.

11. Equipped interface linking motion with a continuity option to result in smoother boundary motion in the resulting submodel. Use ISFCNT on *CONTROL_OUTPUT for this purpose.

12. The number of *CONTROL_RIGID cards is now output to d3hsp.

13. Added flag COHEQC to *CONTROL_SOLID to control the COHesive Element Quality Check. We added this check in 2020, which caused an error termination if LS-DYNA detected a bad quality (inverted element) in cycle 0 or 1. This flag allows you to change the behavior for the check:

    - COHEQC = 0: Error termination as before (default)
    - COHEQC = 1: Warning and continue
    - COHEQC = 2: Warning with element erosion, continue

14. Added option CRVP to *CONTROL_SOLUTION. Activating this field causes bypassing time-based evaluation of non-time-dependent curves, such as stress-strain curves for materials. This feature can improve CPU performance when many curves are used (such as from a big material database) in smaller models.

15. Added option CRVP to *CONTROL_SOLUTION to possibly improve curve evaluation performance in special cases and perhaps additionally reduce memory requirement of rediscretized curves.

## 7.2.12 Discrete Element Method

1. All *DEFINE_DE_... keywords now support *INCLUDE_TRANSFORM.

2. For *DEFINE_DE_BOND:

    - Added rate sensitivity parameters ARSP and BRSP to Card 3 for bond strengths that increase linearly with deformation rate.
    - Added Card 4 variables to statistically distribute certain bond parameters. IDIST on Card 1 enables distributions, which can be uniform, Gaussian, or Weibull. These distributions can also be deterministic or non-deterministic.

3. Added field BENDSF to *DEFINE_DE_BOND and *DEFINE_DE_BOND_OVERRIDE to remove bending/twisting from bond failure criteria.

4. Added Cards 2a and 2b to *ELEMENT_DISCRETE_SPHERE to statistically distribute mass, inertia, and radius of DES. IDIST on Card 1 enables distributions with a choice of uniform, Gaussian, or Weibull. These distributions can also be deterministic or non-deterministic.

5. Added option ISOFT to *DEFINE_DE_TO_SURFACE_COUPLING for better DES to surface contact.

6. Added support for full deck restarts with DES.

7. Support redecomposition with DEM for better MPP scaling.

8. Added ISOFT = 2 to *DEFINE_DEM_TO_SURFACE_COUPLING to adjust the coupling stiffness between a DES and the target segment for high speed impact.

9. At the start of the simulation, added printing the initial configuration of DES particles in a bond if the coordination number is less than or equal to MAXCN on *DEFINE_DE_BOND.

### 7.2.13 Dual CESE Compressible Fluid Solver

1. Added the convenience feature of converting boundary faces missing an assigned boundary condition to have an outflow boundary condition. In addition, added robust error reporting for dealing with all boundary conditions.

2. Added output of the counts of used *DUALCESE keywords.

3. Improved the robustness of the dual CESE solvers.

4. Modified the local CFL number formula with a new time-step-size calculation. In some of the 3D dual CESE example problems distributed on https://lsdyna.ansys.com/knowledge-base/cese/dual-cese/, the given CFL number may need to be reduced to get similar time-step sizes as seen with R14.

### 7.2.14 Elements

1. Added support for more than one *ELEMENT_SHELL_SOURCE_SINK instance in a model.

2. The option ITORM on *SECTION_BEAM now additionally supports beam types 1 and 2.

3. Added the option NCUT to *ELEMENT_BEAM_SOURCE. With NCUT = 0 the last element remains attached to the source node. For NCUT = 1 the last element gets cut from the source node with sudden release of the pull-out force. If NCUT is greater than 1, the pull-out force decreases linearly over the last NCUT elements in the source.

4. Added the option ITORM = 2 on *SECTION_BEAM to recompute the mass related to rotational and torsional modes more accurately for eigenvalue analyses. ITORM = 1 considered only torsional modes. Beam types 1, 2, and 13 are supported.

5. Issue an error message when duplicate 2D belts (*ELEMENT_SEATBELT are detected.

6. Implemented an adaptive load-limiter with LCFL and FLOPT on *ELEMENT_SEATBELT_RETRAC-TOR.

7. Restructured the solid nodal volume calculation of solid element 13 for better performance.

8. Added cohesive elements -19 and -21 to *SECTION_SOLID. They can be used with or without offset, connecting to both solid and shell elements.

9. Added INER = -1 and -2 to *SECTION_BEAM. These options automatically compute the rotational inertia of discrete elements.

10. Added SCOOR = -14 and 14 to *SECTION_BEAM. This field affects discrete beam elements (ELFORM = 6). SCOOR = -14 and 14 are like -13 and 13, respectively, but they conserve energy better. Specifically, with SCOOR = -13, nonphysical rotational energy could accumulate at node 2 of the beam, and with SCOOR = 13, it could accumulate at node 1.

## 7.2.15 EM (Electromagnetic Solver)

1. Improved consistency between MINRES and GMRES for convergence criteria in the monolithic solver. See *EM_SOLVER_FEMBEM_MONOLITHIC.

2. Improved the robustness of the Eddy current 2D axisymmetric solver when using the monolithic solver.

3. Improved the robustness of the Eddy current 2D axisymmetric solver electric contact detection and added support for *EM_CONTACT for this solver. Previous versions only took *EM_CONTROL_CONTACT into account.

4. Introduced three separate ways to include temperature dependency of magnetic permeability for nonlinear magnetic materials. See *EM_EOS_PERMEABILITY and Card 2 of *EM_MAT_002.

5. Improved the accuracy of *EM_CIRCUIT_SOURCE for nonlinear magnetostatic cases.

6. Previous versions always considered circuits defined by *EM_CIRCUIT_SOURCE to be static. Now, positions are correctly updated if those circuits move during the run.

7. Added MTYPE = 7 and 8 for magnetic gear applications to *EM_PERMANENT_MAGNET.

8. Added an option to define a spatially varying external magnetic field using *DEFINE_FUNCTION to *EM_EXTERNAL_FIELD.

9. An external field can now be combined with a case that includes contact. See *EM_EXTERNAL_FIELD and *EM_CONTROL_CONTACT.

10. New volume-based area type option for the batmac model. See RDLAREA = 4 on *EM_RANDLES_BATMAC.

11. Added option to have battery joule heating expressed at the element level rather than the nodal level. This option might result in results that are more diffusive but also more stable (less sharp thermal gradients). See THCPL = 2 on *EM_CONTROL_COUPLING.

12. Added power boundary condition for the resistive heating solver. See CONTYPE = 7 on *EM_ISOPOTENTIAL_CONNECT.

13. Added impedance boundary condition that is only available for radiofrequency (RF) applications. See VAL2 and LCID2 on *EM_BOUNDARY_PRESCRIBED.

14. Added more options to control erosion effects on the short for the solid model in the battery module. See *EM_RANDLES_SHORT.

15. For *EM_EOS_PERMEABILITY, introduced a polynomial interpolation approach for scenarios where input *B-H* curves are not well-defined with few points for improved convergence.

16. For *EM_EP_CELLMODEL_OPTION, added ability to define the cell model on a node set with MID < 0.

17. Added an alternative way of defining 2D axisymmetric Eddy current cases that does not require the definition of segment sets in *EM_2DAXI and *EM_CIRCUIT. See Card 2 on *EM_ROTATION_AXIS.

18. It is now possible to model the tabs as conducting shells in the BATMAC battery module.

19. Restructured some code to speed up solving for the vector potential.

20. Enabled the ability to make the external fields' recomputation frequency (see *EM_CIRCUIT_SOURCE) a function of time by entering a negative value for IFREQST.

21. To save calculation time, perform mesh resistance output to em_circuit.dat files at each matrix recomputation.

22. Improved the robustness of the 2D axisymmetric Eddy current solver electromagnetic contact feature.

23. Added extended monodomain with a second system to solve for the external potential from the transmembrane potential. See parameters EPLAMDA on *EM_MAT_001 and LAMBDA on *EM_MAT_003.

24. Added capability to have several Purkinje networks (e.g., 1 per ventricle). See *EM_EP_PURKINJE_-

25. Improved initial reading of *ELEMENT_SOLID_ORTHO if the fiber orientations are user defined, for large cases.

### 7.2.16 Fatigue Cards

1. Added counted cycles output to d3ftg for time domain fatigue analysis.

2. Added support for using 1 point integration for shell elements in fatigue analysis.

3. Removed the limit on the number of D3FTG databases for initial fatigue damage (Card 1b on *INITIAL_FATIGUE_DAMAGE_RATIO).

4. Removed the limit on the number of D3PLOT databases for initial fatigue damage (Card 1c on *INITIAL_FATIGUE_DAMAGE_RATIO).

5. Enabled using multiple *FATIGUE_MEAN_STRESS_CORRECTION keywords in one model.

### 7.2.17 Forming Analysis

1. Added optional check for the correct material type of the target to *INCLUDE_STAMPED_PART with field MTYPE. MTYPE provides the expected material type of the target part. If MTYPE does not match the material type of the target part, an error is thrown. It is useful if history variables are mapped to avoid an unintended change of material type.

### 7.2.18 ICFD (Incompressible Fluid Solver)

1. Added support for *CHANGE_CURVE_DEFINITION in small deck restarts for load curves used by the ICFD solver.

2. Added flags DTST and DTVISC to *ICFD_CONTROL_TIME to allow including surface tension effects and viscous effects when determining the automatic time step.

3. Added IANG = 1 to *ICFD_CONTROL_IMPOSED_MOVE to enable using Tait-Bryan angles for the rotation.

4. Changed the residual calculation in the steady-state solver to be more representative of the actual convergence of the solution. Previous versions showed slower convergence in the residual than what was effective.

5. Added 2D automatic surface mesh remeshing capabilities. See *ICFD_CONTROL_SURFMESH.

6. Improved the convergence rate of FSI cases by removing a forced line search reinitialization.

7. Added ATYPE = -2 to *ICFD_CONTROL_GENERAL to completely remove ICFD but leave level set or thermal. This allows for the simulation of velocity fields that can be imposed directly using a *DEFINE_FUNCTION.

8. The file icfd_tmp_sm.k is now scaled to match the model dimensions. It used to be the original mesh sent to the mesher.

9. Added ability to add viscous forces to the interface fluid forces in FSI. To activate, set VFORC = 1 or 2 on *ICFD_CONTROL_FSI.

10. Added NITBS to *ICFD_CONTROL_DEM_COUPLING to change the default number of time steps between particle searches in DEM coupling. The default is 50 in explicit and 1 in implicit. Increasing this value results in faster runs at the cost of less accurate particle force/displacement tracking.

11. For DEM coupling, improved FORM = 2 on *ICFD_CONTROL_DEM_COUPLING by computing the pressure gradient at the element instead of using the fluid gradient already computed at the node.

12. Added SRL to *ICFD_CONTROL_LEVELSET to control the reinitialization technique.

13. Added AST to *ICFD_CONTROL_LEVELSET to control surface tension formulation.

14. Added SGC to *ICFD_CONTROL_LEVELSET to control the approximation used for the gradient and

curvature of the level set.

15. Added ability to add an isovelocity condition on top of the imposed pressure with ISO = 1 on *ICFD_-BOUNDARY_PRESCRIBED_PRE.

16. Added ability to restrict adaptivity to one *MESH_VOLUME and exclude others. See *ICFD_CONTROL_ADAPT.

17. Improved the gap detection accuracy of *ICFD_CONTROL_GAP, especially at the intersection of the boundary layer and interior nodes. Also, for FSI cases, stopped transferring the force from the fluid to the structure at the gap location so that only structural contact forces exist at the gaps.

## 7.2.19 Implicit (Mechanical) Solver

1. Support LSOLVR = 22 with LCPACK = 3 (see *CONTROL_IMPLICIT_SOLVER). When using a nonsymmetric matrix assembly (LCPACK = 3), iterative solver options LSOLVR = 22-26 were ignored, and a direct solver was used. Now, LSOLVR = 22 uses a nonsymmetric iterative solver (GMRES). LSOLVR = 23-26 still fallback to a direct solver.

2. The new default heuristic for ORDER = 4 on *CONTROL_IMPLICIT_ORDERING provides significant performance improvements, particularly for very large numbers of MPP processes.

3. For EIGMTH = 111 on *CONTROL_IMPLICIT_EIGENVALUE (sectoral symmetry eigenanalysis), created better error messages for executables that don't support this solver.

4. For LSOLVR = 30 (MUMPS) on *CONTROL_IMPLICIT_SOLVER, upgraded MUMPS from version 5.4.0 to 5.6.2.

5. For MTXDMP > 0 on *CONTROL_IMPLICIT_SOLVER:

   - Changed indfmt from 10I8 to 8I10 in the output for large problems.
   - Write indices along with right-hand side values in the output.

6. For EIGMTH = 102 (LOBPCG) on *CONTROL_IMPLICIT_EIGENVALUE:

   - Improved communication patterns in MPP.
   - Improved performance by reusing ordering and symbolic factorization when the shift changes.
   - Improved logging and diagnostics.
   - Added RPARM2 < 0.0 to force an exact factorization instead of a BLR preconditioner.
   - Improved logging and diagnostics.

7. For LSOLVR = 22 with LCPACK = 3 (GMRES) on *CONTROL_IMPLICT_SOLVER, made performance improvements for very large problems.

8. Added ROTSCL to *CONTROL_IMPLICIT_EIGENVALUE to control scaling the rotational inertia.

9. Added parameter ROTSCL to optional Card 4 of *CONTROL_IMPLICIT_EIGENVALUE. This parameter controls how the rotational inertia terms are scaled in eigenvalue analysis.

10. For intermittent eigenvalue extractions, EIGMSCL on *CONTROL_IMPLICIT_EIGENVALUE controls whether to use the original (unscaled) or the scaled mass (DT2MS ≠ 0 on *CONTROL_TIMESTEP).

11. For key points specified with *CONTROL_IMPLICIT_AUTO, output data to intfor and d3part if these databases are requested, like is already done for d3plot.

12. In implicit dynamics analysis (IMASS = 1 on *CONTROL_IMPLICIT_DYNAMICS), sped-up models containing many load curves, each with a large number of points. In previous versions of LS-DYNA, the analysis could take progressively longer to solve each time step.

13. Made improvements for implicit dynamic analysis (IMASS = 1 on *CONTROL_IMPLICIT_DYNAMIC) when used with *DAMPING_FREQUENCY_RANGE_DEFORM:

   - The stiffness matrix now accounts for the effect of the damping model. This change has some influence on convergence behavior compared to previous versions.
   - Element types that work with this damping card in explicit analysis should now also work in implicit analysis.

14. Print the Implicit Memory Alert message in models that include Intermittent Eigenvalue Computations.

15. Added logic to left justify the matrix names from keyword input and the dmig files (NASTRAN) for *ELEMENT_DIRECT_MATRIX_INPUT when reading them. This improvement makes input easier for the user as they do not have to have the same number of blanks before the matrix names.

16. Patched MPP eigensolvers to deal with the case of the number of requested modes being greater than the matrix order (*CONTROL_IMPLICIT_EIGENVALUE`). This patch saves memory.

17. Added additional error checking to the MPP Lanczos eigensolver (*CONTROL_IMPLICIT_EIGEN-VALUE). The MPP eigensolver was reporting errors and then continuing. Now it terminates at the point of error detection instead of continuing.

18. Enhanced the computation of constraint modes using *CONTROL_IMPLICIT_MODES where a node is attached to rigid bodies. It now allows degrees of freedom on nodes attached to rigid bodies where the rigid body is free to move in those degrees of freedom. For example, if the rigid body is free to move in the z-direction, the attached node can have a prescribed motion in the z-direction. In the past no prescribed motion was allowed on nodes attached to rigid bodies.

19. Provide cleaner termination and error handling for implicit cases where some of the processes do not have enough memory for a direct linear equation solution while other processes do have enough memory.

20. Added additional error messages for MPP Lanczos Eigensolver (*CONTROL_IMPLICIT_SOLVER) for the cases of insufficient I/O space (-711) and indefinite mass matrices (-735).

21. Implicit buckling (*CONTROL_IMPLICIT_BUCKLING), due to the internal sign convention for geometric stiffnesses, requires a sign change for the eigenvalues. This enhancement was to apply the same sign change for the center frequency and interval specification from *CONTROL_IMPLICIT_-EIGENVALUE.

22. Improved the output of residual vectors (*CONTROL_IMPLICIT_RESIDUAL_VECTORS) to d3resvec to include rotational degrees of freedom. Added the ability to use residual vectors in modal dynamics (*CONTROL_IMPLICIT_MODAL_DYNAMIC) to enhance the resolution of specific loading conditions such as point loads.

23. Added check for whether a contact involves an end-of-beam node to vastly reduce the number of end-of-beam node not connected warnings.

24. Enhanced the direct linear equation solver performance in MPP by enabling the reuse of the symbolic processing from the previous factorization if the matrix structure has not changed. See REUSE in *CONTROL_IMPLICIT_ORDERING.

25. Modal stresses are output for discrete elements in eigenvalue analyses.

26. Improved the geometric stiffness for the type 5 membrane element to provide the correct eigenvalues for a preloaded membrane.

27. Properly support solid element -18, beam type 2, and the SPR3 connector for linear analysis in multiple steps and multiple load cases.

28. Support linear analysis with nonlinear contacts, activated by NSOLVR = -12 on *CONTROL_IMPLIC-IT_SOLUTION.

29. Added support for the consistent tangent for solid element 13 (TET13K in *CONTROL_SOLID) in MPP. It is recommended for linear and eigenvalue analysis but not for nonlinear analysis as it is too expensive and not necessary in that case.

30. Support IRATE from *CONTROL_IMPLICIT_DYNAMICS in *MAT_058, *MAT_063 (MODEL > 0), *MAT_187, and *MAT_280.

## 7.2.20 Initial Cards

1. *INITIAL_VELOCITY_GENERATION now supports SPH part IDs (STYP = 1 and 2). Previously, SPH was only supported with node sets (*SET_NODE).

2. Added the following detonating surfaces to *INITIAL_DETONATION_GEOMETRY:

   - Elliptical-based truncate cone
   - Elliptical-based cylinder

- Ellipsoid

3. Allow the initialization of history variables based on nodes (*INITIAL_HISTORY_NODE(_SET)) for the case that the same node is part of various definitions of node set. For instance, if a node is part of two node sets where the history variable #5 is initialized for the first node set and history variable #7 is initialized for the second, both history variables are now initialized for the node.

4. Added option IVFLG to *INTERFACE_SPRINGBACK_OPTION to not write out the initial volume for solid elements in *INITIAL_STRESS_SOLID. By default, this is always written.

## 7.2.21 Isogeometric Analysis (IGA)

1. Updated the default drilling stiffness formulation for the type 3 IGA shells. The new formulation is simpler than the type 4 drilling stiffness, but it is not quite as robust for challenging implicit problems. It does much better for rigid body modes than the previous implementation.

2. Support using *LOAD_FACE_XYZ with *IGA_FACE_XYZ_BASIS_TRANSFORM.

3. *FREQUENCY_DOMAIN_FRF can work with *IGA_FACE_XYZ and *IGA_POINT_UVW for excitation input and *IGA_POINT_UVW for response output.

4. Enabled defining multiple IGA shells (*IGA_SHELL) on a single unstructured patch (*IGA_2D_BASIS_TRANSFORM_XYZ).

5. Performance enhanced for untrimmed IGA solid elements (*IGA_SOLID. Turn on sum factorization on untrimmed elements and turn it off for trimmed elements. This technique accelerates numerical integration.

6. Output all the control points to the error message when there is a negative determinant.

7. Added MPP support for implicit *CONTACT_TIED_SURFACE_TO_SURFACE_OFFSET for the old IGA element keywords (*ELEMENT_SOLID/SHELL_NURBS_PATCH).

8. Added support for gravity loading for isogeometric elements. This support applies to both *IGA_SHELL/SOLID and *ELEMENT_SHELL/SOLIDS_NURBS_PATCH.

9. Allow the definition of multiple *PARTs on the same underlying surface description (*IGA_2D_NURBS_XYZ). This can be done by defining various (in general non-overlapping) sub-faces (*IGA_FACE_XYZ and *IGA_SHELLs. No *IGA_TIED_EDGE_TO_EDGE are applied at the sub-face interfaces because they share the same underlying control-point grid. Thus, full continuity across the sub-face interfaces is maintained. This feature might be interesting when changes in material properties or thicknesses need to be taken into account.

10. Enabled constraint-type tied contacts (*CONTACT_TIED_SHELL_EDGE_TO_SURFACE and *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET) for IGA shells (*IGA_SHELL). For this to work correctly, the reference segments should be from the *IGA_SHELL parts, and the tracked nodes should be standard FE-nodes (such as from cohesive elements and solid spot weld hexahedral elements). This feature is only available in MPP as groupable contact (GRPABLE = 1).

11. Support full deck restarts for IGA shells (*IGA_SHELL).

12. Map the thickness change (ISTUPD ≠ 0 on *CONTROL_SHELL) of the IGA shell elements (*IGA_SHELL) to the associated interpolation elements in every cycle to make sure that the correct thickness is used in the evaluation of contact.

13. Properly map strains from IGA shells (*IGA_SHELL) to associated interpolation shells to allow for a correct post-processing of strains and thickness change (L = 1 for STRFLG on *DATABASE_EXTENT_BINARY).

14. Introduced a purely geometric-based characteristic element length for IGA shells (*IGA_SHELL) that is automatically used for the regularization of damage and failure evolution in *MAT_ADD_DAMAGE_DIEM/GISSMO.

15. For multi-patch *IGA_SHELL models that use *IGA_TIED_EDGE_TO_EDGE to mechanically couple topologically connected surface patches, exclude the interpolation nodes that are located along the tied interface from the standard Edge-Kontakt behavior. This exclusion helps to reduce the appearance of unphysical element erosion along coupling edges when using higher values for EDGEK in Optional Card

C (*CONTACT).

16. Properly map IGA shell (*IGA_SHELL) thickness (at integration points) to related interpolation shells. Even if ISTUPD = 0 on *CONTROL_SHELL, the thickness needs to be mapped in case we do have *INITIAL_STRAIN_IGA_SHELL (ITHK = 1) or *INCLUDE_STAMPED_PART with *ELEMENT_SHELL_THICKNESS.

17. Allow a non-matching number of in-plane and through-thickness integration points in *INITIAL_STRESS/STRAIN_IGA_SHELL.

18. Enable the use of *INCLUDE_STAMPED_PART for *IGA_SHELLs. This feature enables mapping stamping information from a standard finite element model to *IGA_SHELLs.

19. Corrected the computation of mass moment of inertias for IGA shells (*IGA_SHELL). Previously we used a nodal approach like we do for standard finite elements. Now we perform numerical integration for the inertias, taking into account rigid nodes.

## 7.2.22 Load Cards

1. Support specifying the ground motion below the top surface with *LOAD_SEISMIC_SSI_DECONV.

## 7.2.23 Materials and Equations-of-State

1. Added VOLFRAC to *MAT_ADD_DAMAGE_DIEM/GISSMO. VOLFRAC in these keywords has the same functionality as VOLFRAC in *MAT_ADD_EROSION. It is the volume fraction to fail before deleting a higher order or IGA element.

2. Account for additional costs of *MAT_ADD_DAMAGE_DIEM/GISSMO in MPP decomposition. The cost calculation already accounted for shells and the default cost estimate but now also accounts for solids and the newcost decomposition option in the pfile.

3. Add option LCSOFT to *MAT_ADD_DAMAGE_GISSMO. This field references a load curve or table ID. A curve give the soft reduction factor as a function of triaxiality. A table specifies the soft reduction factor as a function of triaxiality and element size. Like for SOFT, the sign of LCSOFT determines which strains are scaled by the factor.

4. Added new option LCSOFT to *MAT_ADD_DAMAGE_GISSMO. This can refer to a load curve or table with ID |LCSOFT| giving the soft reduction factor for failure strain in crashfront elements. It can be defined as a function of triaxiality (curve) or triaxiality and element size (table).

5. Added new option to *MAT_ADD_DAMAGE_GISSMO for triaxiality-dependent regularization. Two new triaxiality values, RGTR1 and RGTR2, between 0 (shear) and 2/3 (biax) together with SHRF = 1 and BIAXF = 1 describe a trapezoidal tub-shaped regularization. This would already be possible with LCREGD < 0, but the new approach is faster.

6. Implemented GISSMO invoked with IDAM > 0 on *MAT_ADD_EROSION for 2D ALE.

7. For *MAT_ADD_EROSION:

   ○ Maximum principal stress SIGP1 can now be a function of strain rate by specifying a negative load curve ID.
   ○ Enabled DTEFLT to be used with SIGP1 < 0 and SIGVM < 0. DTEFLT is the strain rate filtering implemented for MXEPS < 0. Now it is also available for the other two rate-dependent failure criteria.

8. Enabled part-wise definition of the nonlocal energy criterion of *MAT_ADD_EROSION and *MAT_280. This enhancement relates to the input parameters ENGCRT and RADCRT that exist for both materials. Up to now, only one global energy value could be defined (see Remark 1i in *MAT_ADD_EROSION for details). Now each material card definition is taken into account separately.

9. Added option LP2BI to *MAT_ADD_GENERALIZED_DAMAGE (shells only). Failure can now be a function of triaxiality and a bending indicator. The same option already exists for GISSMO.

10. Added support for IRATE from *CONTROL_IMPLICIT_DYNAMICS for truss beams with *MAT_001. There is a viscous damping of 5% by default that can now be switched off.

11. Support using *MAT_015 (*MAT_JOHNSON_COOK) with IGA solid elements.

12. Improved accuracy in the material system calculation for *MAT_033 (*MAT_BARLAT_ANISOTROP-IC_PLASTICITY) with solid elements to avoid instabilities.

13. In *MAT_FABRIC_MAP, the mapping tables can now depend on temperature. Up to 9 temperature points can be selected, each associated with a stress map. The resulting stress is based on interpolation between the maps with respect to the temperature.

14. Equipped coating options for fabric materials with transverse shear stiffness to avoid nonphysical zig-zagging due to lack of shear resistance. Use SHRF on *SECTION_SHELL for this purpose.

15. Added ability to make the stress map in *MAT_FABRIC_MAP temperature dependent.

16. Added better implicit tangent (for solids) to *MAT_038.

17. Added new parameters EXT and EPSHV to *MAT_USER_DEFINE_MATERIAL_MODELS for reading user materials from other finite element codes.

18. Changed the implicit tangent for *MAT_054 for solids if using the Jacobian-Free Newton-Krylov method (NSOLVR = 13 on *CONTROL_IMPLICIT_SOLUTION).

19. Modified longitudinal tensile failure (XT) of *MAT_ENHANCED_COMPOSITE_DAMAGE (*MAT_-054) in implicit solutions when DFAILT = 0. This option was developed to fail slowly over many cycles, which works okay for explicit solutions but prevents failure of implicit solutions. Failure is now immediate in implicit solutions.

20. Enabled *MAT_058_SOLID/*MAT_LAMINATED_COMPOSITE_FABRIC_SOLID to work for thick shell elements (ELFORM = 3, 5, and 7).

21. For *MAT_060C, added an option to allow viscosity to depend on the shear strain rate. Before it could only depend on temperature.

22. Added option MODEL = 2 to *MAT_063. This option provides a tension-compression asymmetric version of MODEL = 1 with one additional new parameter.

23. Improved the viscoplastic algorithm in *MAT_063 for MODEL $\geq$ 1. This change affects RFILTF = 0. Brent's method is used to avoid non-convergence.

24. Added new option to *MAT_063 with MODEL = 1 or 2. LCID can now refer to a *DEFINE_TABLE_3D to make the yield stress a function of strain (CURVE), strain rate (TABLE), and value of history variable #8 (TABLE_3D). History variable #8 can be set with *INITIAL_STRESS_SOLID or *INITIAL_HIS-TORY_NODE. This feature can be used to model foam with spatially varying porosity/density.

25. Take damping into account when estimating the critical time step for *MAT_LINEAR_ELASTIC_DIS-CRETE_BEAM (*MAT_066).

26. Added failure information to the output of *MAT_068. Added failure flags to the output using history variables 12 (force resultants) and 13 (displacement resultants). The flags range from 0 (intact) to 1 (failed).

27. Enabled IGA shells for *MAT_077.

28. For *MAT_GENERAL_HYPERELASTIC_RUBBER/*MAT_OGDEN_RUBBER, a simultaneous fit of uniaxial, planar, and biaxial data can be done.

29. For *MAT_GENERAL_HYPERELASTIC_RUBBER/*MAT_OGDEN_RUBBER, the volumetric response can be represented by compressibility constants D1, D2, and D3. Their reciprocals are used as coefficients in the polynomial expression of the volumetric strain energy density.

30. Added to *MAT_084 (*MAT_WINFRITH_CONCRETE) option RATE = 8 and related additional input fields, including stress-strain curves for post-peak responses in tension and compression (LCTST and LCCMP, respectively). See the manual for further details.

31. Enabled *MAT_091 and *MAT_092 for IGA solids.

32. For *MAT_102, output $Z$, the Zener-Holloman temperature compensated strain rate, as history variable #11 with the _THERMAL option or #8 without.

33. Support extended Cockcroft-Latham criterion for *MAT_107/*MAT_MODIFIED_JOHNSON_COOK for shells.

34. Allow the load curves, LCA, LCB, LCC, LCAB, LBC, and LCCA, on *MAT_126 to each be a table of strain-rate-dependent curves.

35. Added filtration factor, RFAC, to *MAT_126 to smooth the variation of strain rate.

36. Added optional Poisson effect for uncompacted status with field PRU to *MAT_126.

37. Added option to define failure as a function of strain ratio to *MAT_126. Input parameters TSEF < 0 and/or SSEF < 0 refer to curve IDs: failure strain vs. "ratio of compressive to tensile strain".

38. Improved strain-rate-dependent hardening *MAT_133 with respect to computational speed.

39. Using *MAT_135 with FLG = 0 now generates files for plotting yield information (Contour_1#, Contour_2#, Contour_3#, R_and_S#) similar to using FLG = 1.

40. Stabilized *MAT_GAS_MIXTURE during advection.

41. For *MAT_MUSCLE (*MAT_156), improved the critical time step estimation by including the stiffness of the active stress (SVR function).

42. Added EXTRA = 3 (strain-based Tsai-Wu failure criterion) and EXTRA = 4 (strain-based Tsai-Hill failure criterion) to *MAT_157.

43. Added new input field LCFIB to *MAT_172 (*MAT_CONCRETE_EC2) for defining tensile stress versus strain in a load curve. This feature is intended for steel-fiber-reinforced concrete.

44. Parameter IRATE of *CONTROL_IMPLICIT_DYNAMICS can be used in combination with *MAT_-SAMP-1 and *MAT_SAMP_LIGHT.

45. Added outputting history variable names for *MAT_187L to d3hsp.

46. Support IRATE from *CONTROL_IMPLICIT_DYNAMICS in *MAT_SAMP_LIGHT with RATEOP = 1.

47. Added check and warning to *MAT_187L (*MAT_SAMP_LIGHT). The shape of the uniaxial yield curves LCID-T and LCID-C can lead to problems for the plasticity algorithm due to initial softening.

48. *MAT_211 (*MAT_SPR_JLR) now works in implicit solutions.

49. *MAT_213 utilizes a damage sub-model to take the physical phenomenon of fracture and delamination into account. The damage sub-model in *MAT_213 is not temperature or strain-rate dependent. This enhancement adds strain-rate-dependent and/or temperature-dependent damage to *MAT_213.

50. *MAT_213 describes the nonlinear behavior with the help of stress-strain curves defined in different material directions. Before this enhancement, 12 curves define the plastic material behavior: three in tension, three in compression, three in shear, and three in the off-axis (45 degrees) direction. This enhancement enables only specifying nine curves with the assumption that the off-axis yield coefficients are zero. For this case, set the curve IDs for the off-axis curves as zero in the input. This enhancement is for elements that evaluate a 3D stress state.

51. Updated *MAT_213/*MAT_COMPOSITE_TABULATED_PLASTICITY_DAMAGE to version 1.3.6 for thick shell element formulations 3, 5, and 7.

52. The Tsai-Wu failure criterion in *MAT_213 (*MAT_COMPOSITE_TABULATED_PLASTICITY_-DAMAGE) is now available for both solid and shell elements. Previously it was only available for solid elements. Additionally, a linear stress degradation model was linked with the Tsai-Wu failure criterion.

53. Added option ERODE = 2 to *MAT_TABULATED_JOHNSON_COOK (*MAT_224), so there is no stress softening for adaptivity simulation.

54. Added six new history variables to *MAT_240 invoked by RFILTF ≠ 0. History variables 18 to 23 give the normal stress, tangential stress, normal force, tangential force, element area, and mode mixity.

55. Support IRATE from *CONTROL_IMPLICIT_DYNAMICS in *MAT_240.

56. For *MAT_251/*MAT_TAILORED_PROPERTIES, added scaling factor for Young's modulus $E$ on history variable #8. That scaling factor can be defined via *INITIAL_STRESS_SHELL/SOLID. A value of 1.0 means no scaling (default).

57. Added ability to locally overwrite/scale some material parameters for *MAT_252 based on history entries read from *INITIAL_STRESS_SOLID. New parameter IHIS determines the interpretation of the history data.

58. For shells, tshells, and solids, added the possibility to define a damage limit to *MAT_LAMINATED_- FRACTURE_DAIMLER_PINHO (*MAT_261) to allow for maintaining some limit stress after full damage. This can be defined with the parameters DAF, DKF, and DMF.

59. For *MAT_LAMINATED_FRACTURE_DAIMLER_CAMANHO (*MAT_262) with shell elements:

    ○ EA, EB, and GAB can be defined as a load curve (or table) giving the elastic stress vs. strain relationship. This definition enables taking into account nonlinear elastic behavior as well as tension and compression asymmetry.
    ○ Added defining a load curve or table with field LCSS to describe the nonlinear in-plane shear behavior. *MAT_261 has a similar feature.
    ○ Added crashfront parameter SOFTC to reduce the longitudinal compressive strength XC independently from SOFT.
    ○ Added failure strains EF_11T, EF_11C, EF_22T, EF_22C, EF_12, EF_23, and EF_31 to define orientation-dependent failure. These parameters provide more control compared to the combined value of EFS.
    ○ Added coupling factors CF12, CF23, and CF13 to tweak the damage behavior in the different shear planes

60. Support LCI in *MAT_264/*MAT_TABULATED_JOHNSON_COOK_ORTHO_PLASTCITY being a *DEFINE_TABLE_3D as in *MAT_224.

61. Enable *MAT_264 to be used with thick shell elements types 3, 5, and 7.

62. Activate *MAT_PAPER (*MAT_274) for IGA shells and IGA solids.

63. Added option, CTRACK, to activate experimental crack tracking algorithm to *MAT_280. A heuristic approach considers the crack direction in adjacent elements. This can help against mesh-dependent crack propagation (mesh bias). This feature does not yet fully work in MPP, so one glass layer part should go on one processor for now.

64. For *MAT_280 added outputting the global crack direction with history variables 15, 16, and 17 to enable visualizing the first cracks as vectors in LS-PrePost.

65. Enabled IGA solids for *MAT_295.

66. For *MAT_DMN_COMPOSITE_FRC/*MAT_303

    ○ Added options ITC = 2 and 3 for the prediction of strain rate-dependent and tension/compression asymmetric macroscopic behaviors of injection-molded short fiber-reinforced composites.
    ○ Offer a method to capture the effects of thermally/chemically-induced residual stress field, for which the Moldex3D injection molding simulation results can be imported by LS-PrePost.

67. Added CKOPT = 11 to *MAT_307 to use a model-free kinetics approach for curing.

68. Added parameter TZERO to *MAT_307 to define the temperature value of 0 K in the current temperature unit set. It serves as an offset to make it possible to use the Celsius scale for curing simulations.

69. Added new input options for the Prony series on *MAT_307. Instead of defining individual series for $G$ and $K$, it is now possible to define a series for $E$ or $G$ and a constant Poisson's ratio.

70. *MAT_318 / *MAT_TNM_POLYMER can now set the temperature, $\theta$, from thermal load cards.

71. Added support for *MAT_318/*MAT_TNM_POLYMER for solids with ELFORM = 15 and 16 (prisms and quadratic tetrahedrons)

72. Material constants MUA, KAPPA, MUBI, MUBF, and MUC on *MAT_318/*MAT_TNM_POLYMER may now depend on temperature in implicit.

73. Enabled *MAT_318 for thick shells with element formulations 3, 5, and 7.

74. Disable the bulk viscosity force when the material modeled with *EOS_IGNITION_AND_GROWTH_- OF_REACTION_IN_HE changes to gas

75. Improved the convergence of the *EOS_PROPELLANT_DEFLAGRATION / *EOS_010 algorithm, so that a solution is found even if certain material parameter combinations introduce strong nonlinearities.

### 7.2.24 Mesh Cards

1. Added ability to define mesh size shape based on the user node ID coordinates from the solid mechanics problem. See *MESH_SIZE_SHAPE.

2. Added an option to define an internal radius to be excluded from mesh size shape for SNAME = cylinder or sphere with METHOD = 1 on *MESH_SIZE_SHAPE.

3. Boundary layer generation criteria NELTH, BLTH, and BLFE on *MESH_BL can now be made functions of time (in remeshing cases) when a negative value is entered. NELTH and BLTH can also be functions of coordinates by using a *DEFINE_FUNCTION.

### 7.2.25 MPP

1. Relax criteria in MPP redecomposition (*CONTROL_MPP_DECOMPOSITION_REDECOMPOSI-TION) when the REMSPH option is on. LS-DYNA only prints a new connectivity file if over 10% of the total particles become inactive. Previously, a new connectivity file was written even if a single particle became deactivated.

2. Removed and consolidated MPP synchronization calls that will potentially shave off a few percent of simulation time. Also, added timings for the remaining synchronization calls for a better assessment of load balance.

### 7.2.26 Multiscale

1. Added new command line flag `nmsp` for running two-scale co-sim jobs (see *INCLUDE_MULTISCALE and *DEFINE_MULTISCALE) in a very similar way to running a single LS-DYNA MPP job. The command becomes `mpirun -np 96 mppdyna i=input.key nmsp=32`, where nmsp specifies the number of MPI processors for the local model.

2. Enable defining the loading path for each component, $H_{ij}$, of the prescribed macroscopic displacement gradient for *RVE_ANALYSIS_FEM. To do this, set LCID < 0 to indicate that each component is input as a load curve ID for a load curve giving the component as a function of time.

3. Support creating image RVEs in 2D RVE analysis (see *RVE_ANALYSIS_FEM).

4. Support inputting the path and name of the RVE mesh file in the MESHFILE field of *RVE_ANALY-SIS_FEM. Previously only the file name was allowed to be input, and the file was required to be in the same folder as the main input file. This change is useful for analyzing the same RVE geometry under various loading conditions and/or with different material properties.

### 7.2.27 Output

1. For shells, write the parameter NLOC from *SECTION_SHELL to hisnames.xml.

2. Consider OPTION4 of *DATABASE_ELOUT also for resultant truss elements to print their history variables to elout.

3. File dynain.lsda can now be written at given time intervals instead of only at end time. This feature was developed for restarts. Cards 3.1 and 3.2 on *INTERFACE_SPRINGBACK_LSDYNA enable this option.

4. Output regarding the use of IRATE on *CONTROL_IMPLICIT_DYNAMICS to screen and all messag files.

5. Added field NSKIP to *DATABASE_BINARY_RUNRSF/D3DUMP for MPP/HYBRID to reduce the frequency of producing full deck restart files. This option saves CPU time since processor 0 must collect all the information from the processors.

6. Added support for *MAT_031 when HISNOUT > 0 on *CONTROL_OUTPUT.

7. Added support for using *DEFINE_MATERIAL_HISTORIES to control output to d3part.

8. No longer include mass added from selective mass scaling to the added mass output to the d3plot database via MSSCL on *DATABASE_EXTENT_BINARY.

9. Output the area moments of inertia, $I_{xx}$, $I_{yy}$, and $I_{xy}$, and principal area moments of inertia for cross sections defined with *DATABASE_CROSS_SECTION_PLANE and *DATABASE_CROSS_SECTION_SET to d3hsp. These moments of inertia are about local axes of the cross section. For *DATABASE_CROSS_SECTION_PLANE, the z-axis of the local coordinate system aligns with the user-defined normal vector of the plane. The local x-axis and y-axis are then calculated based on this z-axis orientation. For *DATABASE_CROSS_SECTION_SET, the local coordinate system is established based on the the point cloud from the centers of the elements selected when defining the cross section. The z-axis is the normal vector and is determined by the component with the least variance in a principal component analysis of the point cloud. The x-axis and y-axis are then defined by the remaining two principal components.

10. Strain energy density for beams is output to d3plot and d3eigv, when asked for through ENGFLG on *DATABASE_EXTENT_BINARY.

11. Refined timings for element calculations to provide a better picture of where time is spent in the element routines.

12. Refined timings for Misc. timing slots for clarification of its content.

13. Enabled dumping d3max for sense switch sw1.

14. Added support for thick shell elements in d3max (*DATABASE_D3MAX).

15. Added support for small restart and simple restart for *DATABASE_D3MAX.

## 7.2.28 Sensors

1. Added SETOPT = NSN2NSMIND to *SENSOR_DEFINE_NODE to monitor the shortest distance between 2 sets of nodes.

2. Added options to *SENSOR_DEFINE_ELEMENT to trace hydrostatic (COMP = HYDR) and principal stress/strain (COMP = PRIN1, PRIN2, and PRIN3).

3. Added TYPE = CURVE to *SENSOR_CONTROL to activate/deactivate a time-dependent curve.

4. Added the option of I0 = ADDMASS to *SENSOR_DEFINE_MISC when MTYPE = MATSUM.

## 7.2.29 SPG (Smooth Particle Galerkin)

1. Implement transfer of internal variables for multi-stage SPG simulations by setting ISPGHIS = 1 on *INTERFACE_SPG_1 and *INTERFACE_SPG_2. The internal variables are data required by the material law other than the stresses and effective plastic strain, which are always included.

2. Optimized the spg particles to solid coupling algorithm (*DEFINE_SPG_TO_SURFACE_COUPLING). The optimized algorithm uses the initial time step for the stiffness calculation to avoid a sharp change in the stiffness, which can cause numerical instability.

3. Implemented an MPP version of *DEFINE_SPG_TO_SURFACE_COUPLING. Currently, it is in a trial stage.

## 7.2.30 SPH

1. Added support for an artificial compressibility term in Incompressible SPH. This term greatly alleviates spurious pressure oscillations when the time step varies. See ACMP on *CONTROL_SPH_INCOMPRESSIBLE.

2. Improvement to *DEFINE_SPH_INJECTION when the injection area is smaller than one particle size. In this case, we automatically space injection planes out to match the volume rate expected from the input.

3. Added support for IBUFF = 1 with IREACT = 0 in *DEFINE_BOX_SPH.

4. Added flag IRMV to Incompressible SPH (*CONTROL_SPH_INCOMPRESSIBLE) to detect initial

penetrations with the structure and remove penetrated fluid particles.

5. Added IFILE option to *DATABASE_ISPHHTC to create one isphhtc file per part.

6. Added temperature to SPH tracer output if applicable (*DATABASE_TRACER).

7. Added error when NEND is less than NID in *ELEMENT_SPH.

### 7.2.31 Thermal Solver

1. For MXDMP > 0 on *CONTROL_THERMAL_SOLVER:

   - Changed indfmt from 10I8 to 8I10 in the output for large problems.
   - Write indices along with right-hand side values in the output.

2. Made performance improvements for very large problems using SOLVER = 17 (GMRES) of *CONTROL_THERMAL_SOLVER.

3. Redefined the third field of Card 1 of *LOAD_HEAT_EXOTHERMIC_REACTION to refer to a solid element set instead of a node set. This change was needed so that the extra heat term from this load is taken into account by the nonlinear thermal solver. Previous versions required a small time step for accurate results, even when the nonlinear thermal solver was selected because the extra source term could only be added in after the thermal solve.

4. Added a thermal keyword option (THERMAL) to *CONTRAINED_NODAL_RIGID_BODY. With this option, either the first node in a node set can be constrained to the average temperature of the other nodes in the node set, or a specified node can be constrained to the average temperature of the nodes in the node set.

5. Allow for the combination of eroding structures with user-defined fluxes.

### 7.2.32 User-Defined Features

1. Added usermat helper routine `usr_get_hsv_beam` to get history variables for all integration points in a Hughes-Liu beam.

2. Added the possibility to define up to 8 history variables in user subroutine `userdewear`, called by setting WEARC < -100 on *DEFINE_DE_TO_SURFACE_COUPLING. The number of history variables is defined by $\min(|WEARC| - 100, 8)$.

3. Added new keyword options SP and DP to *MODULE_LOAD to allow loading different shared objects depending on the precision of the binary executable.

4. Added new user-defined feature to userdb.f90 to handle reading and writing user data between files and publicly accessible arrays.

5. Added output of the sharelib module file name. LS-DYNA prints it to d3hsp, messag, and screen.

### 7.2.33 Miscellaneous

1. General reductions in the required value of `memory=` on the command line.

2. Added a warning when a *END statement occurs in an encrypted file. If that file is read via a *INCLUDE statement, this would not cause a problem. But if it is included inline, the *END (which is not visible to the user) terminates the reading of the file, which causes difficult to diagnose problems if the user is not made aware of the *END statement.

3. Parameters TENSOR and THKSCL of *INCLUDE_STAMPED can be defined for keyword option MATRIX.

4. Added warning message when *PART_INERTIA is defined with zero mass to let the user know that all properties, including the center of gravity position, will be overwritten by LS-DYNA's internal calculation.

5. Enabled restarts to work with FMI (*COSIM_FMI).

6. Added command line option `shell=16to30/shell=30to16` for switching between shell formulations 16

and 30 without changing the input deck. LS-DYNA adjusts the cost of different formulations during decomposition for better parallel load balance.

7. Added command line option `nd3kil=ncyclek` to set the frequency of checking for the d3kil file. This option can speed up the calculation for HPC.

8. If multiple instances of *DEFINE_FRICTION have the same ID in an input deck, LS-DYNA uses the values from the first occurrence.

9. Added command line option `shell=16to30/shell=30to16` for switching between shell formulations 16 and 30 without changing the input deck. `LS-DYNA` adjusts the cost of different formulations during decomposition for better parallel load balance. Also, added a second method to switch between shell types 16 and 30 with ISHSHL on *CONTROL_SHELL.

10. Enable sensor switch `endtime` with the d3kil file.

11. Added field ISAVE to *DEFINE_PRESSURE_TUBE write shell/solid tubes to file for use in subsequent runs.

12. For *DAMPING_FREQUENCY_RANGE_DEFORM:

    ○ When combined with *CONTROL_PORE_FLUID, the damping now acts on pore pressure as well as on the effective stress. This can be switched off by new input parameter IPWP.
    ○ Optionally, a different damping ratio can be applied to the volumetric stress versus the deviatoric stress in solid elements. See new input parameter CDAMPV.

13. ELFORM = 13 works with *DAMPING_FREQUENCY_RANGE_DEFORM. (Caveat: The damping contribution to internal energy is not yet calculated, so the energy balance will appear wrong.)

14. Vectorized the rigid body kinematics code and optimized it with respect to computational speed.

15. Provide temperature data to *DEFINE_CURVE_FUNCTION in structure-only simulations. A *LOAD_-THERMAL option defines the temperature. Previously, LS-DYNA could only provide temperature data to *DEFINE_CURVE_FUNCTION for thermal or coupled thermal-structure simulations.

16. Extended list of function arguments for *DEFINE_CONNECTION_PROPERTIES with PRUL ≥ 2 or *DEFINE_MULTI_SHEET_CONNECTORS to include the Young's moduli of the connection partners (flanges). This extended list of arguments applies to the *DEFINE_FUNCTIONs that determine the material and failure properties.

17. Added support for *CONSTRAINED_SPR2 to *DEFINE_HAZ_PROPERTIES. Up to now, it was only working with *CONSTRAINED_INTERPOLATION_SPOTWELD (SPR3).

18. Added warning if a load curve (*DEFINE_CURVE) contains no data points.

# 7.3 Bug Fixes

## 7.3.1 Acoustics, NVH, and Frequency Domain

1. Fixed bug to properly read the *FREQUENCY_DOMAIN_SSD cards if using the long format input option `long=s`.

2. Fixed incorrect input processing of METHOD = 211 (variational indirect BEM with fast matrix assembly) of *FREQUENCY_DOMAIN_ACOUSTIC_BEM. The input processing phase ignored METHOD = 211 and used METHOD = 2 instead.

3. Corrected duplicate writing of the last state to the Ensight plot states output when MASEPLT = 4 on *CONTROL_ACOUSTIC_SPECTRAL.

4. The default value for the source strength Q on *LOAD_ACOUSTIC_SOURCE was mistakenly set to zero. This has been changed to 1.0 as shown in the manual.

5. When using *BOUNDARY_ACOUSTIC_COUPLING_MISMATCH in transient explicit solutions, the coupling initialization (search) parameters reset by the optional instruction *CONTROL_ACOUSTIC_-COUPLING were ignored. This has been corrected. Even when not using *CONTROL_ACOUSTIC_-COUPLING, the search process may be more restrictive than before since the default control

parameters (ACECFx) used by the explicit solution path have changed as a result.

6. For *FREQUENCY_DOMAIN_ACOUSTIC_BEM, fixed bug by adding the constant alpha as input for windowing function 7 (FFTWIN = 7 in Card 2), the Kaiser Window. Not having this constant as something to be defined in the input deck led to incorrect results since by default it is 0.

7. For exponential windowing (FFTWIN = 5) on *FREQUENCY_DOMAIN_ACOUSTIC_BEM, added the option to define different pre-window times for different panels by using a negative curve ID for T_HOLD. This is needed for the case with multiple impacts.

8. For *FREQUENCY_DOMAIN_SSD, enabled using scale factor SF on Card 7 for the load curves. The scale factor was previously ignored.

9. Fixed bug in running direct SSD (*FREQUENCY_DOMAIN_SSD_DIRECT) with VAD = 12, 13, and 14 (enforced velocity, acceleration, and displacement). The bug led to incorrect results.

10. For *FREQUENCY_DOMAIN_RANDOM_VIBRATION_FATIGUE, fixed bug in converting SPL (Sound Pressure Level) to PSD (Power Spectral Density) load for aeroacoustic waves. The constants in the conversion equation are dependent on the unit of pressure used in the model.

11. For *FREQUENCY_DOMAIN_RANDOM_VIBRATION, fixed bug in reading multiple PSD loading input files in MPP. They were being read incorrectly.

12. Fixed a bug in inverting the normal of boundary elements (NORM = 1 on *FREQUENCY_DOMAIN_A-COUSTIC_BEM in MPP. Before the fix, it was inverted twice.

13. Fixed a bug in using a single part as a load location for random pressure with *FREQUENCY_DO-MAIN_RANDOM_VIBRATION. The bug caused incorrect results.

14. Fixed a bug in adding the static response to response spectrum analysis results (BINARY = 3 in *DATA-BASE_FREQUENCY_BINARY_D3SPCM). Before the fix, it broke if the number of stress components from static and from response spectrum analysis (*FREQUENCY_DOMAIN_RESPONSE_SPEC-TRUM) did not match.

15. Fixed a bug in writing trimmed d3ssd for MPP. Before the fix, it worked for up to 12 cores only.

16. Fixed a bug with RESTRT = 4 on *FREQUENCY_DOMAIN_ACOUSTIC_BEM, running an acoustic computation on a boundary element mesh with velocity information given with a mismatching boundary element mesh.

17. Fixed a bug in writing d3ssd for *CONTROL_IMPLICIT_SSD_DIRECT. An incorrect nglbv parameter was used in writing this file in the past.

## 7.3.2 Adaptivity

1. Fixed a bug in a multi-stage 3D *r*-adaptive forging simulation using an lsda include file. The bug caused the second stage of the analysis to error terminate at the first adaptive step.

2. Fixed bug in data update routines for new segments generated by in-core adaptivity (INMEMORY = 1 on *CONTROL_ADAPTIVE). This bug may have caused errors in contact force calculations after adaptive cycles.

3. Fixed bug data synchronization at the initialization stage that affected MPP mesh fusion behavior defined in *CONTROL_ADAPTIVE.

4. Fixed a bug in synchronizing adaptive control parameters. The bug may cause processors to hang when INMEMORY in *CONTROL_ADAPTIVE is switched on.

5. Fixed outputting an incorrect warning message in the case that INMEMORY in *CONTROL_ADAPTIVE is switched on and IMSCL in *CONTROL_TIMESTEP is enabled.

6. Fixed a memory error that may be encountered in contact calculations when INMEMORY in *CON-TROL_ADAPTIVE is switched on.

7. Fixed minor bugs that caused incorrect contact data for new adapted elements when INMEMORY in *CONTROL_ADAPTIVE is switched on.

8. Fixed a bug in intfor file output. The bug may cause processors to hang when INMEMORY in *CON-

TROL_ADAPTIVE is switched on.

### 7.3.3 Airbags

1. Fixed a bug introduced in June of 2017 that led to an incorrect venting leakage when an airbag is defined by a *SET_SEGMENT. This bug affected *AIRBAG_HYBRID and *AIRGBAG_WANG_NEFSKE. Control volume airbags defined using *SET_PART work fine.

### 7.3.4 ALE

1. For *ALE_STRUCTURED_FSI, fixed missing coupling segments when defining the Lagrangian structure with solid/tshell parts that contain wedge/tetrahedron elements.

2. Fixed memory bug for S-ALE with *BOUNDARY_NON_REFLECTING that caused the boundary condition to be applied incorrectly to the segments.

3. Fixed bug for S-ALE mesh motion when using COVER_LAG with FRCPAD on *ALE_STRUCTURED_-MESH_MOTION. The default value of FRCPAD was wrongfully taken as 1.0.

4. Fixed bug in S-ALE mesh generation using a local coordinate system with *DEFINE_COORDINATE_-SYSTEM. The mesh was built wrong.

5. Fixed bug that caused the run to crash when using *ELEMENT_SHELL_OFFSET_COMPOSITE with S-ALE.

6. An S-ALE restart bug caused double precision runs to deviate from single precision ones.

7. Fixed crash in the keyword phase that occurred for *ALE_FSI_SWITCH_MMG when SID was a part set.

8. Fixed issue with the S-ALE 2D plane strain formulation when used in a mixed manner with Lagrange parts of a different formulation. For example, a 3D rigid plane. This issue caused the run to crash.

9. Fixed memory error for S-ALE mapping 3D to 3D with *INITIAL_ALE_MAPPING. The error occurs when the first run contains Lagrange parts.

10. Fixed bug in S-ALE FSI when coupled to a solid structure. The S-ALE mesh submerged under the Lagrangian solid parts is automatically constrained, meaning S-ALE materials inside the Lagrangian solid mesh will not move. Before the fix, these S-ALE cells were marked and frozen throughout the whole simulation. This method only works well if the structure does not move (or move too much). Now the cells are checked every cycle to see if they are submerged in the Lagrangian mesh.

11. Fixed bug in S-ALE velocity/displacement boundary conditions when controlled by *SENSOR. Previously, the check for birth/death time was not correctly called, so the sensor control did not work.

### 7.3.5 Boundary Cards

1. Fixed issue in *BOUNDARY_FLUX where more than eight history variables could not be read from the data cards.

2. Fixed incorrect transformation of BIRTH/DEATH on *BOUNDARY_SPC when included with *IN-CLUDE_TRANSFORM. FCTTIM was not applied. Output to dyna.inc was also missing.

### 7.3.6 CESE Compressible Fluid Solver

1. Made improvements to the chemically-reacting flow solver.

2. Fixed an internal read error for the situation where *CESE_INITIAL without the keyword options is not included in the input deck.

3. The 0-D inflator solvers are now available in the R15 MPP double precision version of LS-DYNA, as well as the R15 SMP double precision version of LS-DYNA. Previously they were not available to users.

### 7.3.7 Chemistry Cards

1. Fixed an array passing issue in the chemistry model setup phase.

### 7.3.8 Constraint Cards

1. *CONSTRAINED_JOINT_CONSTANT_VELOCITY behaves the same for the penalty and Lagrange multiplier formulations (LMF on *CONTROL_RIGID).

2. Fixed a bug for overriding constrained nodal rigid bodies (*CONSTRAINED_NODAL_RIGID_BODY_-OVERRIDE) that led to wrong results when *BOUNDARY_PRESCRIBED_MOTION_SET and *CON-STRAINED_NODAL_RIGID_BODY_OVERRIDE are in the same model.

3. Fixed an MPI bug that happened when only part of the model belonged to the overriding rigid body (*CONSTRAINED_NODAL_RIGID_BODY_OVERRIDE).

4. Fixed *CONSTRAINED_JOINT_RACK_AND_PINION which gave wrong results for implicit.

5. For *CONSTRAINED_NODAL_RIGID_BODY, nodal rotational velocities were not included when computing the initial conditions of the rigid body.

6. Fixed issue with *CONSTRAINED_SPOTWELD_FILTERED_FORCE. The behavior of the weld was wrong.

7. Fixed warning INI+485 to not apply to *CONSTRAINED_JOINT_TRANSLATIONAL_MOTOR since it does not require node 4.

8. Bug fix for the combination of *CONSTRAINED_NODE_SET with *DAMPING_PART_MASS affecting MPP solutions only. The symptom was that the constraint was not obeyed for all of the nodes in the set, and warning MPP+156 was issued.

9. Fixed bug for SPOTHIN on *CONTROL_CONTACT when applied to *CONSTRAINED_INTERPOLA-TION_SPOTWELD (SPR3). MPP synchronization was missing for scaled contact thickness.

10. Fix for internal energy computation in *CONSTRAINED_SPR2 to avoid nonsensical results when damage reaches 1.

11. Fix for SPR3 (*CONSTRAINED_INTERPOLATION_SPOTWELD) with MODEL > 10. Function IDs were not working with *INCLUDE_TRANSFORM and large IDs $> 2^{24}$ also failed. The same correction applies to *MAT_265 when using it with SPR3.

### 7.3.9 Contact

1. Modified the MPP treatment of the IPBACK option (optional Card E of *CONTACT_OPTION) so that each constraint/penalty contact pair is treated individually. Previously, if a tracked node was in more than one constraint contact, it might get tied in one constraint contact, and none of the backup penalty contacts would tie it. Now, it should get tied in every penalty contact whose constraint contact did not tie it.

2. For MPP contact, perform contact thickness update changes for shells on both cycle 0 and cycle 1 when THKCHG = 1 on *CONTROL_SHELL is enabled so that initial penetration results correctly account for shell thicknesses (ISTUPD > 0 on *CONTROL_SHELL) on cycle 0.

3. Improve the MPP contact failed element check to account for shell elements that failed at time 0.0.

4. MPP treatment of IPBACK on Optional Card E of *CONTACT could, in very rare cases, cause deadlock if the constrained tied contact was treated as non-groupable, but the created penalty side was groupable due to other contact options being set, and a processor dropped the constrained contact side due to having no tied nodes.

5. Fixed warning message for nodes not being tied in MPP for tied contact. In some cases where this message should have been written to the messag file:

   ```
   *** Warning 50129 (MPP+129)
       Tracked node is not constrained
       since it is not found on a segment.
   ```

   a message like this was being issued with garbage values for the connectivity:

```
*** Warning 50146 (MPP+146)
    segment contains rigid body nodes
```

6. Fixed issue in full deck restart of automatic tiebreak contacts (*CONTACT_AUTOMATIC_..._-TIEBREAK) in MPP. The restart was reinitializing (as untied) all nodes tied in the previous run, and thus losing all the surface stress and damage information.

7. In MPP, if a *CONTACT_AUTOMATIC_GENERAL was followed by a different contact type, a memory allocation was overwritten in some rare cases. This issue caused poor behavior of the edge contact treatment.

8. Fixed some minor contact issues in segment-to-segment contact (SOFT = 2) related to when SHLEDG and SRNDE conflict. This change causes the following behavior for SHLEDG and SRNDE:

   ○ If SHLEDG is set to 1 or 2, SRNDE is ignored.
   ○ If SHLEDG is set to 0, the SRNDE option is used.
   ○ If DEPTH = 23, 25, 35, or 55, both SHLEDG and SRNDE are ignored.

9. Fixed the edge-edge checking of the DEPTH = 5 and 15 options of segment-based (SOFT = 2) type contact. A failure to normalize a vector caused unit dependence of edge-edge checking.

10. Fixed the overlap area contact scaling option of segment-based (SOFT = 2) contact that is invoked by setting FNLSF > 0 and DNLSF = 0. The areas calculated were small, causing penetration.

11. Fixed a problem with *CONTACT_2D_AUTOMATIC when node sets are used, and MPP cores have no nodes in the set. The bug caused the job to hang.

12. Fixed a problem with MPP segment-based (SOFT = 2) contact when contact uses a birth time and also PSTIFF = 1. An error occurred when not all cores participated in the contact.

13. Fixed inconsistency when using *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_OFFSET with hybrid executables.

14. Fixed crash that could occasionally occur on MPP while reading the model if *CONTACT_TIED...OFFSET has nodes shared between SURFA and SURFB.

15. Reactivate the stiffness scale factor, SSF, on *PART_CONTACT for segment-based (SOFT = 2) contacts with DEPTH = 35. The scale factor has had no effect for a while (since R11.2.2).

16. Fixed SMP problem with *CONTACT_AUTOMATIC_..._TIEBREAK when using OPTION = 13 or 14. This bug could lead to errors (SOL+1345 or SOL+1346) if |ncpu| > 1.

## 7.3.10 Control Cards

1. Fixed calling erroneous keyword input error when activating NOCOPY on *CONTROL_SOLUTION.

2. Fixed inconsistency among hybrid runs that occurs when using a different number of threads for models in which mass scaling (*CONTROL_TIMESTEP) is enabled.

3. Fixed *CONTROL_PORE_FLUID combined with either *DAMPING_PART_STIFFNESS or *DAMPING_FREQUENCY_RANGE_DEFORM. The damping was ineffective in the parts containing pore pressure, even for shear stress components.

4. Fixed a long-standing bug for mass scaling of tetrahedral elements. The algorithm computed the added mass based on the current element mass instead of the original element mass using the initial volume. Comparison with hexahedral elements revealed that the mass scaling was incorrect, especially for large volume changes (like when foam gets compressed). In such cases, the added mass was too small during compression and too large during unloading.

## 7.3.11 Discrete Element Method

1. Fixed limitation that caused LS-DYNA to hang with greater than 50 *DEFINE_DE_BOND keywords.

## 7.3.12 Dual CESE Compressible Fluid Solver

1. The dual CESE immersed boundary method (IBM) FSI solvers allow for material erosion to occur in the

structural parts undergoing erosion or material failure. These solvers also allow for excluding some of the structural parts from the FSI interaction. This ability is desirable when a given part has little interaction with the flow field or when it is expedient for the sake of calculation speed to remove as many parts as possible from the FSI interaction. Until recently, if material erosion occurred in parts excluded from FSI interaction calculations, the calculation would halt or give erroneous results. With R15 (and also R14.1), the material erosion FSI calculation has been extended to take into account this rare situation. Another error was the incorrect updating of IBM-FSI information when a material erosion event occured. This has also been fixed in R15 and R14.1 but not in earlier versions of LS-DYNA. Please contact us if this is an issue for your use of the dual CESE solver. Another bug that was fixed involved IBM-FSI problems where the only material erosion that occurred was with thin shell elements.

2. Fixed many missing variable initializations that have not seemed to matter in the past. These initializations appeared to be correctly done for the Linux double precision version of LS-DYNA when built with the Intel compilers. However, these initialization fixes seem to correct dual CESE solver bugs in the Windows version of LS-DYNA.

3. Reinstated the calculation of the shared node information, and fixed several other bugs that were causing the moving mesh method FSI solvers to fail in several cases.

4. Corrected the 2D cyclic boundary condition. It did not work previously.

5. Fixed the d3plot output of these variables in the dual CESE IBM FSI solvers:

   - temperature
   - pressure
   - Schlieren number
   - Void fraction
   - Mach number

6. Corrected several problems in the 2D IBM FSI Navier-Stokes solver.

7. Fixed important bugs in the moving mesh FSI solvers, especially those related to shared nodes. Because of these bugs, the MPP version of these solvers likely did not work reliably in the past.

## 7.3.13 Elements

1. Fixed the second polar moment of area for ELFORM = 4 on *SECTION_BEAM. Before, this could lead to a torsional moment up to 10 times larger compared to ELFORM = 1 or 2.

2. Fix for ITORM on *SECTION_BEAM to properly work in MPP with more than one processor.

3. Fixed issue causing instability for *ELEMENT_BEARING. The original formulation of the bearing had an error in the local coordinate system transformations. We contacted one of the initial developers of the model, Kelly Carney, about the issue, and he contributed the new formulation.

4. Cohesive shell ELFORM = 29 now erodes when neighbors erode.

5. Fixed output thickness of cohesive shell ELFORM = 29 in d3plot.

6. Fixed issue where structural elements connected to cohesive shells were deleted shortly after initialization when using DELFR = 3 on *CONTROL_SHELL.

7. Inertia of discrete elements (ELFORM = 6 on *SECTION_BEAM) was not included in the total model inertia output to d3hsp.

8. For discrete elements (ELFORM = 6 on *SECTION_BEAM) with |SCOOR| = 12, extra rotational inertia to counteract decreased time step is no longer added (see remarks under *SECTION_BEAM).

9. Shell elements with thickness stretch (ELFORM = 25, 26, and 27 on *SECTION_SHELL) did not work in adaptive simulations. The thickness stretch was not properly passed to the new mesh.

10. Fixed *ELEMENT_MASS_PART with thick shell elements. It was calling an incompatible routine that caused an error termination.

11. Fixed an error in the MCID option of *ELEMENT_SHELL that led to calculating the wrong direction if any nonzero component of X0, Y0, or Z0 was in the plane of the element.

12. Fixed CNTCO of *CONTROL_SHELL for MPP solutions. This issue occurred when shell elements and solid elements shared nodes, and the shared nodes were along the decomposition boundary such that the solids element segments were on one processor and shell element segments on another. When the shell segments were offset, the shared node of the solid segment may not have gotten offset, causing inconsistent solutions with different numbers of processors.

13. Fixed incorrect velocity and displacement of an accelerometer to nodout when INTOPT = 1 in *ELEMENT_SEATBELT_ACCELEROMETER. This issue happened when the part on which the accelerometer is mounted is merged into another part.

14. Fixed bug that caused oscillating forces in attached cable elements. This bug applied to *ELEMENT_BEAM_PULLEY and *ELEMENT_BEAM_SOURCE.

15. The presence of *ELEMENT_BEAM_PULLEY in the model caused beam elements made of certain material types to disappear from the model on the first time step, irrespective of whether they were attached to pulley elements. Affected material types include 191, 209, and potentially others that refer to the initial length of the element.

16. Fixed bug for cases where initial forces in attached beam elements are not zero, such as due to *INITIAL_STRESS_BEAM or in multi-stage problems. The bug caused the beam element forces to be (wrongly) reset to zero.

17. Bug fix for internal energy calculation for thick shell ELFORM = 3, 5, and 7 when using *DAMPING_PART_STIFFNESS and *DAMPING_FREQUENCY_RANGE_DEFORM.

18. For solid element ELFORM = 99 on *SECTION_SOLID:

    - This element previously ignored OSU = 1 on *CONTROL_ACCURACY, but now it applies OSU.
    - This element previously did not work correctly with *DAMPING_FREQUENCY_RANGE_DEFORM.

19. Bug fix for discrete elements with Displacement Limit (CDL or TDL on *SECTION_DISCRETE) combined with an orientation vector (VID on *ELEMENT_DISCRETE). The bug could sometimes cause LS-DYNA to crash or terminate abnormally.

20. The combination of using *ELEMENT_T/SHELL_COMPOSITE and *CONSTRAINED_NODAL_RIGID_BODY may have introduced some incorrect evaluation of part sets and thus led to unexpected results.

21. Fixed a problem that may have appeared when a model contains a very large number of *ELEMENT_T/SHELL_COMPOSITE keywords while using *INITIAL_VELOCITY_GENERATION for these elements. Running such models may have led to unexpected errors due to very large part IDs in newly automatically generated parts within LS-DYNA.

22. Fixed bug in case STYP = 3 (part ID) or STYP = 4 (part set ID) was used in *DEFINE_ELEMENT_EROSION_T/SHELL for parts composed of composite thick shells or shells (*ELEMENT_T/SHELL_COMPOSITE).

23. Fix for I10 format input with *ELEMENT_SEATBELT. Parameter N4 was not interpreted correctly.

24. Fixed issue with mass scaling of cohesive solids (*SECTION_SOLID with ELFORM = 19 through 22). Inverted elements (negative volume) could lead to huge negative added mass. Replacing the current volume by the initial volume fixes this issue.

25. Fixed bug for *PART_STACKED_ELEMENTS with INPLCMP = 1. If new shells should be generated, errors could occur.

26. Consider correct slip in strain output for *ELEMENT_BEAM_PULLEY.

27. Fixed shell thickness update with PSTUPD on *CONTROL_SHELL. This feature has not been working properly for ELFORM = 16 since May 2021 (R14 releases).

28. Fixed issue with *DEFINE_SD_ORIENTATION. In large models (>16.8 million nodes), it was possible that the internal node IDs of NID1 and NID2 were the same, leading to error STR+509 (`orientation vector 8: nodes are coincident`).

## 7.3.14 EM (Electromagnetic Solver)

1. Fixed bug in EP fiber orientation creation (*EM_EP_CREATEFIBERORIENTATION) when four nodes of a tetrahedral element were in the same *SET_NODE defining fiber orientations.

2. Specifying a nonzero value in field seven (DEATHT) of *EM_MAT caused extra verifications and checks that added calculation time even when the deatht feature was never triggered. We have addressed this issue.

3. In cases with imposed potential, the algorithm sometimes did not attribute the inductance calculation in an em circuit to the correct circuit. This is now fixed.

4. Fixed erroneously outputting the joule power instead of joule heating energy in the EM energy output file.

5. Correctly take into account scale factors of *B-H* curves in *DEFINE_CURVE.

6. Fixed broadcasting of AREATYPE in *EM_RANDLES_EXOTHERMIC_REACTION to all CPUs, which could cause wrong results in MPP. Only CPU 0 had the value input for AREATYPE in the keyword deck. All other CPUs had the default value.

7. Fixed incorrect joule heating term calculation when conductive 3D shells were present in inductive heating problems.

8. Fixed incorrect *EM_MAT_003 orientation issue that could cause a small problem in the electric field calculation.

### 7.3.15 Fatigue Cards

1. Fixed bug in the fatigue solver accessing single precision binary databases. It could not read the initial damage ratio from d3plot or d3ftg database.

### 7.3.16 Forming Analysis

1. For *INCLUDE_STAMPED_PART, defining field IAFTER and giving a jobid is now supported.

2. Fixed a memory clobber that caused metal-forming problems that used trimming or lancing to error terminate.

### 7.3.17 ICFD (Incompressible Fluid Solver)

1. The introduction of *DEFINE_FUNCTION support in *ICFD_CONTROL_IMPOSED_MOVE introduced some issues when VADT or VADR is set to 1. We have made some fixes here, but we recommend using VADT = 0 and VADR = 0 whenever possible.

2. Solved the issue of an unstable pressure field in the case of 2-way FSI with the Windkessel outlet boundary condition (*ICFD_BOUNDARY_WINDKESSEL).

### 7.3.18 Implicit (Mechanical) Solver

1. Fixed buckling analysis with *CONTROL_IMPLICIT_BUCKLE for linear beam type 13. Before, the resulting buckling loads and mode shapes were irrational.

2. The consistent mass option (IFLAG = 1 on *CONTROL_IMPLICIT_CONSISTENT_MASS) now works correctly with shell type 21.

3. For LSOLVR = 2 with LCPACK = 3 on *CONTROL_IMPLICIT_SOLVER (direct solver for nonsymmetric problems):
   - The fill-reducing method set using *CONTROL_IMPLICIT_ORDERING (Card 1, field 4) or *CONTROL_IMPLICIT_ORDERING (Card 1, field 1) was ignored; it is now taken into account.
   - The "NMETIS" parameter (*CONTROL_IMPLICIT_ORDERING, Card 1, field 2) was ignored; it is now taken into account.

4. For EIGMTH = 102 *CONTROL_IMPLICIT_EIGENVALUE (LOBPCG):
   - The fill-reducing method set using *CONTROL_IMPLICIT_ORDERING (Card 1, field 4) or *CONTROL_IMPLICIT_ORDERING (Card 1, field 1) was ignored; it is now taken into account.

- Fixed bug for very large problems (#dofs × #eigenmodes > 250e6) in MPP. Convergence could be slow or even lost in some configurations.
- Fixed a memory leak.
- Fixed some reproducibility issues.
- Fixed incorrect output message (negative number of iterations) in some configurations.

5. For MTXDMP > 0 on *CONTROL_IMPLICIT_SOLVER, mark nonsymmetric matrices with "U" instead of "S" in the mtxtype field of the output.

6. Made a clearer error message for executables not compatible with LOBPCG (EIGMTH = 102 on *CONTROL_IMPLICIT_EIGENVALUE).

7. For EIGMTH = 102 (LOBPCG) on *CONTROL_IMPLICIT_EIGENVALUE, the d3eigv files were not closed properly, leading to an error termination when using the files with other keywords, such as *FREQUENCY_DOMAIN_SSD.

8. Fixed issue with *CONTOL_IMPLICIT_EIGENVALUE for MSTRES = 1 where stresses were not output correctly in d3eigv for anisotropic materials.

9. Fix for the d3eigv database. For *ELEMENT_TSHELL_COMPOSITE and *ELEMENT_SHELL_COMPOSITE, the parts in the d3iegv file referenced internally created parts IDs and not the original user part IDs.

10. Fixed error where intermittent eigenvalue extraction disabled mass scaling (DT2MS ≠ 0 on *CONTROL_TIMESTEP).

11. Made original (unscaled) mass default in intermittent eigenvalue extraction. Prior to this release, the scaled mass was incorrectly used.

12. Fixed error where eigenmode stress output could lead to a floating point exception upon termination.

13. Corrected the processing of constraint modes (*CONTROL_IMPLICIT_MODES) for use with implicit modal dynamics (*CONTROL_IMPLICIT_MODAL_DYNAMICS) using prescribed motion constraints (*BOUNDARY_PRESCRIBED_MOTION).

14. Corrected an error when including consistent mass terms (*CONTROL_IMPLICIT_CONSISTENT_MASS) in the mass matrix for IGA.

15. Made corrections to MPP Lanczos (*CONTROL_IMPLICIT_EIGENSOLVER) to fix the factorization restore feature. I/O files were mismatched between save and restore.

16. Fixed issue in the MPP Lanczos eigensolver that causes the initial shift to be 1.0. This shift is bad for models with singular stiffness matrices.

17. Made two corrections to output for *CONTROL_IMPLICIT_MODES.

- Reset the number of eigenvalues output in the case of fewer computed than requested so that the output matches the number computed.
- Ensure the output or not of stress is consistent to keep the d3mode file correct.

18. Made corrections to *CONTROL_IMPLICIT_MODES and *CONTROL_IMPLICIT_MODAL_DYNAMICS for processing of solid element only models. The issue led to a corrupted d3mode. Also, some conflicts with stress specifications occurred.

19. Enhanced *CONTROL_IMPLICIT_MODAL_DYNAMICS to work for a small solid element only model with prescribed motion constraints.

20. Corrected indexing error while loading masses for *CONTROL_IMPLICIT_EIGENVALUE. The issue could have led to skipping some nodal rotational inertias.

### 7.3.19 Initial Cards

1. Fixed a bug introduced in January 2021 with *INITIAL_FOAM_REFERENCE_GEOMETRY. It resulted in wrong results for type 10 solid elements.

2. Fixed issue in *INITIAL_STRESS_SECTION where implicit gave wrong internal energy for IZSHEAR = 0.

3. Fixed issue in *INITIAL_STRESS_SECTION where the force dropped at the start of an implicit simulation after explicit dynamic relaxation.

## 7.3.20 Isogeometric Analysis (IGA)

1. Fixed a bug in which mode stress was not visualized for IGA models using *IGA keywords.

2. For trimmed IGA solids (*IGA_SOLID), fixed a bug in the integration weights for the $s$ and $t$-directions. They should be set to unity.

3. Corrected some mass and inertia output in d3hsp, glstat, and massout for IGA shells (*IGA_SHELL).

## 7.3.21 Load Cards

1. Fixed a bug in *LOAD_SEGMENT that happened when the segment has 6 or 8 nodes. The bug could lead to erroneous results, especially in implicit.

## 7.3.22 Materials and Equations-of-State

1. Corrected outputting erroneous error for *MAT_ADD_COHESIVE when using the TITLE keyword option.

2. Fix for large table IDs ( > $2^{24}$) for Q1 < 0 on *MAT_ADD_DAMAGE_DIEM when using a single precision executable.

3. Fix for the combination of *MAT_ADD_EROSION and *MAT_ADD_DAMAGE_DIEM. In some instances, a criterion from *MAT_ADD_EROSION did not trigger failure because *MAT_ADD_DAMAGE_DIEM overwrote parameter NCS (number of failure conditions to satisfy) with the strain increment DEPS.

4. Fix for *MAT_ADD_DAMAGE_DIEM. Do not extrapolate in strain rate tables (2D) given by P1.

5. Fix for *MAT_ADD_DAMAGE_GISSMO. Damage is driven by plastic strain (DTYP = 0 or 1) or a history variable (DTYP ≥ 10). If that history variable is not monotonically increasing (plastic strain always is), damage could grow even below the already reached maximum value. This behavior is now prevented by remembering that maximum value. See also IFLG4 = 1 on *MAT_ADD_GENERALIZED_DAMAGE.

6. Fix for *MAT_ADD_DAMAGE_GISSMO when used in a full deck restart. Some variables (element size, regularization factor, fading exponent) were not correctly initialized for new parts.

7. Fixed the time step limit option, DTMIN, of *MAT_ADD_EROSION when it is used with thick shell forms 1, 2, or 6. The time step calculation used for this option was wrong, so it did not work as expected.

8. Fixed issue with the combined usage of *MAT_ADD_THERMAL_EXPANSION with *MAT_002 for shell and thick shell elements. Before, the expansion in the shell plane was missing.

9. Fixed issue for anisotropic solid materials with AOPT = 1 or 4 when used *INITIAL_FOAM_REFERENCE_GEOMETRY. To compute the deformation gradient more accurately, element nodal coordinates were moved to the origin. But this does not work with AOPT=1/4, where an originating point P is defined.

10. Fixed issue with *MAT_024_LOG_INTERPOLATION with VP = 1 (table) for shell elements when using the implicit solver. The algorithm used the same interpolation as in the negative first strain rate approach, but here we have "regular" strain rates with a logarithmic interpolation.

11. Fixed issue for *MAT_024_STOCHASTIC with VP = 1, LCSS being a table, and shell elements. Scaling of the yield stress could have been wrong.

12. Fix for *MAT_024 with MITER = 2 on *CONTROL_SHELL. Yield stress on history variable #5 could be zero if SIGY was not defined. That issue could lead to NaNs in the constitutive matrix for implicit.

13. Fixed reading of Chaboche-Roussilier parameters in *MAT_036, which has been broken for a while.

14. Fix for *MAT_036 with table LCID > 0 and HR = 3 (strain rates) or HR = 8 (temperature). LS-DYNA did not consider FCTTIM and FCTTEM from *INCLUDE_TRANSFORM for the unit conversion.

15. Fixed bug for the combination of *MAT_USER_DEFINED_MATERIAL_MODELS with IORTHO = 1 and shell types 23/24. The number of integration points was incorrect for orthotropic user materials with these shells, leading to an error.

16. For *MAT_LAMINATED_COMPOSITE_FABRIC (*MAT_058), fixed bug in table evaluation of EA, EB, or GAB < 0 for shell elements. Prior to this fix, the table may not have been evaluated correctly by not using the correct load curve associated with a specific strain rate.

17. For shell elements using *MAT_LAMINATED_COMPOSITE_FABRIC, (*MAT_058), use initial, undamaged stiffness for the computation of sound speed/time step.

18. Fixed problem with wrong initial temperature for *MAT_065 and *MAT_088 in combination with solid element type 2.

19. Fixed transformation of curve values for curves input on *MAT_074 when transformed with *IN-CLUDE_TRANSFORM. Curve values for FLCID, HLCID, and GLCID were not correctly transformed.

20. Fixed inconsistency among hybrid runs that occurs when using a different number of threads for models that contain *MAT_GENERAL_VISCOELASTIC (*MAT_076) when WLF is used and an additional Prony series is not used. This fix is for solids and shells.

21. For *MAT_079 (*MAT_HYSTERETIC_SOIL), fixed a bug affecting non-masing damping (input field LCD) when more than 10 points are defined in the shear stress-strain curve. This bug caused the requested damping in LCD not to have been applied correctly. It was also causing the omission of warning message STR+1397 on some operating systems.

22. For *MAT_079 (*MAT_HYSTERETIC_SOIL), the strain rate effect (input field LCSR giving enhancement factor versus strain rate) worked only for a limited range of curve shapes. It worked okay for curves with a constant gradient but not for some other shapes. Symptoms of the bug could include either LCSR being completely ineffective or a noisy response with stresses sometimes dropping back to the non-rate-enhanced level. Furthermore, the error message SOL+1082 was tripped unnecessarily if the first strain-rate point in the curve was input as zero. Zero is now allowed.

23. Fixed the RCDC option in *MAT_082 for solid pentahedral elements. A vital parameter was not computed.

24. Revised combination of *MAT_083_LOG_LOG_INTERPOLATION and implicit accuracy flag IACC = 1 (*CONTROL_ACCURACY). The logarithmic interpolation was automatically deactivated. It is now enabled again.

25. For *MAT_GENERAL_JOINT_DISCRETE_BEAM, fixed the positioning of the history variables for neighboring parts and failure. These could become offset, such as when using *DEFINE_MATERIAL_-HISTORIES.

26. Fix for *MAT_098 solids initialized by dynain (*INITIAL_STRESS_SOLID). They were not correctly initialized.

27. Fixed an error that affected solid spot weld failure options OPT = 0 and -1 on *MAT_100 when nonzero shear terms were input. The strength values were permuted, causing failure at the wrong weld force. This error first appeared in release version R11.0. This fix is backported to all versions from R12.2 and later.

28. Fix *MAT_INV_HYPERBOLIC_SIN_THERMAL for implicit that prevented the implicit solver from converging.

29. Fixed issue with combination of *PART_ANNEAL and *MAT_102_THERMAL. Too many history variables were set to zero. For instance, this issue could lead to a spurious stiffness jump and hence incorrect stresses.

30. Resolved inconsistency problem for hybrid when using *MAT_119 with IFLAG = 2.

31. Fixed issue causing RTCL damage to not work for *MAT_123/*MAT_MODIFIED_PIECEWISE_LIN-EAR_PLASTICITY.

32. Fix for the internal energy computation in *MAT_123 with VP = 1 and tetrahedral solids of ELFORM 13. Energy could have been wrong (e.g., negative) if all elements in a vector block were elastic.

33. Fixed issue for several materials if a huge number of curves (> 9999) is used in a model. This bug could

lead to an error or the material using the wrong curve. This bug affects *MAT_124, *MAT_136, *MAT_217, *MAT_218, *MAT_244, *MAT_254, *MAT_267, *MAT_270, *MAT_271, *MAT_278, *MAT_296, and *MAT_318.

34. Fix for GISSMO with *MAT_133. GISSMO overwrote memory used for the implicit tangent.

35. Fix for unit conversion of table values (strain rates) in *MAT_133. Table HARD < 0 should take FCTTIM from *INCLUDE_TRANSFORM into account.

36. Fix for *MAT_133 when used with plane strain or axisymmetric elements (ELFORM = 13, 14, or 15 on *SECTION_SHELL). This combination did not work before. The stresses were most likely zero all the time.

37. Bug fixes for *MAT_135 (*MAT_WTM_STM):

    ○ SIGMA0 < 0 (yield curve) did not work for WTM (FLG = 2).
    ○ Option NUMFI was not working as expected.
    ○ Put failure mode on history variable #7 as promised in the manual.

38. When using IHIS on *MAT_157 (*MAT_ANISOTROPIC_ELASTIC_PLASTIC for the initialization of stiffness and/or material orientations (q1/q2) together with laminated shell theory turned on (LAMSHT=3/5), the stiffness values where not properly initialized. This bug could have led to the wrong stiffness computation, affecting the contact behavior and the overall response. This problem applied to shells and solids.

39. For solid elements with *MAT_ANISOTROPIC_ELASTIC_PLASTIC (*MAT_157), when using IHIS to initialize the stiffnesses, the corresponding hourglass stiffness computation was wrong.

40. Fixed issue with *MAT_159 when used with thick shell types 3, 5, and 7. Initialization of history variables was missing.

41. Fixed improper Poisson's ratio checks of materials 161, 162, and 219. Some valid materials error terminated unnecessarily.

42. Fixed incorrect error, KEY+315, for *MAT_ARUP_ADESIVE (*MAT_169), when SHRP, the shear plateau ratio, is set to 0.9.

43. Allow large function IDs ( $> 2^{24}$ ) for some *MAT_169 entries using a single precision executable. This bug affects negative input of TENMAX, GCTEN, SHRMAX, GCSHR, SHRP, SDFAC, and SGFAC.

44. Fixed bug in *MAT_172 affecting thick shells (*ELEMENT_TSHELL) only. Certain functions requiring knowledge of element size did not work correctly:

    ○ If CHARL > 0, the post-crack tensile stress did not decay correctly with crack opening. The response could even be perfectly plastic with no softening.
    ○ If AGGSZ > 0 and CHARL = 0, the maximum shear stress transferred across open cracks did not follow the expected relationship with crack opening.

45. The *MAT_172 erosion criterion for reinforcement worked differently than users expected: reinforcement in both directions had to reach the erosion limit before erosion occurred. This criterion has now been changed such that reaching the erosion limit in either of the two reinforcement directions causes erosion, which is what users expected.

46. *MAT_172 did not work with *DEFINE_ELEMENT_EROSION_SHELL.

47. For *MAT_172 (*MAT_CONCRETE_EC2), if LCHAR and AGGSZ are nonzero, the calculation of maximum shear stress across an open crack used the wrong crack width. The "w" was wrongly scaled by LCHAR/(element size).

48. Fix for bug when using implicit with the combination of *MAT_SAMP_LIGHT (*MAT_187L) and GISSMO. The history variables clashed.

49. *MAT_197 (*MAT_SEISMIC_ISOLATOR) was generating spurious moments on node 2 of the discrete beam. This bug existed in R12, R13, and R14.0. A workaround was to apply rotational SPCs to node 2 of the isolator, but this method is only recommended if the new SPCs would not modify the behavior of the structure.

50. Fixed bug in *MAT_205 (*MAT_DISCRETE_BEAM_POINT_CONTACT). When the curve LCZ is used

and a gap of more than 1.0 exists between the *x*-axis values of consecutive points in the curve, the force did not follow LCZ correctly.

51. Fixed bug causing a segmentation fault for *MAT_213/*MAT_COMPOSITE_TABULATED_PLASTIC-ITY_DAMAGE when used for thick shell element formulations 1, 2, and 6.

52. Fixed bug in *MAT_213 that occurred in SMP or hybrid for more than one thread. The bug led to inconsistencies.

53. For *MAT_215, using a 3D table for LCI could lead to incorrect results if the triaxiality table started with a negative value.

54. Fixed bug for *MAT_224 and *MAT_224_GYS. If LCI refers to a *DEFINE_TABLE_3D, results could have been wrong if the triaxiality table started with a negative value.

55. Fix for *MAT_224 when used in a full deck restart. The bug fix corrected the initialization of temperature and element size for new parts.

56. For *MAT_248, allow real-type input values on Cards 10 and 11. Real-type led to an error while integer-type did not, and these fields are for real-type values.

57. Fixed bug for *MAT_254 when used with axisymmetric solids. The bug could result in segmentation violations.

58. Determine current E/PR at current temperature for *MAT_255. They were determined at the center temperature, but that did not match with the computation of the average value and the increment.

59. Fixed thick shells with material 261 when used with damage.

60. Fixed the computation of the characteristic element length for thick shells when used together with *MAT_261 or *MAT_262.

61. For *MAT_LAMINATED_FRACTURED_DAIMLER_CAMANHO (*MAT_262), fixed bug in table evaluation of EA, EB, or GAB < 0 for shell elements. Prior to this fix, the table may not have been evaluated correctly by not using the correct load curve associated with a specific strain rate.

62. For shell elements using *MAT_LAMINATED_FRACTURED_DAIMLER_CAMANHO (*MAT_262), use initial, undamaged stiffness for the computation of sound speed/time step.

63. Fixed type 6 and 7 hourglass control with *MAT_264 (*MAT_TABULATED_JOHNSON_COOK_-ORTHO_PLASTICITY) and *MAT_224_GYS (*MAT_TABULATED_JOHNSON_COOK_GYS). The material constants were wrong, causing the unpredictable hourglass behavior.

64. Fixed incorrect tangent for *MAT_267.

65. Fixed issue in *MAT_PAPER causing AOPT < 0 not to activate.

66. For *MAT_PAPER, fixed issue causing a segmentation fault due to unphysical strain increments for shells.

67. Fixed issue for *MAT_280 with RATENL > 0 or FT < 0 to avoid negative strain rates.

68. Fixed issue with *MAT_280 when used for thick shell elements. Sudden crack propagation through the thickness with option NIPF = 1 did not work as intended.

69. For *MAT_295, fixed (re)activation/deactivation of active models ACTYPE = 2, 3, and 4.

70. For *MAT_295:

    - Fixed electrophysiology-solid mechanics coupling when a different time step size is invoked.
    - Based (re)activation and deactivation on the calcium concentration at the current time step.

71. Fixed an issue that can show up when multiple materials and multiple *MAT_295 definitions appear in one input. The issue caused the job to crash.

72. Fixed issue in *MAT_HOT_PLATE_ROLLING/*MAT_305 where the residual convergence check led to premature termination.

73. Fixed convergence issues where *MAT_HOT_PLATE_ROLLING/*MAT_305 failed to converge for single precision.

74. Fixed d3hsp echo of material properties for *MAT_SPH_INCOMPRESSIBLE_FLUID and *MAT_-SPH_INCOMPRESSIBLE_STRUCTURE. The format was incorrect.

75. Fixed potential infinite loop when using *EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE / *EOS_007 and decreased the tolerance for double precision in the root-finding algorithm.

76. Fixed issue where materials with Poisson's ratio defined by a load curve may give erroneous results.

### 7.3.23 MPP

1. Fixed MPP decomposition issue that prevented any model that contained both SPH and IGA elements from running.

2. Fixed potential memory overwrite error in MPP processing of beam-beam contact, as used in *CON-TACT_AUTOMATIC_GENERAL.

3. Fixed ordering issue that could cause MPP processing failures for models with a mix of *CON-STRAINED_INTERPOLATION and *CONSTRAINED_INTERPOLATION_LOCAL.

4. Fixed issue with MPP support of multiple *ELEMENT_SHELL_SOURCE_SINK. Improper reporting of memory allocation errors occurred in some cases.

### 7.3.24 Multiscale

1. Fixed a bug in reading the node set defined for the local model in two-scale co-simulation jobs (*IN-CLUDE_COSIM).

2. Fixed a bug when calculating homogenized stress output to rveout for a 2D RVE (*RVE_ANALYSIS_-FEM) by considering the physical thickness of shell elements. Previously, the thickness of shell elements was assumed to be 1.0.

### 7.3.25 Output

1. Fixed behavior of *DATABASE_BNDOUT which has not been properly honoring the OPTION2-OPTION4 flags.

2. Fixed MPP output of the d3part file. It was wrong if DCOMP on *DATABASE_EXTENT_BINARY was input as < 2 for a model with rigid parts.

3. Fixed a bug for nodal stress/strain output of 20-node hexahedral elements that occurred when NODOUT = STRESS_GL, STRAIN_GL or ALL_GL on *DATABASE_EXTENT_BINARY. The bug led to a memory issue and crashed the job.

4. Fixed output error of jntforc to the binout file.

5. Fixed issue in *INTERFACE_SPRINGBACK for RFLAG = 1 that caused *REFERENCE_GEOMETRY not to be correctly written to dynain_geo for MPP.

6. Fixed the following bugs for *DATABASE_EXTENT_BINARY:

   - For integrated user solid elements (NIP > 0 on *SECTION_SOLID), the output integration point values were averaged, regardless of the value of NINTSLD. Now, when NINTSLD = 8, up to eight individual values are output.
   - Keyword option COMP with IXYZ = 0 did not work for MPP eigenvalue analysis.

7. Option to update reference nodes for beams (NREFUP = 1 on *CONTROL_OUTPUT) did not work in adaptive simulations.

8. Fixed bug associated with *DATABASE_RBDOUT. Displacements were not being output.

9. Fixed incorrect moments/inertia/area output to secforc. It was broken in December of 2018.

10. Fixed the computation of velocities in nodout when using REF = 2 in *DATABASE_HISTORY_NODE_-LOCAL. The velocities were incorrect when the moment arm, the distance between the node and the local coordinate system, is zero.

11. Fixed error in output files when QUADSLD and CUBSLD of *DATABASE_EXTENT_BINARY are greater than 0. This bug affects MPP only. It corrupted the output files, leading LS-PrePost to crash.

12. Fixed bug affecting the d3thdt file when NINTSLD = 8 on *DATABASE_EXTENT_BINARY. The strain output for solids consisted of the element-averaged value repeated eight times instead of separate strains for each integration point.

13. This bug manifested when using a dynain file written by LS-DYNA from a previous analysis as part of the input for a new analysis if the models contained certain types of tied contact that caused LS-DYNA to write *INITIAL_TIED_CONTACT_DATA into the dynain file. A crash could sometimes occur when starting the new analysis if LS-DYNA detected nodes that could not be tied because they are too far from a segment.

14. If the analysis starts from a nonzero time (e.g., in a staged construction analysis), the contact force file output by including *DATABASE_BINARY_INTFOR contained a very large number of states instead of what the user requested.

15. If a dynain file is written (e.g., due to *INTERFACE_SPRINGBACK or *CONTROL_STAGED_CON-STRUCTION), the topology of beam elements attached to the pulley was written incorrectly to the dynain file.

16. Corrected output to d3eigv for SMP for large problems, making it the same as MPP. The record size was incorrect, corrupting the states so that LS-PrePost could not properly process the data.

17. Fix contact force for 2D mortar contact (output in rcforc). The output force was always zero.

18. The strains for trusses are now output to the d3plot database.

19. Fixed a bug in appending d3max data to the d3part database (OUTPUT = 1 on *DATABASE_D3MAX). Before the fix, it failed if the header data in d3plot and d3part did not match.

20. Fixed a bug in computing principal stress and strain in d3max. Now the equations for computing these values match those from LS-PrePost.

## 7.3.26 Restarts

1. Dump and restore the local variable data for *DEFINE_FUNCTIONs so that restarts involving them will give the same results as if there was no restart. This currently only works for simple variables in the functions, not arrays.

2. Fixed inconsistent internal energy in a remap restart run with `q=remap` on the command line.

3. Fixed a bug causing the contact bucket sort to take longer than it should after a full deck restart.

## 7.3.27 Sensors

1. For *SENSOR_CONTROL:

   - Fixed a bug for type 2 and 3 pretensioners that happened when TYPE = BELTPRET. The bug was introduced in November 2019 and could lead to an unexpected status change of the pretensioner.
   - Fixed a bug for TYPE = CONTACT for thermal-only simulation. The bug caused LS-DYNA to fail to turn off the contact as designed.

2. Fixed a bug in TYPE = PRESC-MOT on *SENSOR_CONTROL for *BOUNDARY_PRESCRIBED_MO-TION_RIGID entities. The bug prevented the sensor from controlling the entity. This issue only occurred in implicit.

3. Fixed a bug for *SENSOR_DEFINE_ELEMENT that resulted in a segmentation fault when ETYPE = TSHELL.

## 7.3.28 SPG (Smooth Particle Galerkin)

1. Fixed the negative temperature bug in SPG flow drilling by setting the minimum temperature to be the minimum initial temperature.

## 7.3.29 SPH

1. Fixed bug that occurred when *DEFINE_BOX_SPH with IBUFF = 1 is used in conjunction with *DE-FINE_SPH_ACTIVE_REGION also with IBUFF = 1. The buffer zone associated with *DEFINE_BOX_-SPH was overwritten to fully active by *DEFINE_SPH_ACTIVE_REGION.

2. Fixed issue when an SPH part uses *EOS_007 or *EOS_010. History variables were incorrectly initialized.

3. Forces recorded with an *INTERFACE_COMPONENT_SPH keyword on a rotating local coordinate system were incorrectly rotated back on the global coordinate system when read in by the *INTER-FACE_LINKING_FORCES keyword.

4. Fixed bug in SPH causing an inconsistency for hybrid when updating the forces.

5. Fixed bug where incorrect coordinates would be written to file for some SPH tracers (*DATABASE_-TRACER).

6. Corrected an arithmetic error when shifting particles and a memory issue that would lead to a segmentation fault in certain cases. These bugs only affect SPH simulations for which ISHIFT = 1 in *CONTROL_SPH.

7. Fixed a group of bugs that would cause segmentation faults in some cases where *CONTROL_MPP_DE-COMPOSITION_REDECOMPOSITION and *DEFINE_SPH_INJECTION were used together.

8. Corrected the motion of ISPH structure particles tied to a rigid body with *DEFINE_SPH_MESH_OBJ when using a local coordinate system.

## 7.3.30 Thermal Solver

1. For MXDMP > 0 on *CONTROL_THERMAL_SOLVER:

   - Mark nonsymmetric matrices with "U" instead of "S" in the mtxtype field of the output.
   - Write the right-hand side in MPP.
   - Fixed incorrect ptrcrd calculation.

2. Fixed bug in SOLVER = 17 (GMRES) on *CONTROL_THERMAL_SOLVER for single precision executables. This method would give (easily identifiable) incorrect results.

3. Fixed error causing the iterative solver to use obsolete CGTOL if defined instead of the default value of RELTOL when reltol is left blank in *CONTRL_THERMAL_SOLVER.

4. Fixed issue where structural history variables could not be read from thermal materials for tetrahedrons and pentahedrons.

5. Fixed bug related to thermal for TIED_WELD contacts in MPP. When contact segments are on different processors, nodal flags (tied condition fulfilled or not) must be communicated. Before this fix, the switch to a different heat transfer coefficient for the welded region sometimes did not work.

6. Corrected handling of thermal boundary conditions (convection and radiation) in the case of adaptivity. Before this fix, refining a segment with the boundary condition applied could result in an unphysical temperature field.

7. Fixed an inconsistency in the birth/death times for temperature boundary conditions when using the thermal solver (*BOUNDARY_TEMPERATURE). LS-DYNA evaluates the birth/death condition at the end time of the time step, but before the fix, the time integration parameter (TIP on *CONTROL_-THERMAL_SOLVER) erroneously affected the condition.

8. Fixed bug in full deck restart with the thermal solver. In cases where the load curves (for dt, dtmax, dtmin, or tmax) control the time stepping in the restart input, LS-DYNA probably incorrectly read this load curve data.

9. Fixed bug in tprint output (to ASCII and binout). The bug led to an inconsistency in the output with respect to the number of thermal boundary conditions. Multiple boundary conditions of the same type assigned to the same segment set resulted in a segmentation fault of the l2a tool. Now for multiple boundary conditions defined on one segment set, the data for the heat transfer from the boundary condition is given individually for each condition.

10. Fixed a bug when filtering the dangling nodes out of the temperature boundary conditions. It sometimes resulted in very strange temperature distributions in MPP.

11. Fixed an issue for redundant constraints in the thermal solver. Now the code ensures that *BOUND-ARY_TEMPERATURE_RSW does not constrain 'top' and 'bottom' temperatures of nodes that are not attached to a thermal thick shell.

## 7.3.31 User-Defined Features

1. Fixed issue where *MODULE_PATH was not read if `LD_LIBRARY_PATH` was empty.

2. Added missing subroutines `uctrl1/2` and `utimestep` to TYPE = UTIL in *MODULE_USE. Before this fix, neither of those user subroutines could be properly used with shared objects.

3. Fixed issue where *MODULE_USE could not be used for thermal materials.

4. Fixed issue with the combination of *PART_AVERAGED and *MODULE. A missing part ID could lead to a crash or error (SOL+1378: `No user subroutine (usrmat) is loaded for part 0`).

5. Fixed issue with *USER_INTERFACE_FRICTION. The determination of element history variables could go wrong if triangular shell elements were involved. Also, a large number of history variables were not taken care of correctly.

## 7.3.32 Miscellaneous

1. Fixed MPP support for part numbers greater than five digits in *DEFORMABLE_TO_RIGID_AUTO-MATIC.

2. Fixed reinitialization issue with the flag that controls the output of the "relax" file during dynamic relaxation. It was not properly set up for *CASE support, so only the first *CASE having dynamic relaxation was writing this file.

3. For *DEFINE_CURVE_FUNCTION, fixed a bug that occurred when the function to be defined involved "TIME." It previously output the simulation time at the next time step, not the current step.

4. Fixed bug in *SET_NODE_ADD that could lead to an abnormal termination due to a memory issue.

5. Fixed a bug for OPTION = SET_TSHELL of *SET_SEGMENT_GENERAL. The bug caused incorrect segment definitions.

6. Downgraded Error to Warning when using reserved parameter names in *PARAMETER.

7. Deactivated the Stress Range computation enabled with *DEFINE_MATERIAL_HISTORIES for the dynamic relaxation phase.

8. When generated part sets (*SET_PART_LIST_GENERATE/_INCREMENT) referenced parts including *ELEMENT_TSHELL_COMPOSITE or *ELEMENT_SHELL_COMPOSITE, the set became corrupted.

9. Fixed inconsistency among hybrid runs that occurs when using a different number of threads for models in which elements are eroded.

10. Compute external work for *INTERFACE_LINKING_NODE(_SET), which was not computed at all.

11. Support *DEFORMABLE_TO_RIGID with consistent mass scaling (RBSMS = 1 on *CONTROL_-RIGID). This combination previously resulted in erroneous behavior.

12. Fix for *PART_STACKED_ELEMENTS when used with *SET_PART. Part sets are modified if they contain PIDREF to add the newly generated parts instead. But the removal of PIDREF itself was missing.

13. Fix for option _NAMES of *DEFINE_MATERIAL_HISTORIES. It did not work if a single precisions version read from a double precision structured input: output to d3hsp and hisnames.xml was broken.

14. Fix for combination of *DEFINE_MULTI_SHEET_CONNECTOR with *DAMPING_PART_STIFF-NESS and RYLEN = 2 (*CONTROL_ENERGY). This combination did not work.

15. Fixed issue with *PART_AVERAGED. A segmentation fault during initialization was possible if the nodes were not numbered "correctly," but the numbering should not matter and now does not.