THIS IS A DRAFT:
You can use it with the following conditions:

- Please send me your **name**, **company name** & **address**, **title**, **email**, (phone). So I can send you the updated version as it becomes available.
- **Please do not alter or circulate any part without LSTC consent!**

I appreciate any suggestions, corrections or feedback.
Any questions, please contact Ian Do at ian@lstc.com.

Livermore Software Technology Corporation

For questions or comments please call LSTC at (925)-449-2500

LS-DYNA – ALE Capabilities (Arbitrary-Lagrangian-Eulerian) Fluid-Structure Interaction Modeling

January 07, 2003

Lars Olovsson

Mhamed Souli

Ian Do

Livermore Software Technology Corporation

OUTLINE

I. Introduction

II. Some simple illustrative results

III. Lagrangian, Eulerian and ALE descriptions

IV. Material Deformation Behavior

V. Explaining the *ALE_ commands

VI. Examples

VII. Some Specialized Applications

[I] INTRODUCTION

[I] Introduction

This tutorial is designed to guide new LS-DYNA users through examples of how Fluid-Structure Interaction (FSI) problems can be set-up using LS-DYNA. Specifically, it aims to explain the basic commands required to describe certain physical phenomena. The goal is to enable the new users to understand "WHAT COMMANDS ARE NEEDED" to get started in modeling various FSI scenarios. Various examples will illustrate simple how-to steps to model FSI.

We will start by looking very quickly at some results, then reviewing the differences among the various computational methods that can be employed. Next, we will look at the *ALE_ commands, and lastly examples to explain their usages. Step-by-step construction of simple examples will be presented.

[I] Introduction (cont.)

The examples were created by Lars Olovsson and Mhamed Souli, and Ian Do. Pat Kulzer helps with the presentation lay-out. Mike Burger helped creating many of the input models. Morten Jensen, Jim Day, Khanh Bui, Todd Slavik, XinHai, and Lee Bindenman helped with many suggestions. Philip Ho and Roger Chen provided support with the post-processing. Dilip Bhalsod created the airbag model to be supplied as an example of airbag deployment modeling using Lagrangian-ALE coupling method. Jason Wang provides information on MPP capability.

ALL DATA ARE FICTITIOUS SHOWN FOR THE PURPOSE OF ILLUSTRATING THE METHOD. THE USERS MUST BE RESPONSIBLE FOR THEIR OWN DATA VALUES!

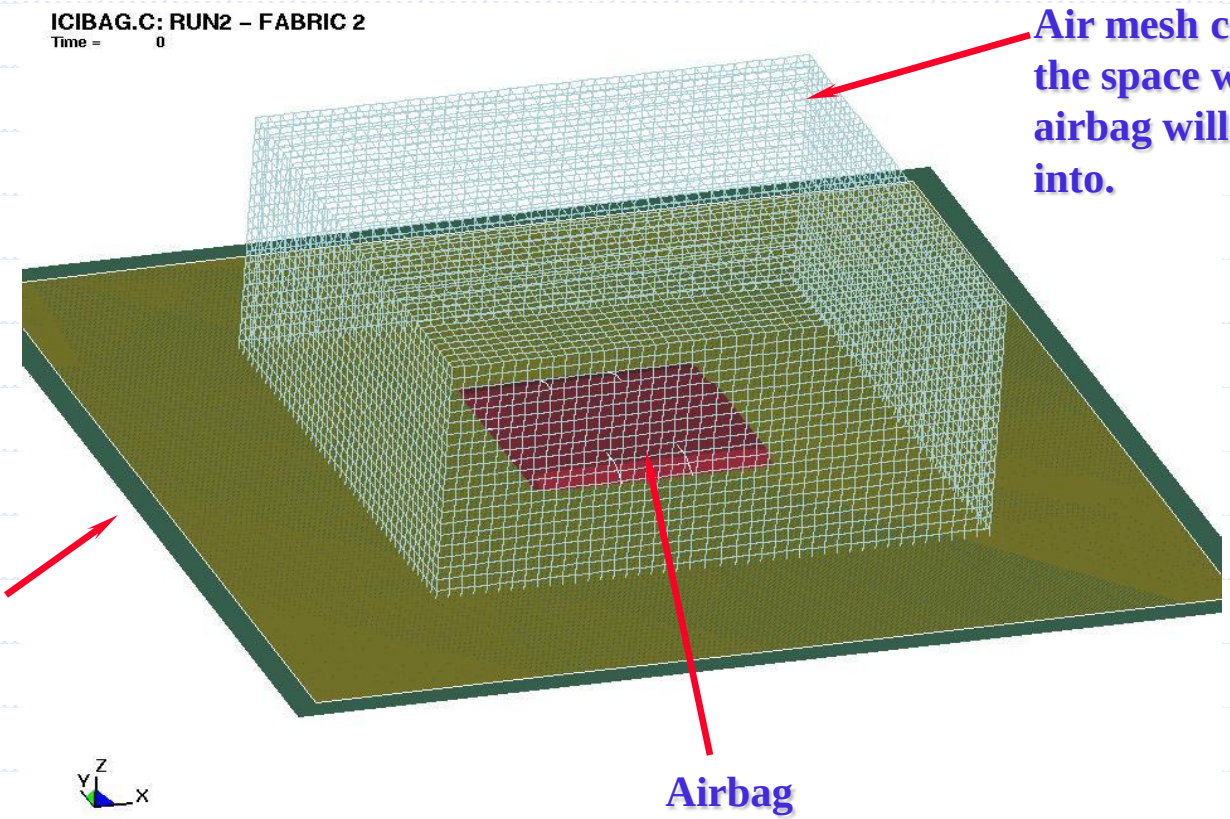
[II] SOME ILLUSTRATIVE RESULTS

[II] Some Illustrative Results

- Airbag
- Dropping water tank/barrel (ALE mesh moves with a user-defined coordinate system - 3 nodes)
- Forging
- Purging of fluid from a chamber
- Plate impacting fluid
- Explosion-induced fluid-structure-interaction

A Simple Airbag Model Using ALE Coupling

Fluid = Air mesh & Structure = Airbag



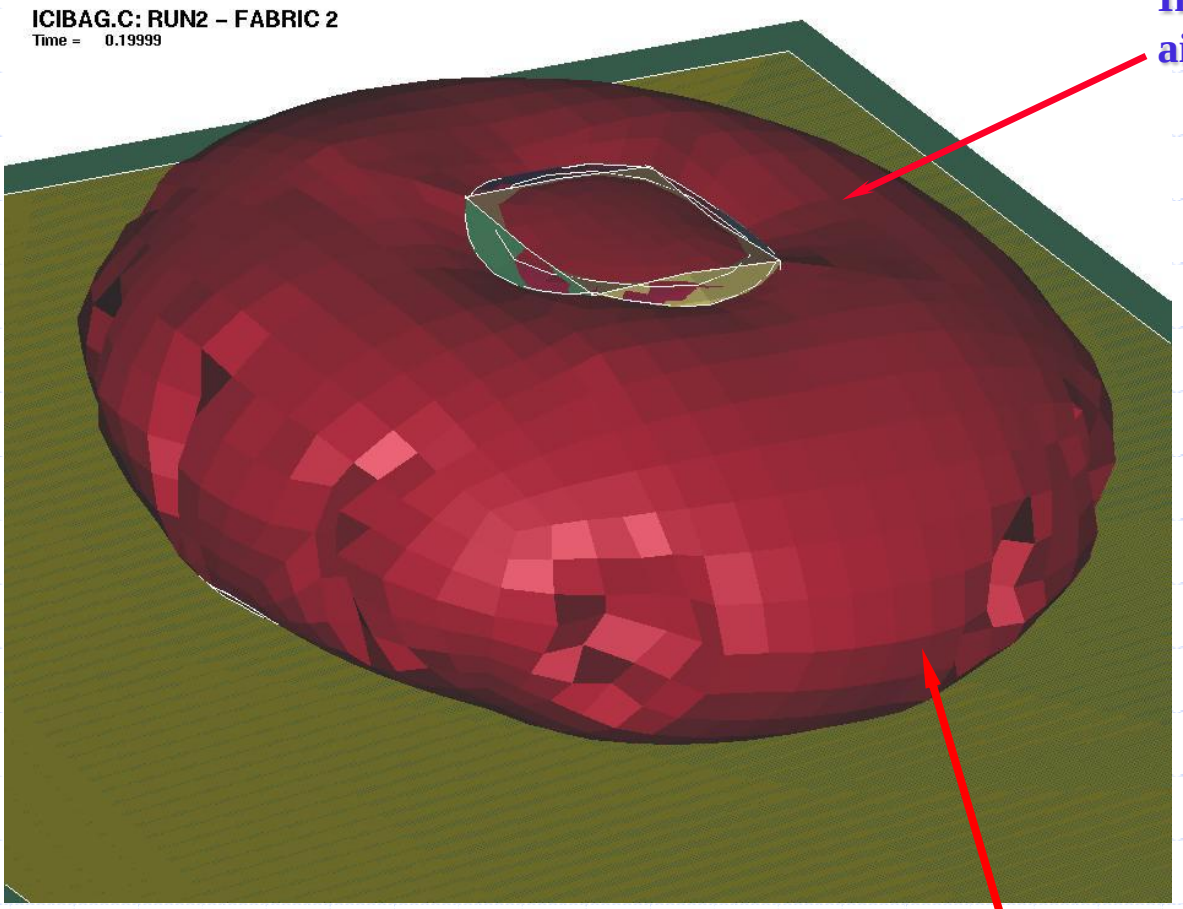
Air mesh covering the space which the airbag will expand into.

Backing platforms

Airbag

A Simple Airbag Model Using ALE Coupling

DEPLOYED CONFIGURATION

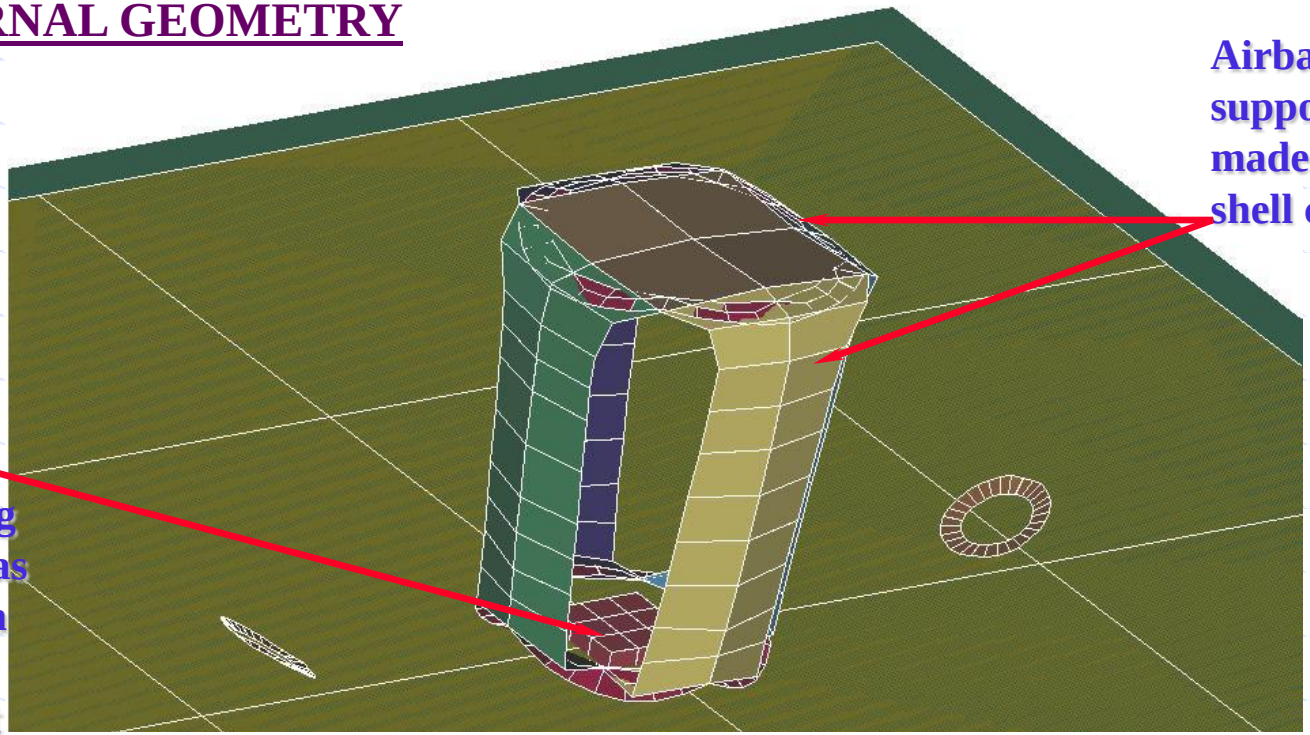


Inflated airbag

Blanking out the bag for internal view … ➔

A Simple Airbag Model Using ALE Coupling

INTERNAL GEOMETRY



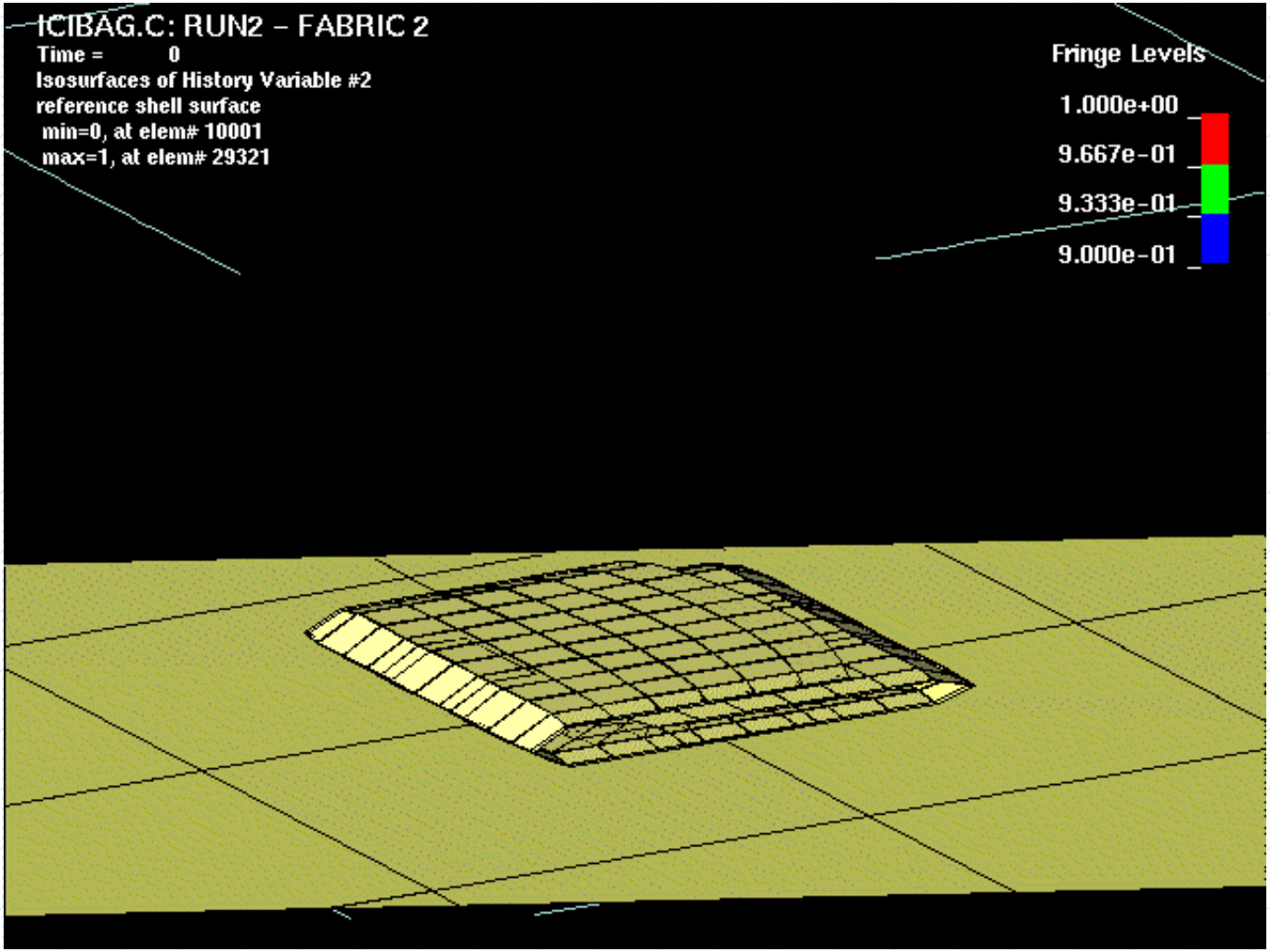
Airbag internal support structures made up of many shell components.

Priming Ambient Fluid Block (supplying pressure as a function of time) located at airbag opening

Contact among the Lagrangian structures:
*CONTACT_TIED_NODES_TO_SURFACE
*CONTACT_AIRBAG_SINGLE_SURFACE, etc.
Fluid-Structure-Coupling:
*CONSTRAINED_LAGRANGE_IN_SOLID

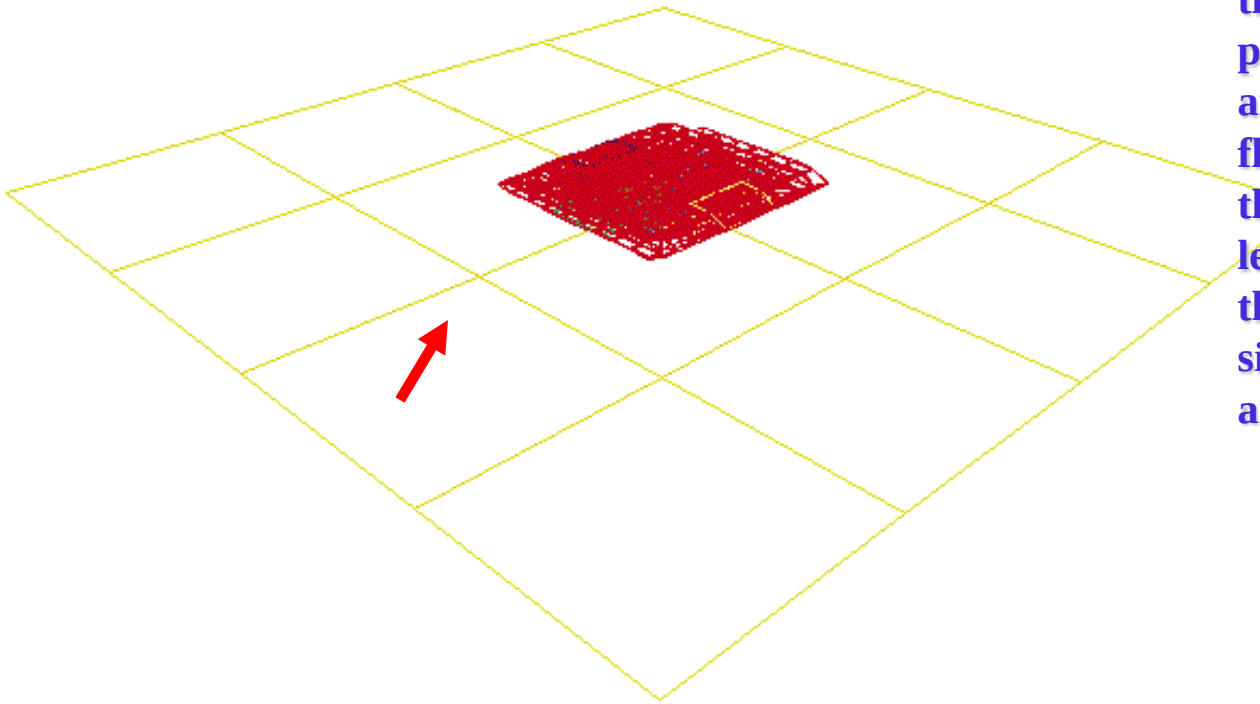
Airbag Deployment AVI

(double-click on picture below)



Airbag Deployment AVI

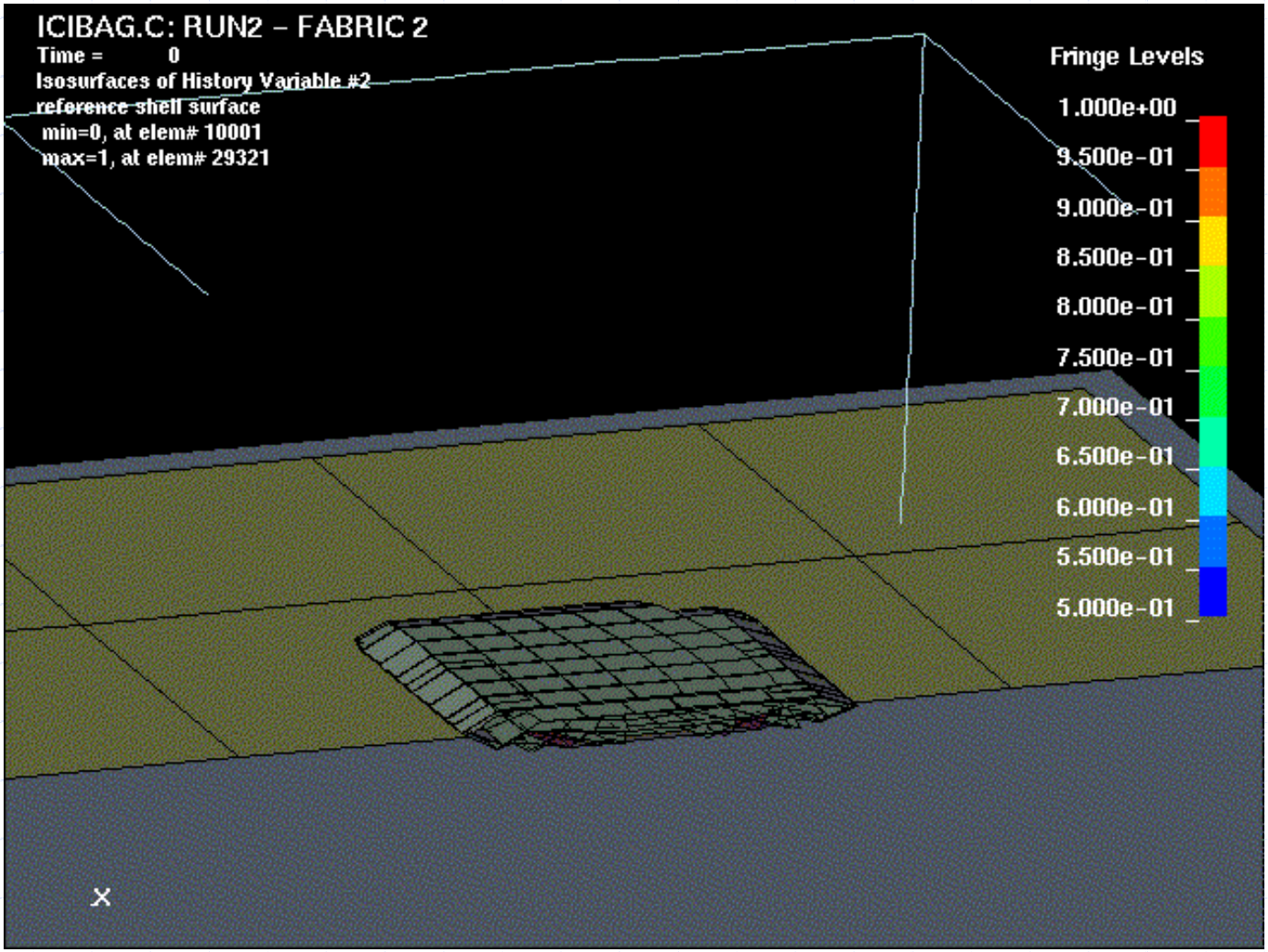
(double-click on picture below)



The red mesh is the airbag. The blue color under the mesh is the volume-fraction contour of the fluid being pumped into the airbag. Some of this fluid is relieved out the vent hole (lower-left quadrant, near the end of the simulation, near red arrow)

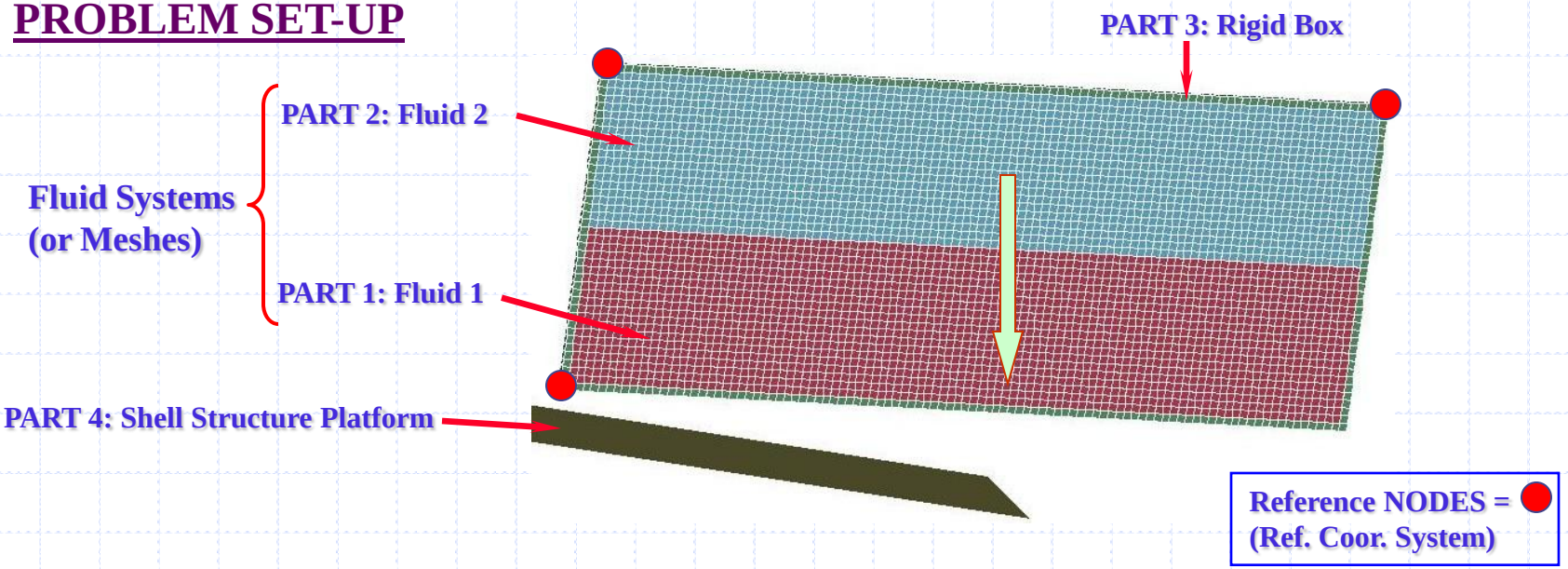
Airbag Deployment Cross-Section AVI

(double-click on picture below)



Dropping Of A Fluid Box

PROBLEM SET-UP

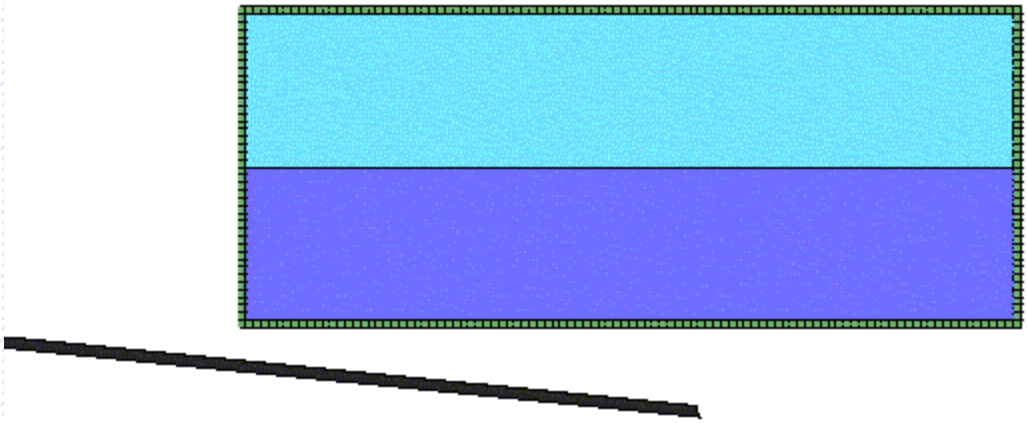


- A rigid box containing 2 different fluids is dropped and hit a platform.
- The fluid system is assumed to follow the motion of the rigid tank.
- The motion of the rigid tank is defined by the 3 nodes attached to the body of the tank.
- The consequent motion results in the sloshing of the fluid inside the tank.

Dropping Of A Tank Containing Fluid(s) - AVI

(double-click on picture below)

Example of ALE mesh motion following a coordinate system defined by three user-defined nodes

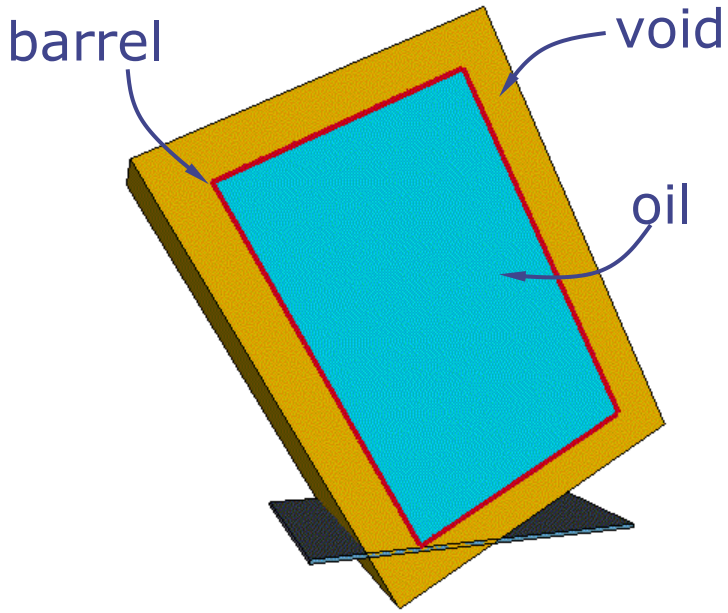


Drop Test Of An Oil Barrel – 3D Model AVI

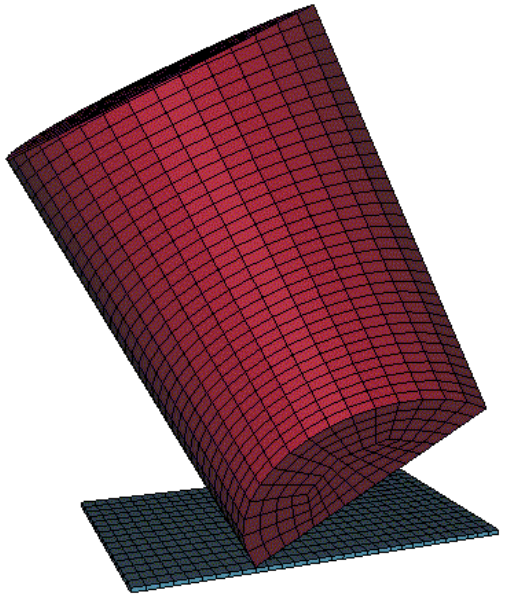
(double-click on pictures below)

A similar example in 3D … (mesh moving following a reference body)

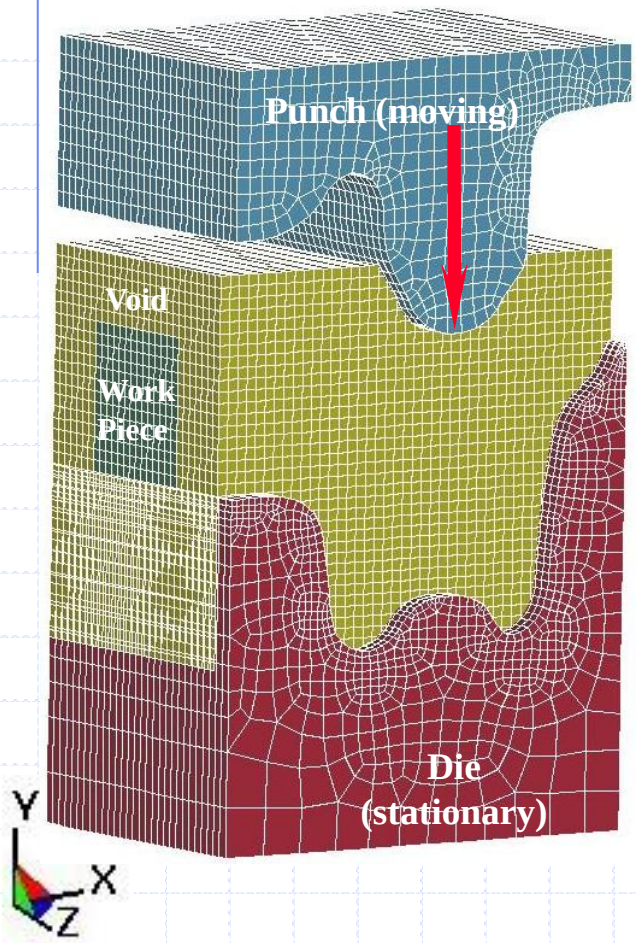
Barrel and fluids



Barrel Only



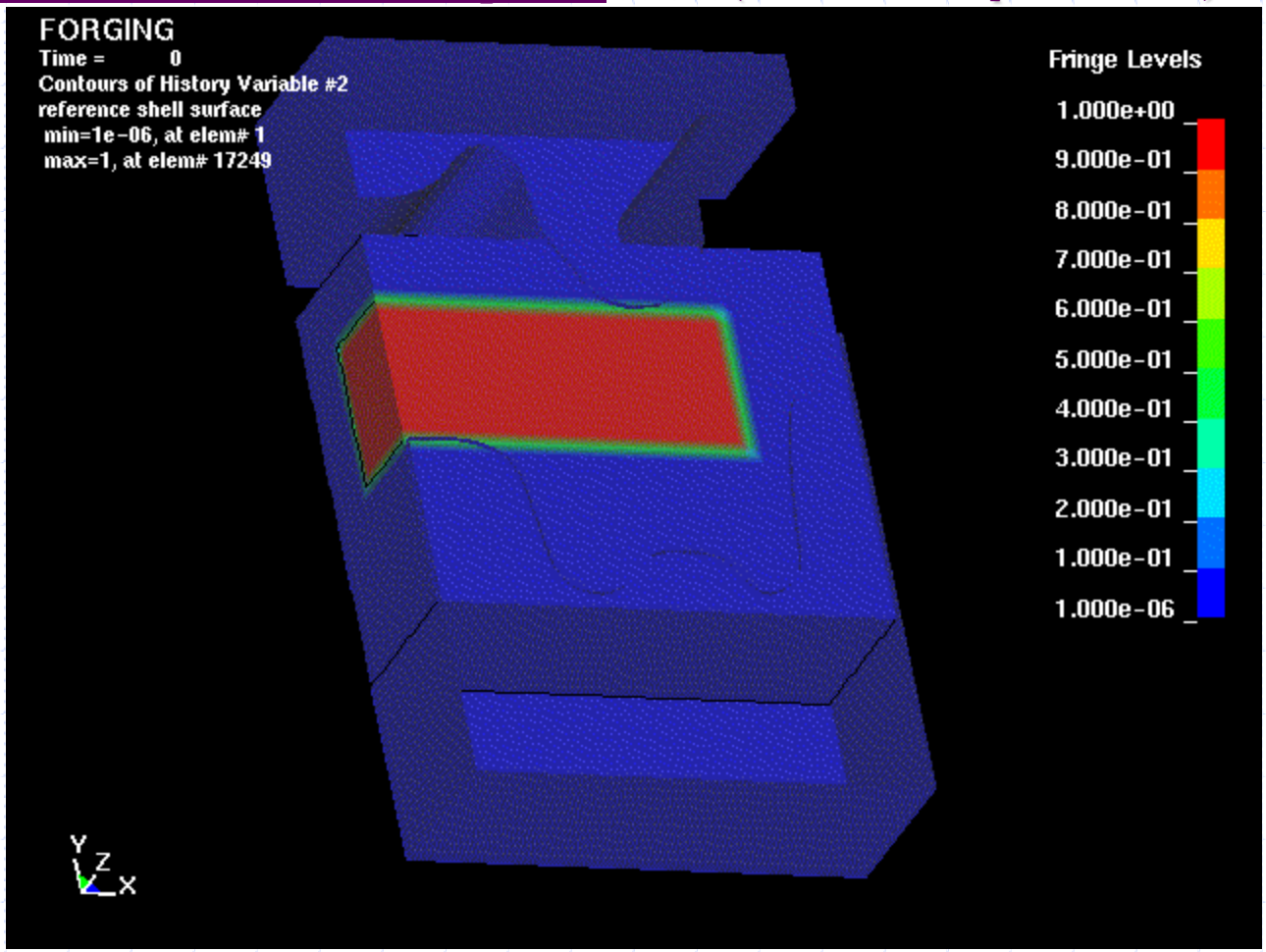
3D Forging: Rigid Tools & Deformable Work-Piece



- Both tool pieces, punch and die, are modeled as Lagrangian *rigid shell* structures .
- The work piece is modeled as *solid* ALE material which is allowed to deform|flow into surrounding void space.
- The void mesh can overlap with the rigid tool structures.

3D Forging: Rigid Tools & Deformable Work-Piece

Result viewed at a cross-section plane: AVI (double-click on picture below)



Purging Flow Through An Orifice (multi-fluids)

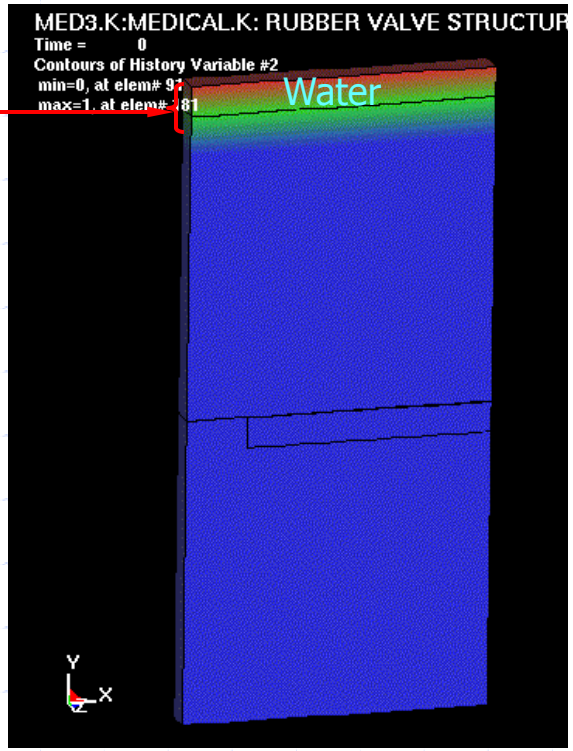
AVI's (double-click on pictures below)

Priming Fluid Reservoir

(Initially Pressurized Fluid)

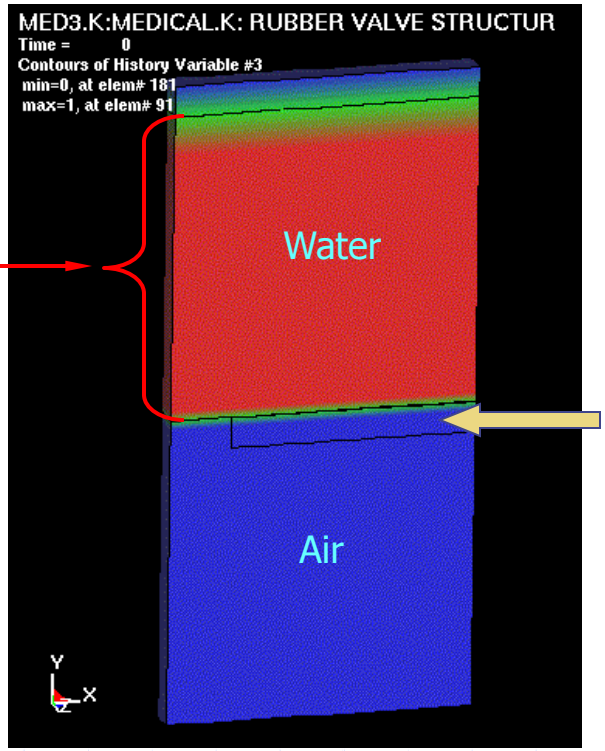
Fluid Being Purged

(Fluid initially at ambient condition).



Priming fluid flow

Zero gravity



Purging fluid flow

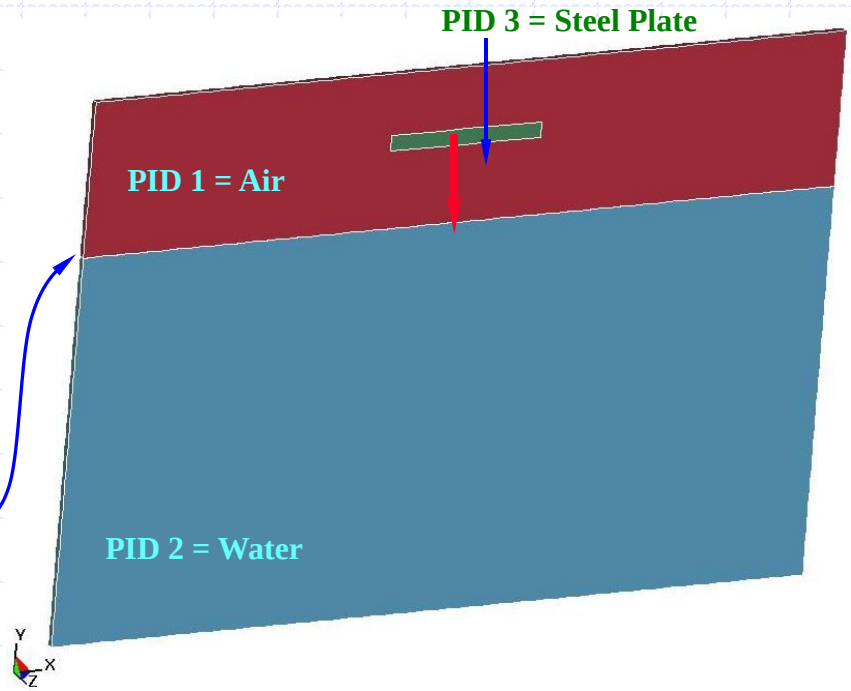
Note the deflection of the simplified "valve"

Lagrangian Plate hitting ALE Multi-Material Fluids

OVERVIEW:

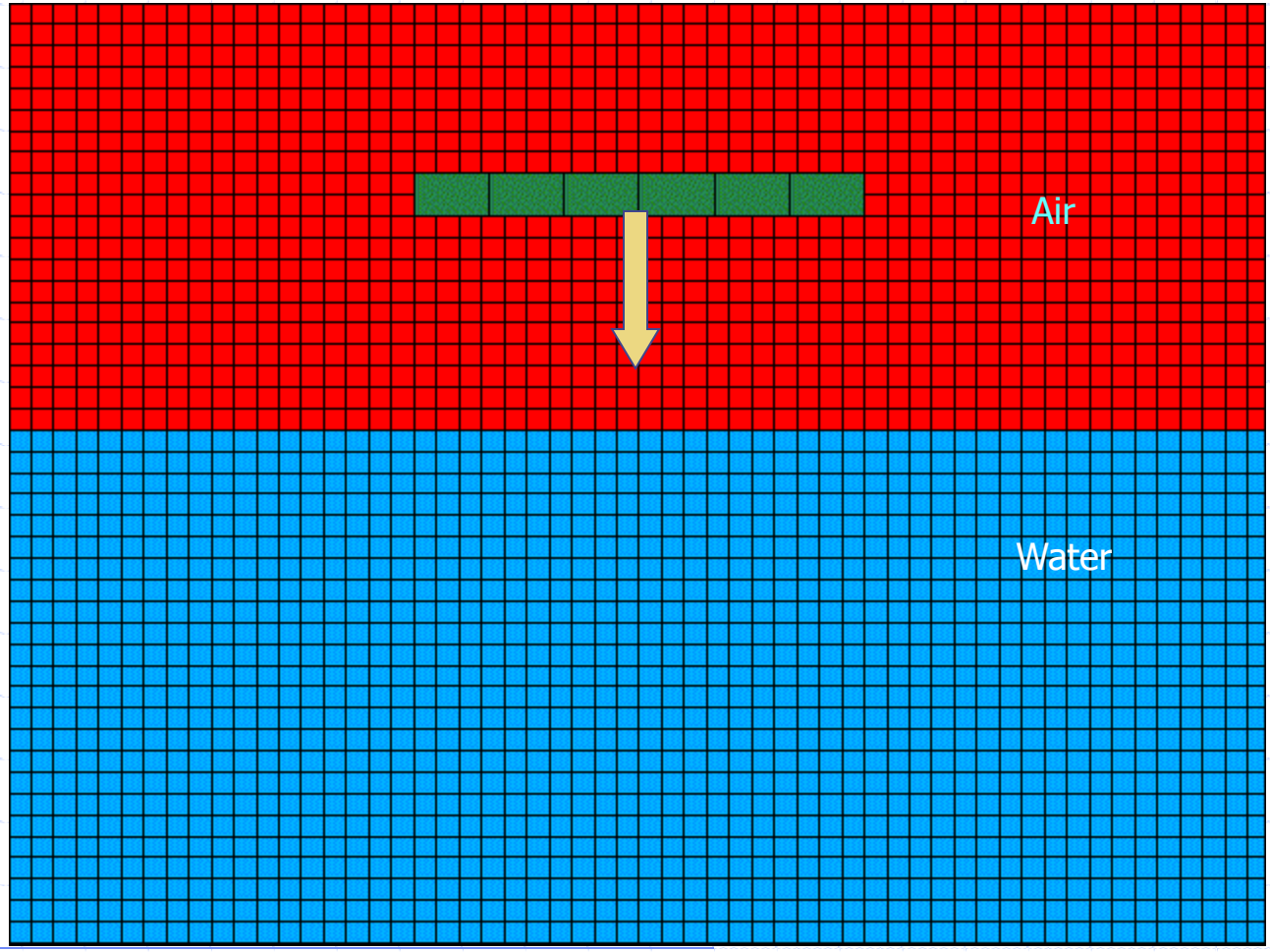
A Lagrangian plate moves with "-y" velocity through air, then hits water.

- The Air and Water are defined as ALE Multi-Materials (tracking the interface of the two material within each element).
- The Steel Plate is defined as Lagrangian.
- The Lagrangian body/mesh can overlap the ALE/fluid meshes.
- The ALE-Multi-Material meshes have merged nodes on their shared boundaries.



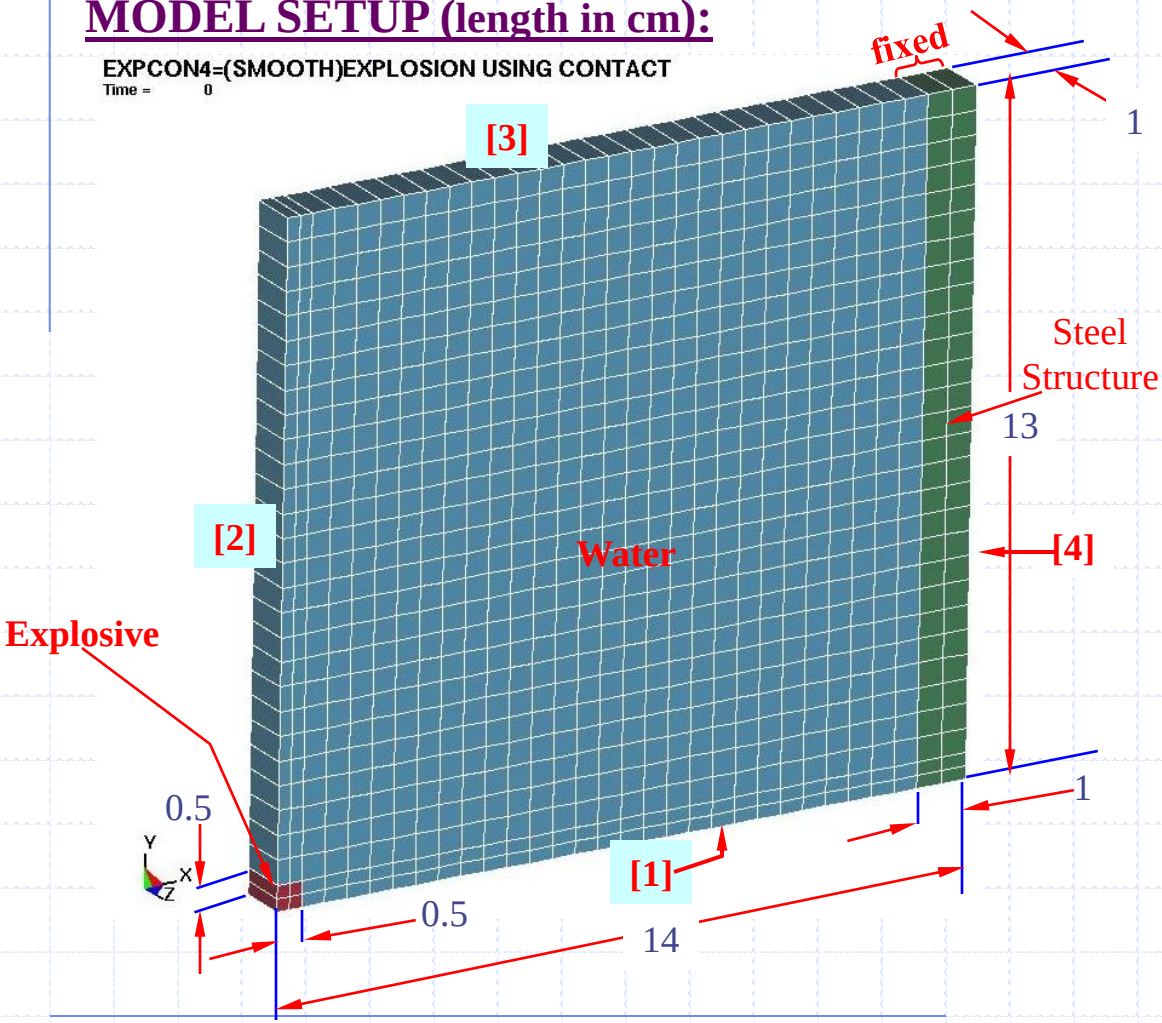
Lagrangian Plate hitting ALE Multi-Material Fluids

AVI (double-click on picture below)



Explosion-In-Fluid Effects on Steel Structure

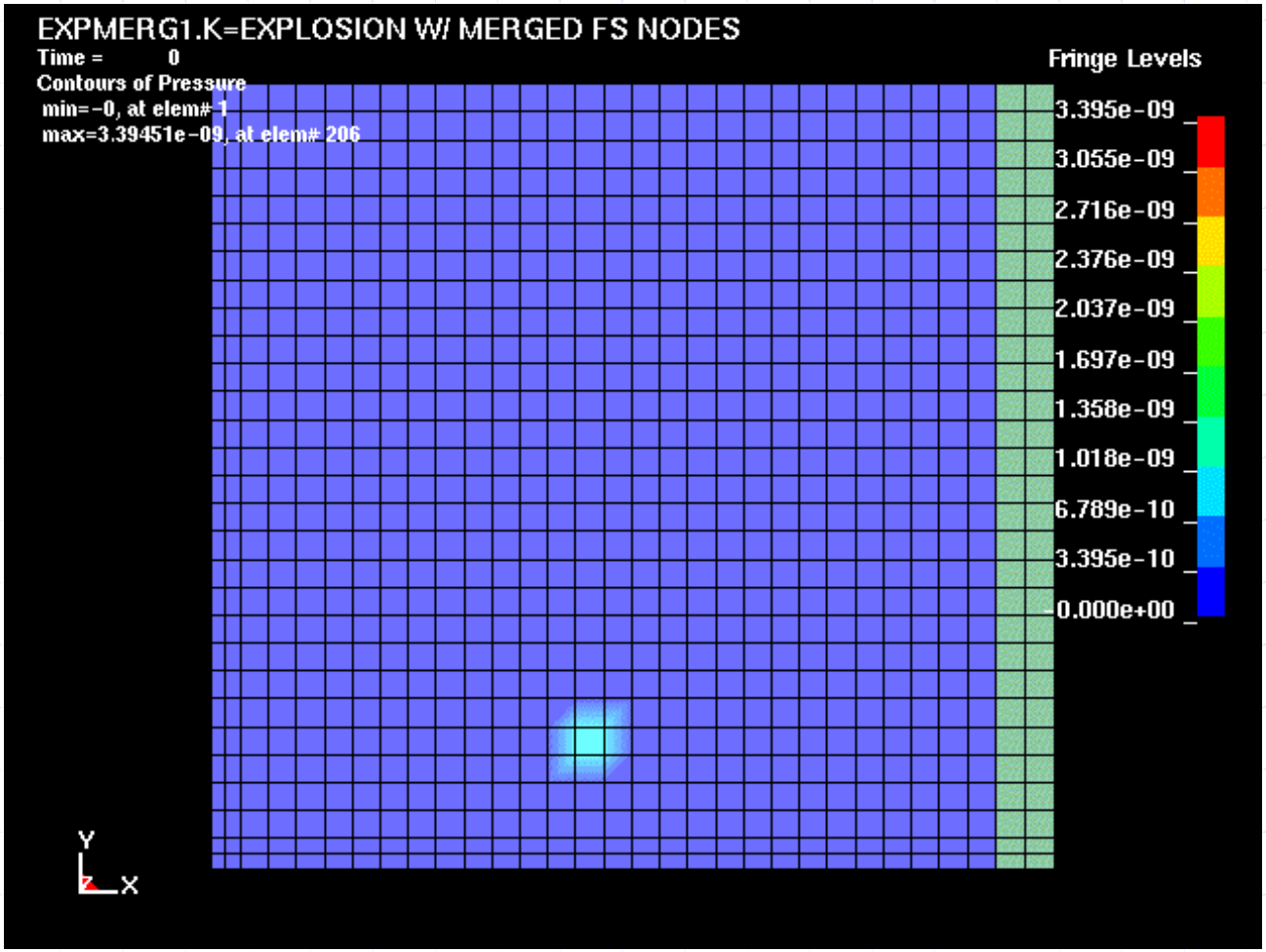
MODEL SETUP (length in cm):



- Surfaces [1], [2] and [3] are constrained with rigid boundaries.
- Surface [4] is covered by the steel structure which will deform under the pressure load caused by the explosive.
- Smoothing is done on the ALE mesh.

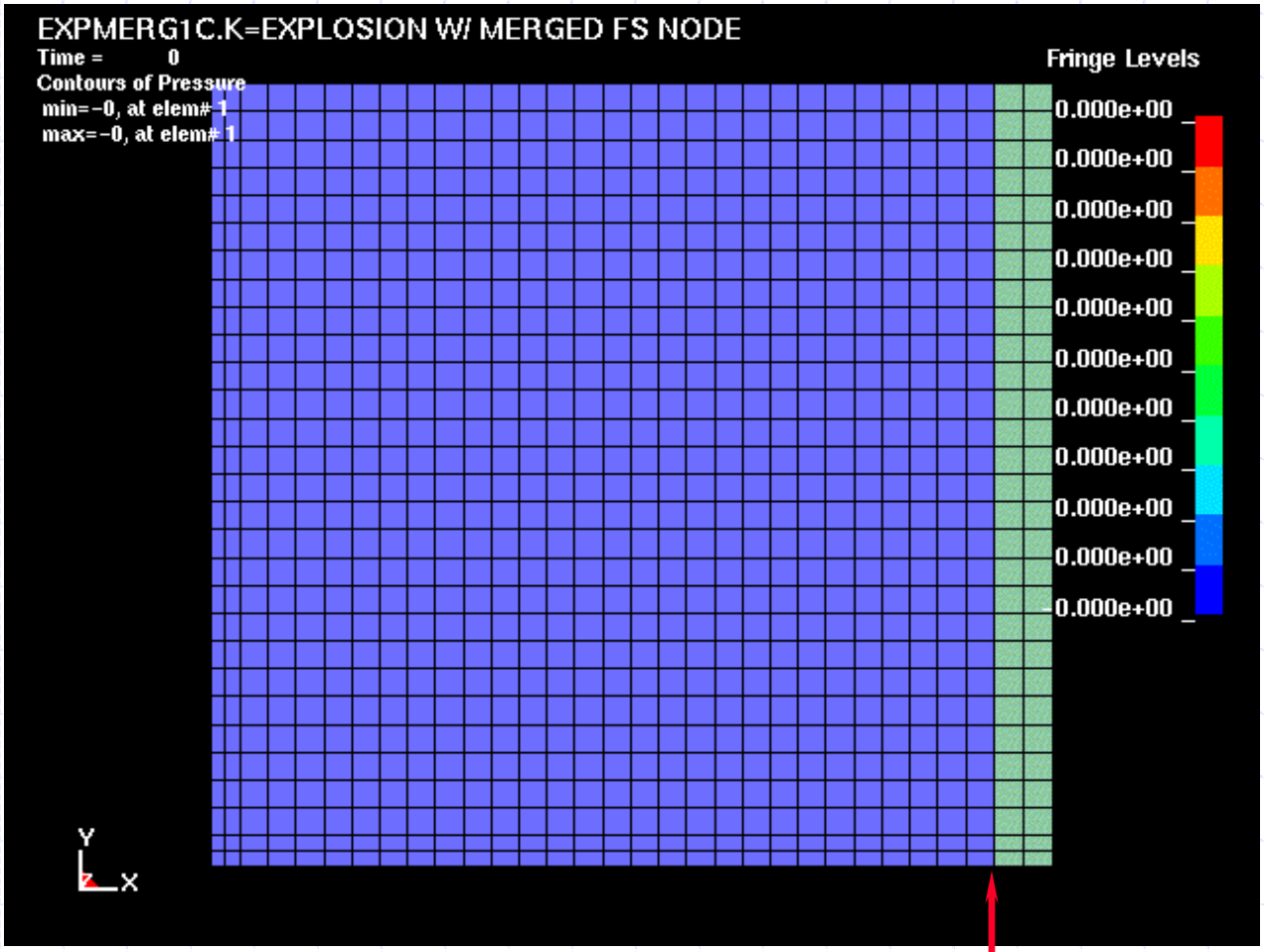
Explosion-In-Fluid Effects on Steel Structure

WITH *ALE_SMOOTHING (mesh smoothing); AVI (double-click on picture below)



Explosion-In-Fluid Effects on Steel Structure

WITHOUT *ALE_SMOOTHING; AVI (double-click on picture below)



Merged Nodes on boundary

[III] LAGRANGIAN, EULERIAN AND ALE APPROACHES

[III] Lagrangian, Eulerian & ALE

Methods of Computation

1) Pure Lagrangian.

2) ALE formulation with 1 material in each element.

3) Eulerian formulation.

4) ALE formulation with multi-materials in an element.
(Translating/ Rotating/ Expanding mesh systems.)

ELEMENT FORMULATIONS REVIEW

There are many ways to model the same physical process. The different methods may depend on different element formulations. Some clarification of the computational approach taken by each element formulation is presented. For fluid modeling, we will focus on the 3D fluid element. Hence, the *SECTION_SOLID command, specifically, the parameter "ELFORM" is discussed.

ELFORM:

1 = Constant stress solid (pure Lagrangian formulation).

5 = 1-point ALE (single material in each cell).

6 = 1-point Eulerian (single material).

7 = 1-point Eulerian Ambient element.

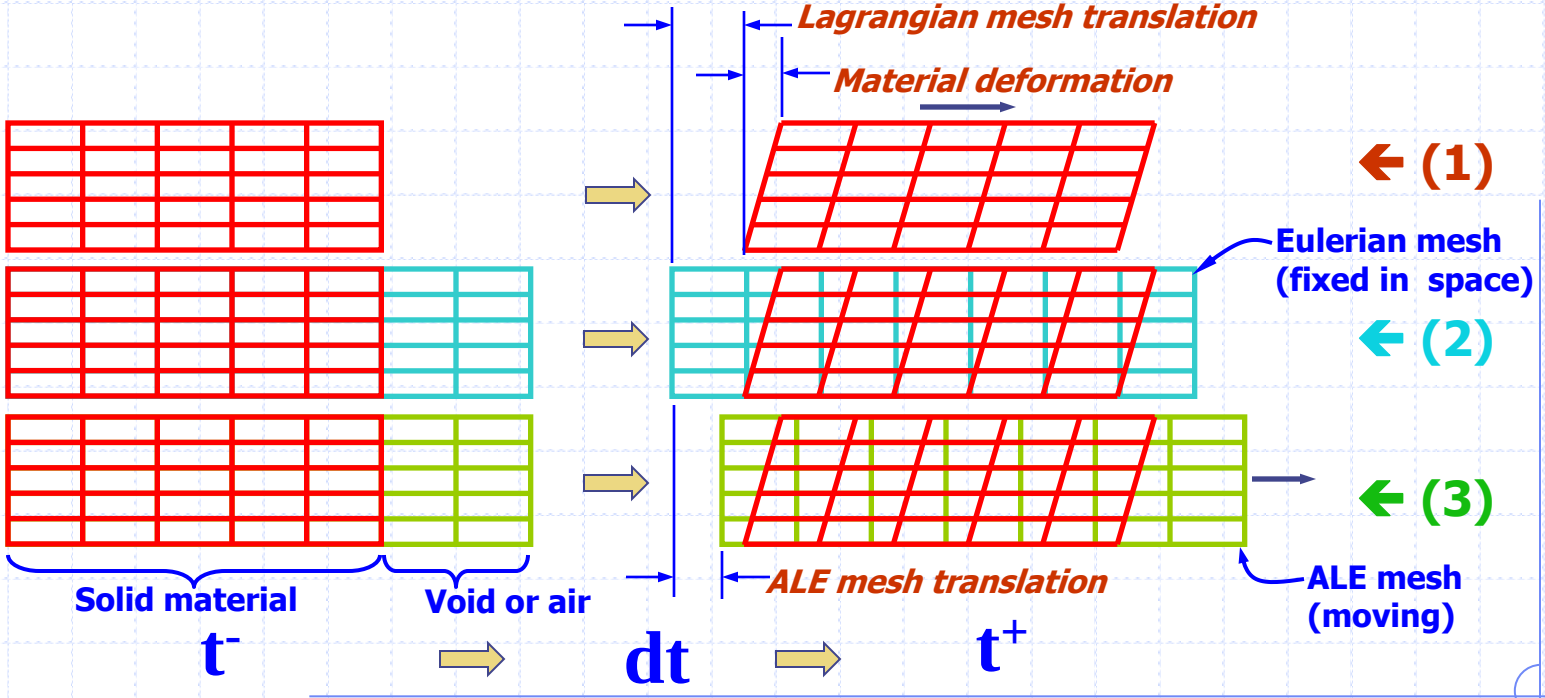
11 = 1-point ALE multi-material element ← *most important*

12 = 1-point ALE single-material-and-void.

ELEMENT FORMULATIONS REVIEW

Element Formulations and Applications:

Let us consider a 2D example, a solid piece of metal is **moved** and then **deformed** as shown below. Three formulations may be used: (1) Lagrangian, (2) Eulerian, (3) ALE (Arbitraty-Lagrangian-Eulerian).



ELEMENT FORMULATIONS REVIEW

In the following, consider 1 time step (referring to the previous figures):

(1) Lagrangian:

The nodes of the mesh are attached to the imaginary material "points". These nodes move and deform with the material. This is shown in (1) above.

(2) Eulerian:

Consider 2 overlapping meshes, one is a background mesh which is fixed in space, and the other is attached to the material which "flows" through the former fixed mesh. This may be visualized in 2 steps:

First, the material is deformed in a Lagrangian step just like the Lagrangian formulation.

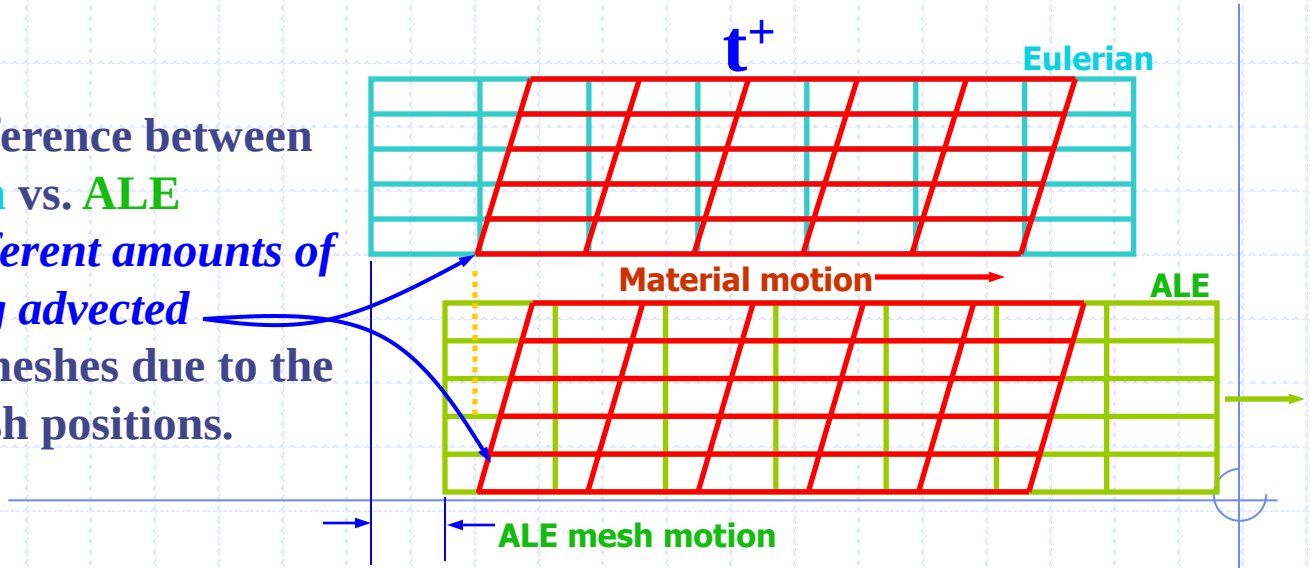
Then, the element state variables in the "Lagrangian elements" (red) are mapped or advected or rezoned or distributed back onto the fixed (background) reference Eulerian mesh (aqua).

ELEMENT FORMULATIONS REVIEW

(3) ALE:

Consider 2 overlapping meshes, one is a background mesh which can moves arbitrarily in space, and the other is attached to the material which "flows" through the former moving mesh. This may be visualized in 2 steps. First, the material is deformed in a Lagrangian step just like the Lagrangian formulation. Then, the element state variables in the "Lagrangian elements" (red) are *remapped* or *advected* or *distributed* back onto the moving (background) reference ALE mesh (green).

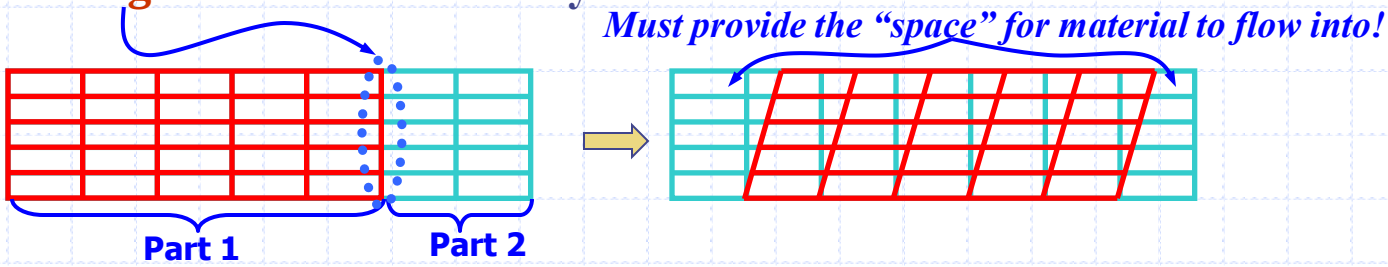
The main difference between pure Eulerian vs. ALE method is ***different amounts of material being advected*** through the meshes due to the reference mesh positions.



ELEMENT FORMULATIONS REVIEW

Some Notes on MULTI-MATERIAL Eulerian or ALE Applications:

- For material to flow from one region or part of the mesh to another, we must have the common nodes on the boundaries of connecting ALE parts be "merged". This means they share the same node ID's.



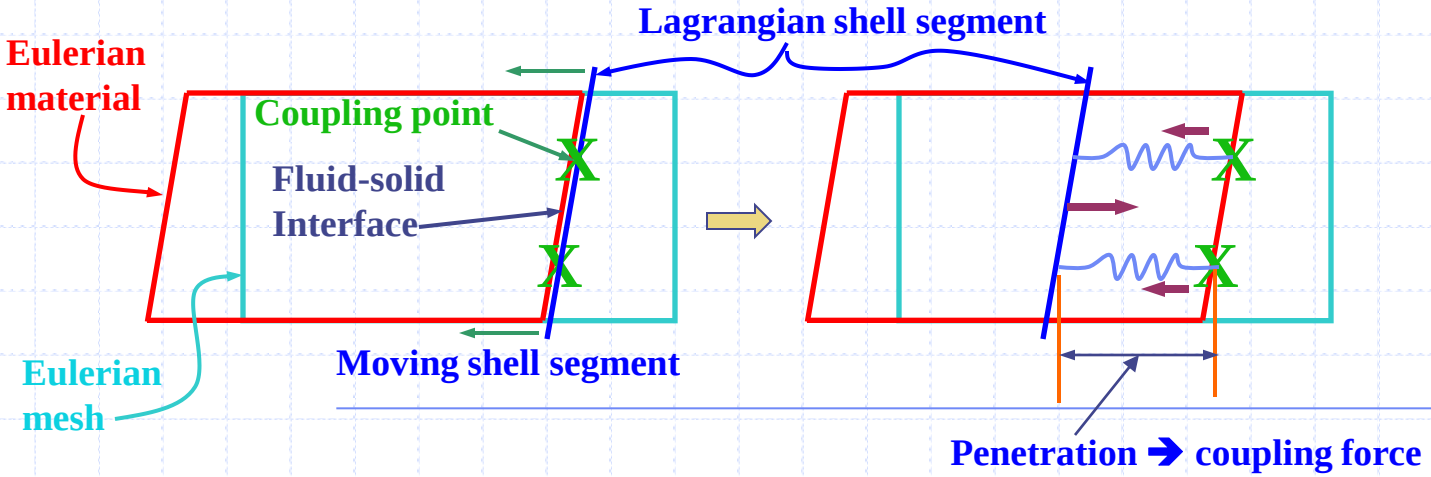
- The Courant criteria restricts the transmission of material (or information) within 1 element per dt ➔ implications:
  - If the mesh moves in the same direction as the material motion resulting in less amount of material advected out of 1 element per dt (thus is more stable and larger dt can be taken), and vice versa.
  - The density change in 1 element per dt is restricted, typically, to less than the current density magnitude. The users should have an intuitive "feel" for the dynamics of the model, i.e. how fast things move.

ELEMENT FORMULATIONS REVIEW

Some Notes on MULTI-MATERIAL Eulerian or ALE COUPLING:

- Lagrangian part hits Lagrangian parts ➔ CONTACT
  Lagrangian part hits Eulerian or ALE parts ➔ COUPLING

  The code searches for the ***INTERSECTIONS*** between the Lagrangian parts & Eulerian (or ALE) parts ➔ If an intersection is detected inside an Eulerian element ➔ It marks the Lagrangian-Eulerian common **coupling points (NQUAD)** on this interface at t^- ➔ It tracks the independent motion of the 2 materials over dt ➔ Then compute the penetration distance ➔ The coupling forces are computed based on this penetration and re-distributed back onto both meshes.



ELEMENT FORMULATIONS REVIEW

Some Notes on MULTI-MATERIAL Eulerian or ALE Applications:

- We will refer to an Eulerian or ALE part as "ALE" in general since Eulerian is just a special ALE case where the mesh velocity is zero.
- In ALE terminology, the user should distinguish between a "*PART ID" and an "ALE-Multi-Material-Group ID" (AMMGID) for the Eulerian or ALE entities:

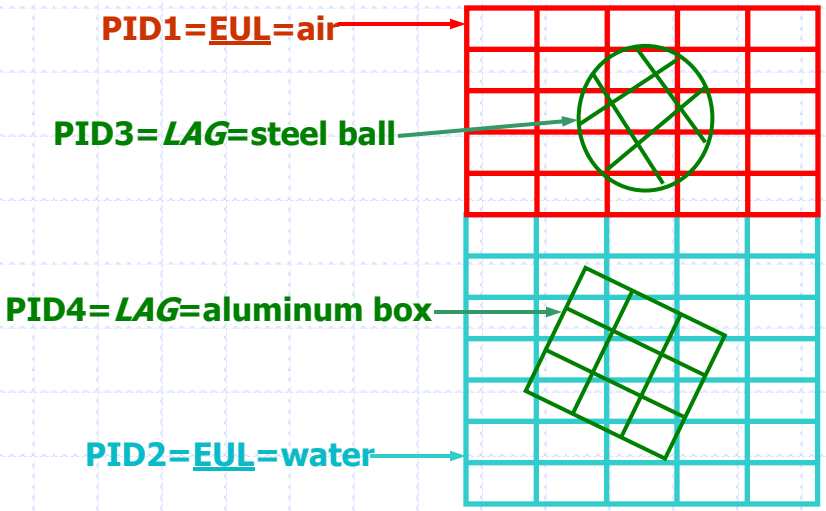
  - A PART usually refers to a mesh entity defined at time zero.

  - An AMMGID refers to a region containing a (fluid) material. For multi-material, the card *ALE_MULTI-MATERIAL_GROUP allows the interface tracking among the many (fluid) materials. This card is discussed in more details later.

  A PART/mesh may contain a single material at time zero, but later may contain other materials as the fluids are moving in and out across the meshes/PART.

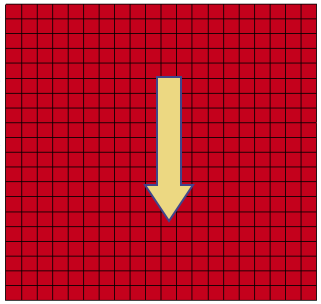
ELEMENT FORMULATIONS REVIEW

- Recall how the COUPLING between Lagrangian mesh and Eulerian mesh occur as shown previously, we see that their meshes must overlap with each other. This is so that their intersections may be detected and their interactions can only be resulting from these intersections. For example, consider the configuration below for concept illustration.

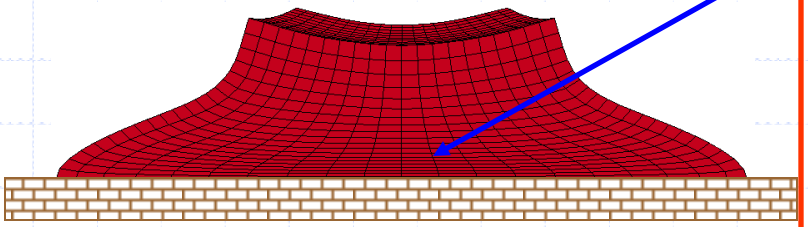


COUPLING: Langrangian meshes overlapping with Eulerian or ALE meshes.

(1) Pure Lagrangian Formulation



(1 material occupies 1 whole mesh)



```
*SECTION_SOLID
$     SECID     ELFORM         AET
          1          1
```

Element formulation 1 = Constant-stress solid

NOTE:

* The mesh deforms with the material.
* Pure Lagrangian method has **no** mesh smoothing.
* Only 1 material in each element.
* The mesh bunches up near the impact surface.

Advantage:

Free surface is followed automatically.

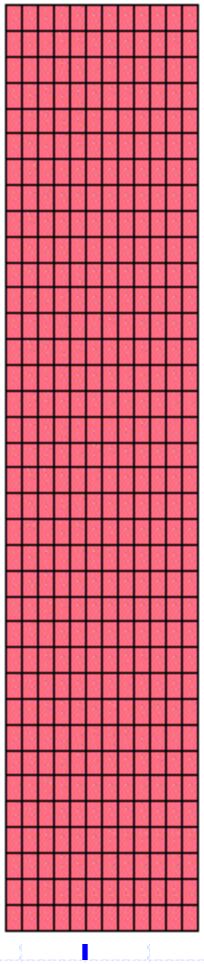
Disadvantage:

Not accurate at large deformation.

Cannot create new surfaces (damaged).

(1) Pure Lagrangian Taylor Bar Impact

(double-click on picture below)



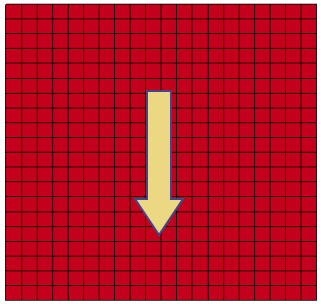
Severely distorted elements near impact surface.

(2) Single Material ALE Formulation with Smoothing

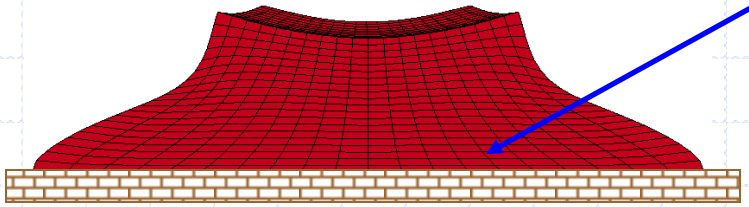
```
*SECTION_SOLID
$     SECID     ELFORM        AET
          1          5
```

Element formulation 5 = 1-point ALE solid

ALE mesh-smoothing activated → *CONTROL_ALE



(1 material occupies 1 whole mesh,just like Lagrangian except it allows for mesh smoothing)



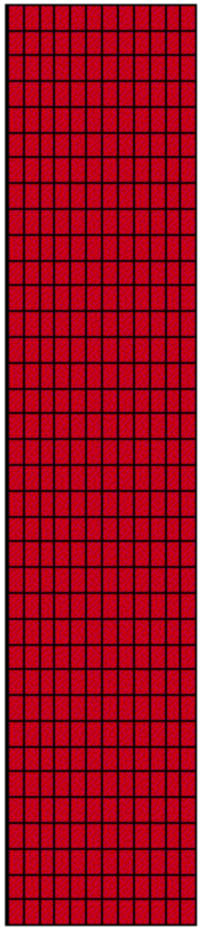
NOTE:
* The mesh deforms with the material.
* ALE method allows mesh smoothing.
* Only 1 material in each element.

Advantage:
Free surface is followed automatically.
Less element error for large deformation.

Disadvantage:
Limited to relatively simple geometry.
Cannot create new surfaces (damaged).

(2) Single Material ALE Formulation with Smoothing

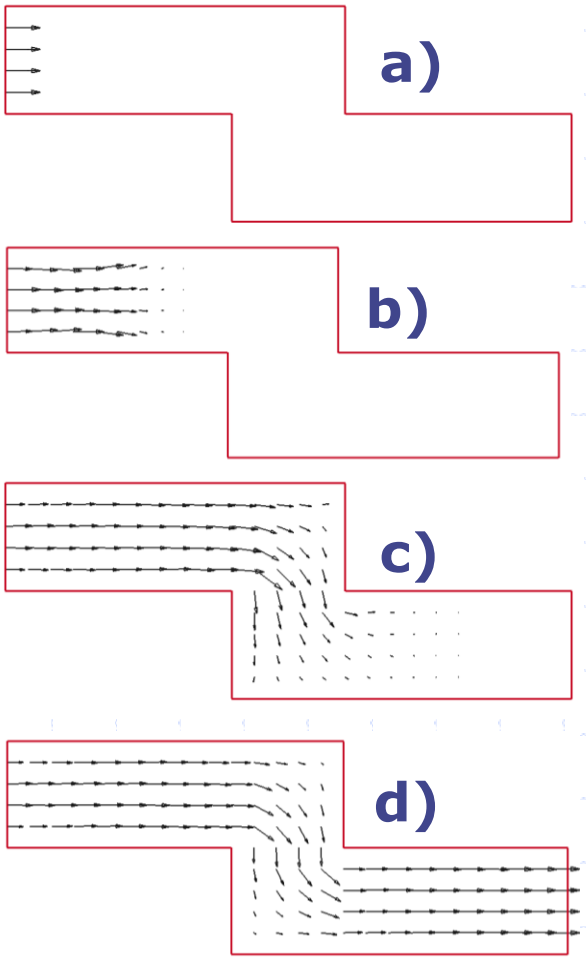
(double-click on picture below)



Smoothed out mesh near impact surface.

(3) Eulerian Formulation with Single Material

(1 material occupies 1 whole mesh)




```
*SECTION_SOLID
$     SECID     ELFORM        AET
          1          6
```

Element formulation 6 (or 7) = 1-point 3D Eulerian element

Advection activated ➔ `*CONTROL_ALE`

NOTE:

* The **mesh is spatially fixed** (no mesh smoothing).
* The material (fluid) flows through the mesh.
* Only 1 material / element.

Advantage:

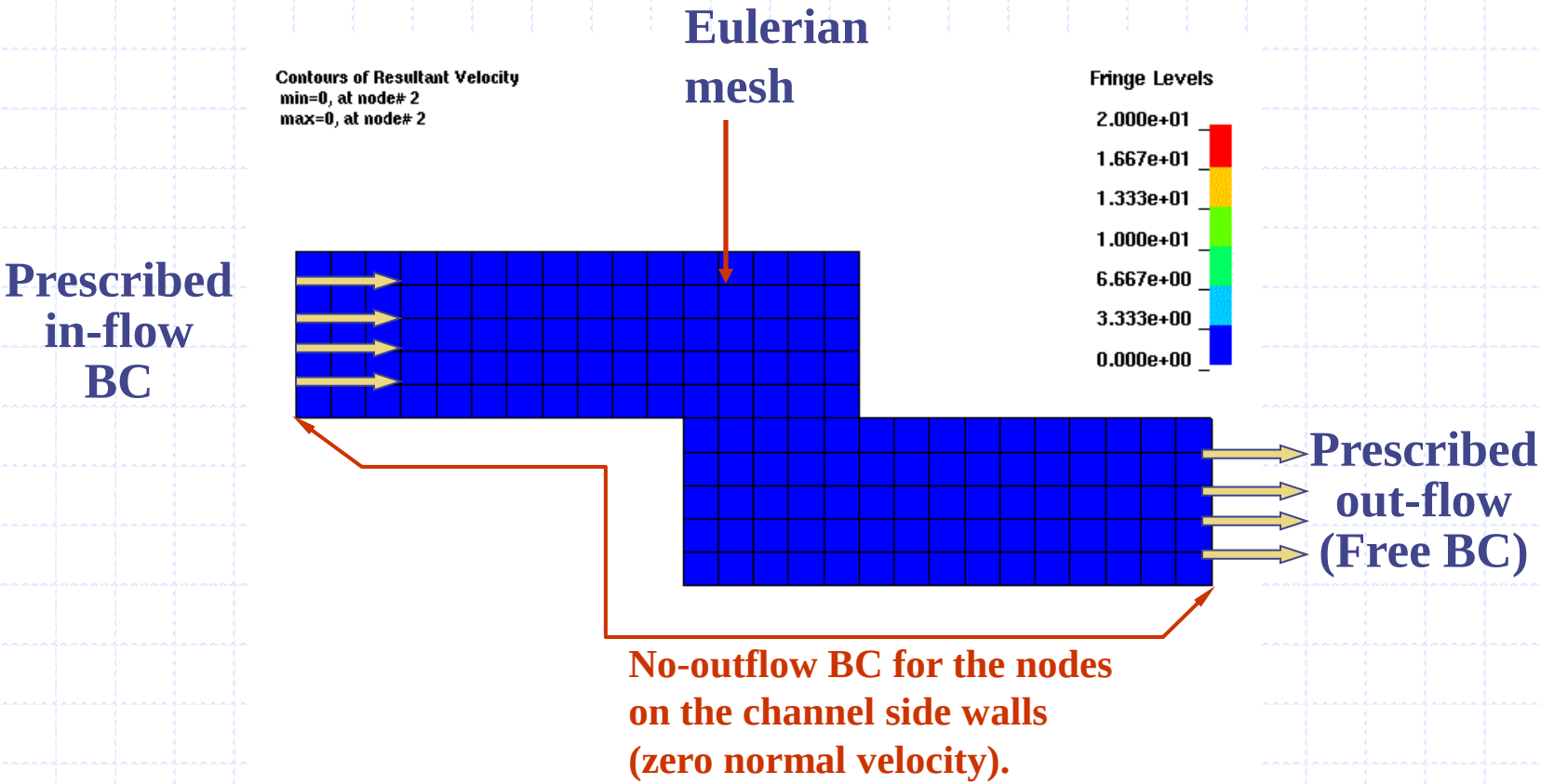
Efficient for very large element deformation (flow).
Can create new free surfaces automatically.

Disadvantage:

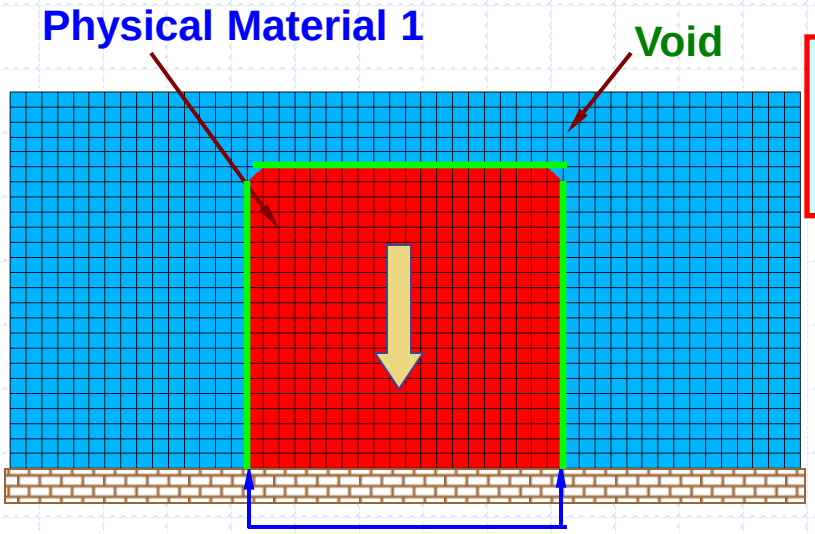
Fixed Eulerian mesh ➔ simple geometry & BC's.
Difficulty in tracking thin material interfaces.

(3) Eulerian Formulation with Single Material

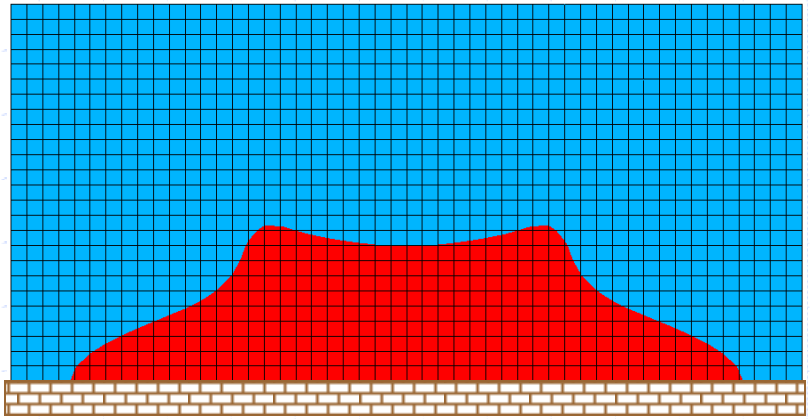
(double-click on picture below)



(4) ALE Formulation with Single Material + Void



Merged nodes on material-mesh and void-mesh boundaries.



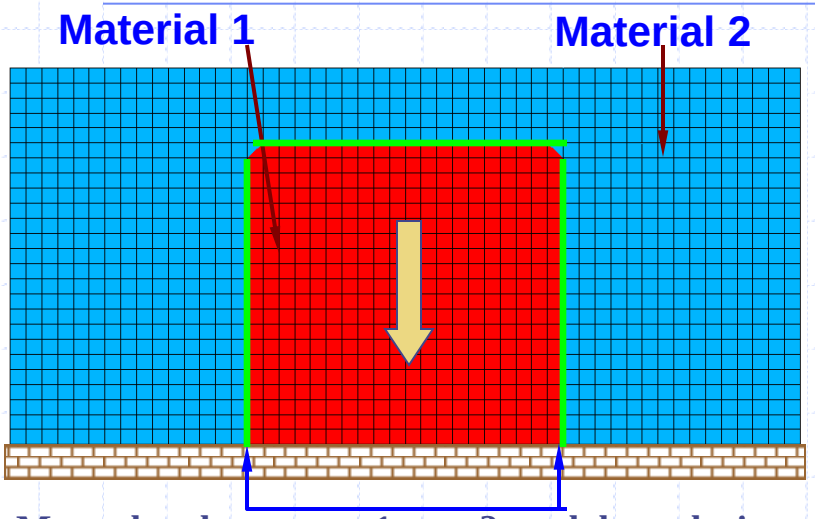
```
*SECTION_SOLID
$      SECID      ELFORM          AET
           1          12
```

Element formulation 12 = 1-point integration 3D-element with 1 material and void

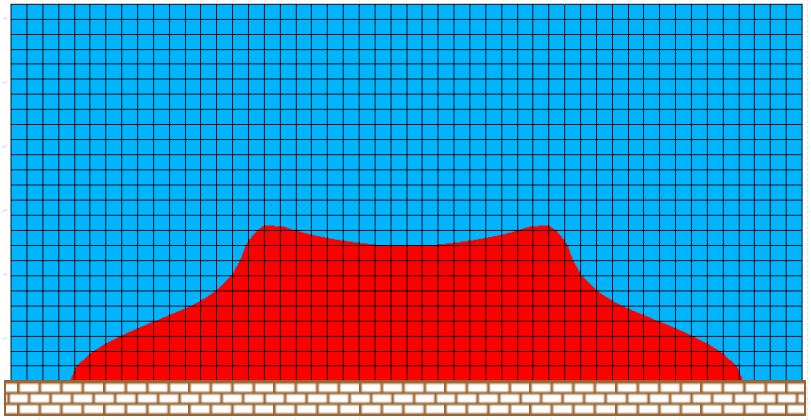
(1 mesh for the "physical" material & 1 mesh for the void ➔ these 2 meshes share merged nodes on their common boundaries)

Advection activated ➔ `*CONTROL_ALE`

(5) Multi-Material ALE Formulation (fixed mesh)



Merged nodes on mat-1-mat-2 mesh boundaries.



```
*SECTION_SOLID
$     SECID     ELFORM         AET
          1         11
```

Element formulation 11 = ALE multi-material

(1 mesh for each "physical" material ➔ all multi-material meshes share merged nodes on their common boundaries to allow for the material flows between them.)

NOTE:
* Material flows through mesh.
* Multi-material/element.

Advection activated ➔ `*CONTROL_ALE`

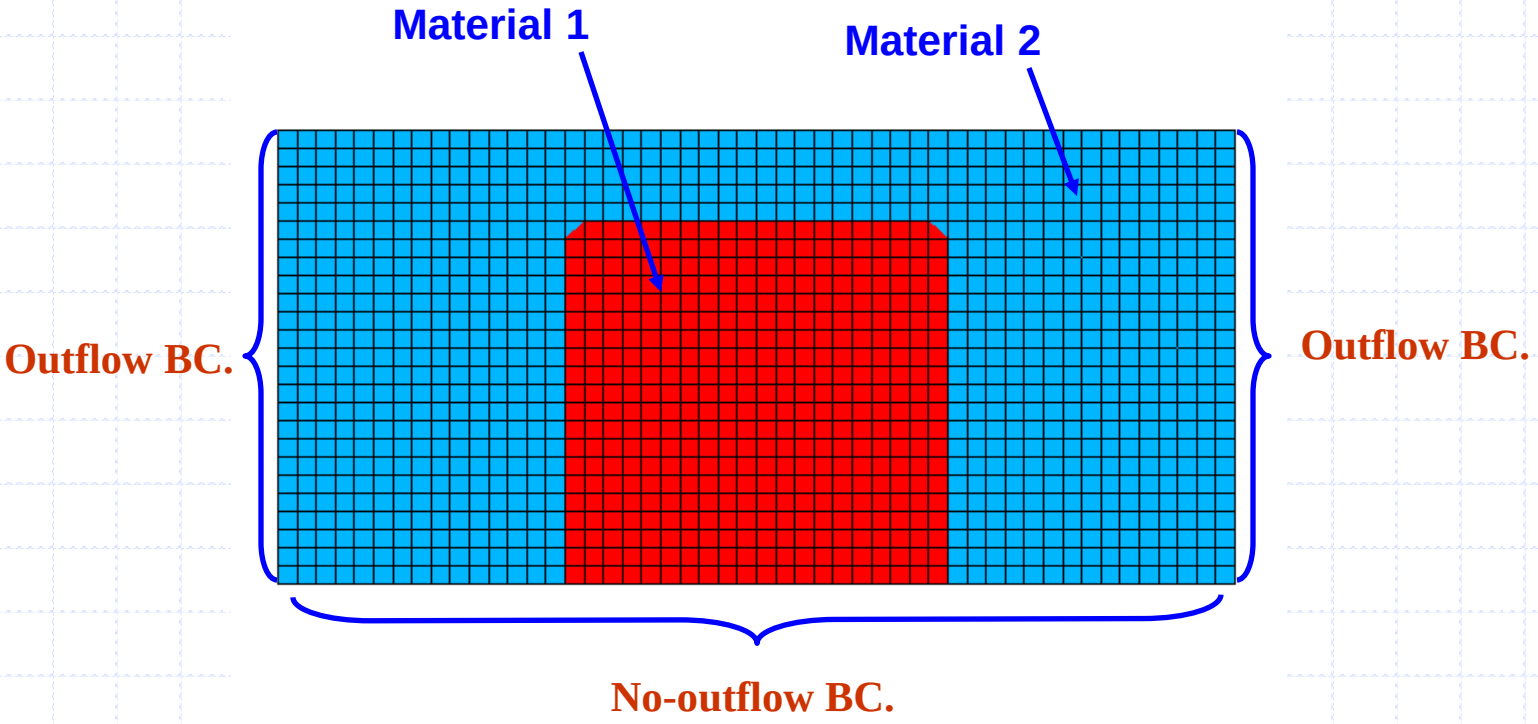
To track multi-material interfaces ➔ use

`*ALE_MULTI-MATERIAL_GROUP`

(5) Multi-Material ALE Formulation (fixed mesh)

(double-click on picture below)

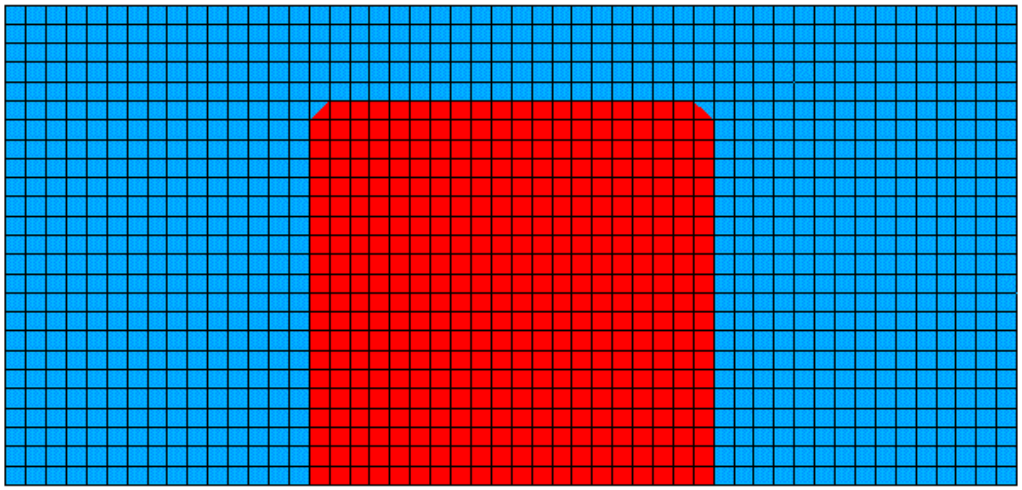
- The material flows through a fixed mesh.
- Each element is allowed to contain a mixture of different materials.



(5) Multi-Material ALE formulation

(double-click on picture below)

- **The material flows through a fixed mesh.**
- **Each element is allowed to contain one or more materials.**



Mesh Motion Allowed

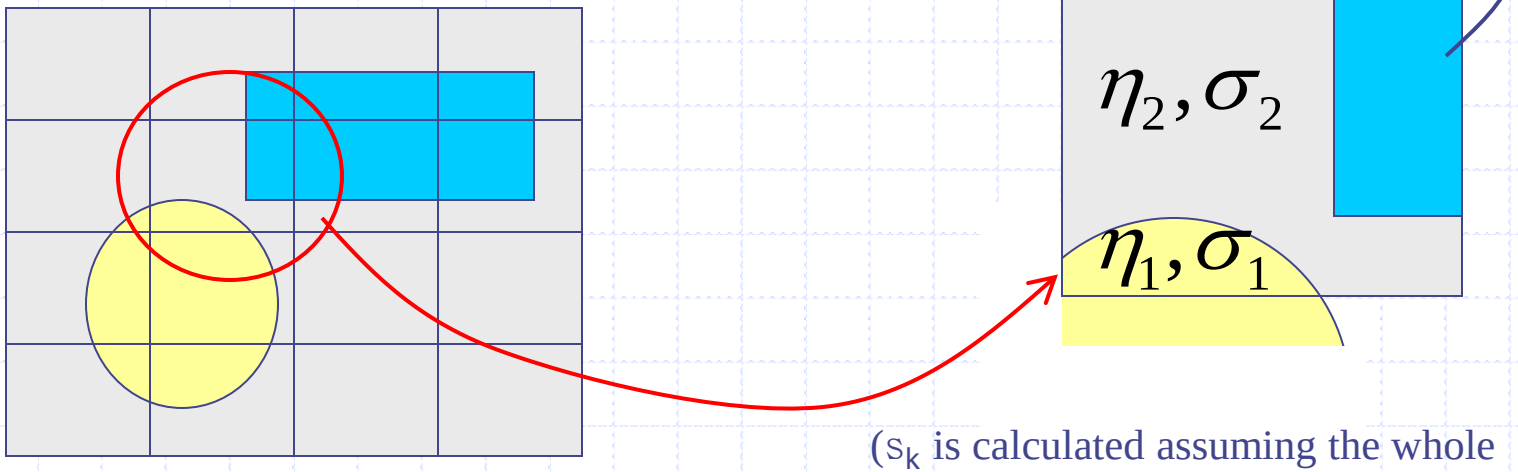
(5) Multi-Material ALE formulation
Element Stress Calculation

The composite stress, σ^*, is the volume fraction weighted average of the individual material group stresses, σ_k $k = [1, \text{nmat}]$.

η_i = Material Volume Fractions

$$\sigma^* = \sum_{k=1}^{\text{nmat}} \eta_k \sigma_k \qquad \sum_{k=1}^{\text{nmat}} \eta_k = 1$$

3 different materials



(s_k is calculated assuming the whole element is occupied by material "k")

(5) Multi-Material ALE formulation
Element Stress Calculation

The internal force vector is based on the composite stress tensor.

internal element force

$$f_i^e = \int_{V^e} \mathbf{B}^t \bar{\sigma}^* dV^e \approx \left| \text{reduced integration} \right| \approx$$

$$\approx \mathbf{B}^t \bar{\sigma}^* V^e_{(\xi_1,\xi_2,\xi_3)=(0,0,0)}$$

derivatives of shape functions

composite stress vector

element volume

[IV] MATERIAL DEFORMATION BEHAVIOR: EQUATION OF STATE & CONSTITUTIVE MODELS

(CAUTION: all material data in this section are fictitious!)

MATERIAL DEFORMATON BEHAVIOR

Material deformation is typically partitioned into 2 components: (1) a volume-preserving and (2) a volumetric components. In the former, deformations occur without changing the volume of the material. In the latter material volumetric deformations occur. The total stress tensor may be partitioned into 2 components: deviatoric stresses and pressure.

$$\sigma'_{ij} = \sigma_{ij} - \sigma_{kk}\delta_{ij} \quad \text{where} \quad \sigma_{kk} = [\sigma_{11} + \sigma_{22} + \sigma_{33}]/3 \propto P \quad \text{and}$$

$$\varepsilon'_{ij} = \varepsilon_{ij} - \varepsilon_{kk}\delta_{ij} \quad \text{where} \quad \varepsilon_{kk} = [\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}]/3 \propto \Delta\upsilon/\upsilon$$

A Constitutive Model (CM) relates $\Delta\sigma'_{ij}$ to $\Delta\varepsilon'_{ij}$.
A Equation of State (EOS) relates ΔP to $\Delta\upsilon/\upsilon$.
This is true whenever we use both the CM & EOS to describe a material. Some sample models will be considered as illustrations in the following.
***MANDATORY:* You must have correct physical data for the materials to expect any decent solution to your model! Arbitrary data may result in program crashing with unknown reasons!**

"FLUID-LIKE" MATERIAL MODEL ➔ *MAT_NULL

***MAT_NULL CM may be used for material with fluid-like deformation characteristics (air, water, etc.). Used by itself, this gives the viscous stress in the material. (The viscous stress is deviatoric in character.)**

$$\sigma^{v}{}_{ij} = \sigma'_{ij} = \mu * \dot{\varepsilon}'_{ij}$$

Used with an EOS, this gives the deviatoric stress component and the EOS provides the pressure component. Together they combine to give the total stress in the material.

$$\sigma_{ij} = \sigma'_{ij} + \frac{1}{3}\sigma_{kk}\delta_{ij} = \mu * \dot{\varepsilon}'_{ij} + P\delta_{ij}$$

$\dot{\varepsilon}'_{ij}$ = **Deviatoric strain rate (1/s)**

μ = **Mu = Dynamic viscosity (Pa*s)**

Some dummy sample input cards (users are always responsible for their own data!)

```
*MAT_NULL
$      MID       RHO0         PC         MU      TEROD      CEROD         YM         PR  ← Air (kg-m-s-K)
         1     1.1800       -1.0 1.7456E-5        0.0        0.0        0.0        0.0
*MAT_NULL
$      MID       RHO0         PC         MU      TEROD      CEROD         YM         PR  ← Water (kg-m-s-K)
         1     998.21      -10.0 0.8684E-3        0.0        0.0        0.0        0.0
```

"FLUID-LIKE" MATERIAL MODEL ➔ *MAT_NULL

The cut-off pressure, **PCUT** parameter in the ***MAT_NULL** card is used as a dilitation pressure limit (**negative pressure**) to reset the pressure in an element. This can actually be a very important variable. Material behavior in compression may be tested, but that in dilitation is very difficult to obtain! How much dilitation pressure does it take to cause a chunk of water to break apart (spalling)? The actual number is most of the time unknown! We can only make some educated guesses ➔ a **small negative value** comparing to atmospheric pressure in many cases (air & water, etc.).

The **dynamic viscosity**, mu (μ), is sometimes helpful in stabilizing the numerical computation of the material deformation.

The "rule of thumb" is "always build a model that best approximates your physical system".

MATERIAL MODELS: JOHNSON-COOK (1983)

Flow stress is a function of strain rate, plastic strain and Temperature.

$$\sigma_y = \sigma_y\left(\bar{\varepsilon}_p, \dot{\varepsilon}, T\right)$$

$$\sigma_y = \left[A + B\cdot\left(\bar{\varepsilon}_p\right)^n\right]\cdot\left[1 + C\cdot\ln\left(\dot{\varepsilon}^*\right)\right]\cdot\left[1 - \left(T^*\right)^m\right]$$

Plastic Strain Hardening | Rate dependent | Thermal Softening

Where {A, B, n} and {C} and {m} are material parameters.

$$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}$$

$$T = T_0 + \frac{e}{C_p}$$

$$T^* = \frac{T - T_0}{T_{melt} - T_0}$$

where

$\bar{\varepsilon}_p$ = **Equivalent plastic strain.**

$\dot{\varepsilon}^*$ = **Normalized plastic strain rate.**

$\dot{\varepsilon}_0$ = **1/s**

T^* = **Homologous temperature.**

MATERIAL MODELS: JOHNSON-COOK (1983)

Johnson-Cook model allows material to fail when its damage parameter, *D*=1

$$D = \sum \frac{\Delta\varepsilon}{\varepsilon^f}$$

$\Delta\varepsilon$ ⟸ Equivalent plastic strain increment

ε^f ⟸ Equivalent fracture strain

And ε^f is defined as

$$\varepsilon^f = \left[D_1 + D_2 e^{(D_3\sigma^*)}\right]\left[1 + D_4 \ln(\dot{\varepsilon}^*)\right]\left[1 + D_5 T^*\right]$$

where

$$\sigma^* = \frac{P}{\sigma_{eff}}$$

P ⟸ Pressure

σ_{eff} ⟸ Effective stress

***D* accounts for (1) loading, (2) strain rate and (3) thermal effects in the failure of the material.**

"VOID-LIKE" MATERIAL MODEL ➔ *MAT_VACUUM

***MAT_VACUUM is a new material model which may be used for void-like material. Mainly it allows void to be used with ALE multi-material element formulation (ELFORM=11). It provides convenient usage in the case if volume filling of a container where the void material may be modeled directly.**

```
*MAT_VACUUM
$      MID       RHO0          ← Air (kg-m-s-K)
         1     1.1800
```

Purpose:

Define vacuum in multi-material Eulerian element grids.

MID - Material ID

RHO0 - Ghost density

This density is not a real density. It is only used to avoid numerical problems (zero mass) if external forces are applied to nodes belonging to vacuum element.

EQUATION OF STATE

Each Equation Of State (EOS) relates the pressure (P) to the specific volume (vols), and temperature (T) of a material at a physical state, i.e. P = P(vols,T).

$$P = P_{cold} + P_{Thermal} = A(\mu) + B(\mu) \cdot e_{ipv0}$$

At any physical state, the vols identifies the compression state, and T identifies the thermal state of the material.

Since by definition:
A specific_volume = vols = volume/mass
A density = rho = mass/volume = 1/vols = 1/specific_volume.
A specific internal_energy = $e_i = C_v*T$ ~ energy/mass.
An EOS can also written as P = P(rho, e_i) = P(rho,T)

Be EXTRA CAREFUL with compression VS. dilitation behaviors of any material in your applications! They are not the same for many materials!

EQUATION OF STATE: IDEAL GAS

The perfect gas EOS can be given in many forms. A few are shown here just for clarity and also provide a check for the unit system used.

[1] Generally, the ideal gas EOS may be written as

Always check your units!

$$\left[\frac{N}{m^2}\right]\left[m^3\right] \sim PV = nR_uT \sim \left[mole\right]\left[\frac{J}{mole * K^o}\right]\left[K^o\right] \sim J$$

where R_u is the universal gas constant = 8.3144 J/(mole*K^o). A gas constant specific to a material can be defined as

$$\overline{R} = \frac{R_u}{\overline{M}}$$ **where** $\overline{M} = M/n =$ **Molecular mass of a species ~ kg/mole**

$\overline{R} =$ **Material-specific gas constant (per-mass unit)**

In an alternate form, *divide both sides of the above equation by mass*

$$\left[\frac{N}{m^2}\right]\left[\frac{m^3}{kg}\right] \sim P\frac{V}{M} = \frac{n}{M}R_uT \sim \left[\frac{mole}{kg}\right]\left[\frac{J}{mole * K^o}\right]\left[K^o\right] \sim \frac{J}{kg}$$

EQUATION OF STATE: IDEAL GAS

[2] Now rewrite it

$$P\upsilon = \bar{R}T \sim \left[\frac{J}{kg * K^o}\right]\left[K^o\right] \sim \frac{J}{kg}$$

where υ = specific volume

Recalling the relations among C_P & C_v & γ & $\bar{R}$ are

$$\bar{R} = C_p - C_v \rightarrow \frac{\bar{R}}{C_v} = \frac{C_p - C_v}{C_v} = [\gamma - 1] \rightarrow \bar{R} = [\gamma - 1]C_v$$

Rewriting (C_P & C_v & $\bar{R}$ all have per-mass unit in this form)

$$P = [\gamma - 1] * \frac{C_v T}{\upsilon} = [\gamma - 1] * \rho C_v T \rightarrow P = [\gamma - 1] * \frac{M}{V} C_v T = [\gamma - 1] * \frac{e_i}{V}$$

$\bar{e}$

[3] In the form used by LS-DYNA

$$P = [\gamma - 1] * \frac{[C_v T/\upsilon_0]}{[\upsilon/\upsilon_0]} = [\gamma - 1] * \frac{e_{ipv0}}{v_{relative}}$$

EQUATION OF STATE: IDEAL GAS

[4] Alternately

*from *MAT_NULL*

$$P = \rho \bar{R} T = \left(C_p - C_v\right)\left[\frac{\rho_0}{v_r}\right] T \sim \frac{N}{m^2}$$

where $v_r = \frac{\rho_0}{\rho} \Rightarrow \rho = \frac{\rho_0}{v_r}$

LS-DYNA provides at least 2 methods of defining an EOS for an ideal gas. For example, the following definition will initialize the gas to 1 atm (or 101,325 Pascal).

compute initial pressure from these values ➔ see above equation

```
*EOS_IDEAL_GAS
$     EOSID        Cp        Cv        C1        C2        T0       Vr0
          2     719.0    1006.0       0.0       0.0    298.00       1.0
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0   <= Air
*EOS_LINEAR_POLYNOMIAL
$     EOSID        C0        C1        C2        C3        C4        C5        C6
          3       0.0       0.0       0.0       0.0     0.400     0.400       0.0
$     EIPV0        V0
   253312.5       1.0
```

NOTE! The initial pressure of a material is often critical to its transient behavior. A user must use the equations to compute the appropriate eipv0 & V0 – always!

EQUATION OF STATE: IDEAL GAS

```
*EOS_IDEAL_GAS
$     EOSID        Cp        Cv        C1        C2        T0       Vr0
          2     719.0    1006.0       0.0       0.0    298.00       1.0
```

Purpose: **Modeling an Ideal Gas EOS.**

ID - **EOS ID**

CV & Cp - **Gas specific heat capacities.**

C1 & C2 - **T dependent coefficients of the heat capacities.**

T0 - **Initial temperature**

V0 - **Initial relative volume**

Note:

If used together with *BOUNDARY_AMBIENT_EOS, the prescribed internal energy curve is to be replace by a temperature curve.

$$P = P_{cold} + P_{Thermal} = A(\mu) + B(\mu) \cdot e_{ipv0}$$

EQUATION OF STATE: LINEAR-POLYNOMIAL MODEL

- **General Form:** $P = P_{cold} + P_{Thermal} = A(\mu) + B(\mu) \cdot e_{ipv0}$

$$P = \left[C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3\right] + \left[C_4 + C_5\mu + C_6\mu^2\right]e_{ipv0}$$

- **Perfect Gas can be modeled by letting**

$$C_0 = C_1 = C_2 = C_3 = C_6 = 0 \quad \text{AND} \quad C_4 = C_5 = \gamma - 1$$

$$P = (\gamma - 1)(1 + \mu)e_{ipv0}$$

$$(1 + \mu) = \frac{\rho}{\rho_0} = \frac{\upsilon_0}{\upsilon} = \frac{1}{\upsilon_{rel}}$$

$$P = (\gamma - 1)\frac{\rho}{\rho_0}e_{ipv0} = (\gamma - 1)\rho e$$

where

v_r = relative volume.

e_{ipv0} = internal energy/reference volume.

e = internal energy/mass.

Note that for perfect gas, P_0 can be defined by setting appropriate e_{ipv0}, and v_r OR by setting C_0 but NOT BOTH!

Definitions: $\eta = \frac{\upsilon_0}{\upsilon} = \frac{\rho}{\rho_0} = \frac{1}{\upsilon_{rel}} = \mu + 1$ **and** $\mu = \eta - 1 = \frac{\upsilon_0 - \upsilon}{\upsilon} = \frac{d\upsilon}{\upsilon}$

EQUATION OF STATE: LINEAR-POLYNOMIAL MODEL

Example of how to calculate a pressure (initial) condition for air:

```
*EOS_LINEAR_POLYNOMIAL
$     EOSID        C0        C1        C2        C3        C4        C5        C6
          3       0.0       0.0       0.0       0.0     0.400     0.400       0.0
$     EIPV0        V0
   253312.5       1.0
```

We can define pressure by defining 2 parameters:

[1] Internal energy per unit reference volume = $e_{ipv0} = \rho_0 e = \rho_0 C_V * T$

[2] Relative volume = $\upsilon_{rel} = \dfrac{\upsilon}{\upsilon_0} = \dfrac{\rho_0}{\rho}$

Then pressure is calculated by $P = (\gamma - 1)\dfrac{1}{\upsilon_{rel}} e_{ipv0}$

2 curves of $e_{ipv0}(t)$, $\upsilon_{rel}(t)$ **are used in the *BOUNDARY_AMBIENT_EOS card to calculate the pressure, P(t), in the elements which belong to a specified PID. This is shown in the airbag model shown in later example.**

Note: Specifying C_0 is not recommended when dealing with perfect gas or any equation requiring the use of "absolute P" (instead of gage P)!

EQUATION OF STATE: JWL for detonation products

- General Form:

$$P = A\left[1-\frac{\omega}{R_1\upsilon}\right]e^{-R_1V} + B\left[1-\frac{\omega}{R_2\upsilon}\right]e^{-R_2V} + \frac{\omega \cdot e_{V0}}{\upsilon}$$

For many high explosives (HE) a common EOS formulation is the Jones-Wilkins-Lee (JWL) EOS. It has 5 adjustable parameters. It can represent well many experiments (explosive cylinder test). This test uses a cylinder of copper filled with an HE, which is then initiated, and the expansion is recorded. The obtained expansion profile can be then used to calibrate the EOS model.

$A, B, R_1, R_2, \omega =$ Material property parameters.

$e_{V0} =$ Internal Energy/Initial_Volume

$\upsilon =$ Specific_Volume = Volume/Mass

EQUATION OF STATE: MIE-GRÜNEISEN MODEL

- **General Form:**

$$P = P_c + P_T = A(\mu) + B(\mu)\cdot e_{V0}$$

Cold part ~ Elastic interaction between atoms at 0K° isotherm.
Thermal part ~ Kinetic contribution due to molecular motion.

- **For Compression ($\mu = \eta - 1 > 0$):**

$$A(\mu) = \frac{\rho_0 C_0^2 \mu\left[2 + (2-\gamma_0)\mu - (\gamma_0 - a)\mu^2\right]}{2\left[1 - (S_1 - 1)\mu - S_2\frac{\mu^2}{(\mu+1)} - S_3\frac{\mu^3}{(\mu+1)^2}\right]^2}$$

$$B(\mu) = (\gamma_0 + a\mu)$$

- **For Tension ($\mu = \eta - 1 < 0$):**

$$A(\mu) = \rho_0 C_0^2 \mu$$

$$B(\mu) = (\gamma_0 + a\mu)$$

where $\eta = \frac{\upsilon_0}{\upsilon} = \frac{\rho}{\rho_0} = \frac{1}{\upsilon_{rel}} = \mu + 1$

and $\mu = \eta - 1 = \frac{\upsilon_0 - \upsilon}{\upsilon} = \frac{d\upsilon}{\upsilon}$

EQUATION OF STATE: MIE-GRÜNEISEN MODEL

Where

C_0 = the bulk sound speed.

S_1, S_2, S_3 = coefficients of the u_s-u_p curve.

$$u_s = C_0 + S_1 u_p + S_2 \left(\frac{u_p}{u_s} \right) u_p + S_3 \left(\frac{u_p}{u_s} \right)^2 u_p$$

For many materials, *LINEAR* u_s - u_p curve is quite adequate.

The initial or boundary condition for pressure can be set by substituting all material data input into the EOS.

EQUATION OF STATE: MIE-GRÜNEISEN MODEL

For example, a definition for water might look like this

```
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
         2    998.21    -10.0 0.8684E-3       0.0       0.0       0.0       0.0
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMA0         A     EIPV0
         2   1.647E3     1.921    -0.096       0.0     0.350       0.0 2.895E+5
$      VR0
       1.0
```

The e_{ipv0} value may be computed from a known P (in this case it is 1 atm or ~101325 Pascals) ➔ Assuming further that the relative volume or density of water is not changed very much from its reference value ➔ V_{R0} ~ 1.0 ➔ Putting all known data for water and these known information back into the Mie-Gruneisen EOS and compute e_{ipv0}!

Whether the initial P is important or not depending on the users' application range and scenario.

[IV] EXPLAINING COMMANDS RELATED TO FLUID-STRUCTURE-INTERACTION

*ALE_MULTI-MATERIAL_GROUP
*ALE_REFERENCE_SYSTEM_GROUP
*ALE_REFERENCE_SYSTEM_NODE
*ALE_REFERENCE_SYSTEM_CURVE
*ALE_REFERENCE_SYSTEM_SWITCH

*CONSTRAINED_LAGRANGE_IN_SOLID
*SET_MULTI-MATERIAL_GROUP_LIST

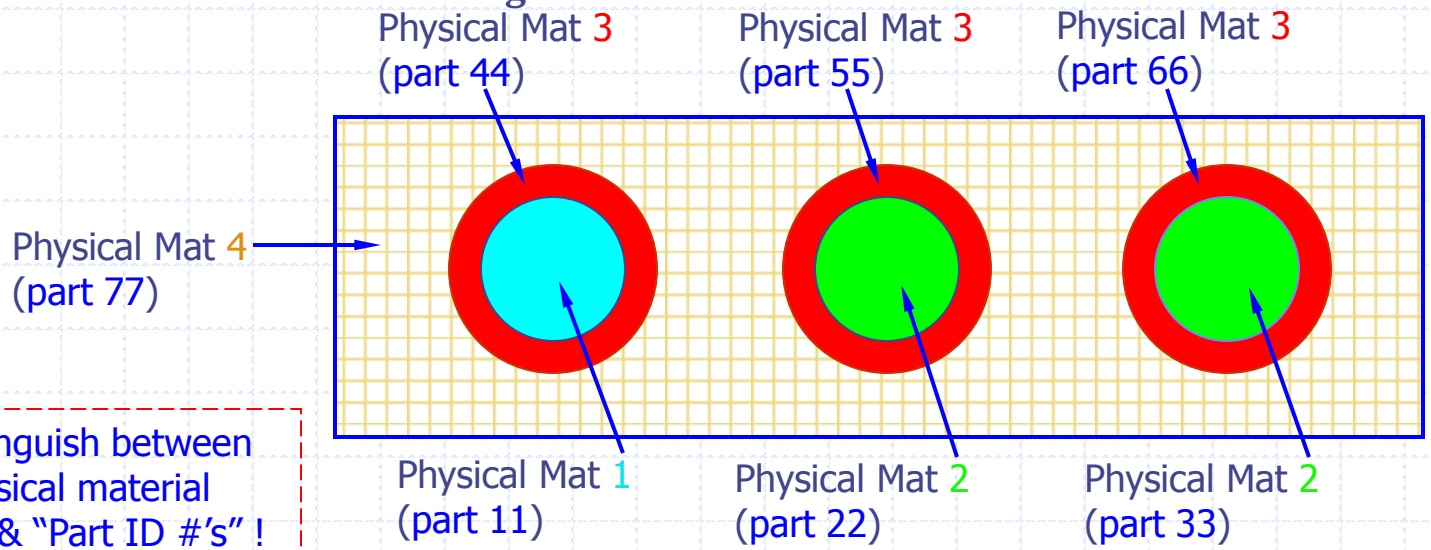
*CONTROL_ALE
*EOS_IDEAL_GAS
*MAT_VACUUM

ALE: Multi-Material Interfaces Tracking

***ALE_MULTI-MATERIAL_GROUP**
SID STYPE

SID Set ID **STYPE** Set type: 0=Part_Set, 1=Part

Consider a structure containing 3 containers containing 2 different physical materials (fluids 1, 2). The containers are made of the same, say, metal. Assume that these containers explode and spilling the fluids. *ALE_MULTI-MATERIAL_GROUP (AMMGID) defines the appropriate material grouping for treating multi-material elements & interface tracking.



Distinguish between "physical material #'s" & "Part ID #'s" !

ALE: Multi-Material Interfaces Tracking

APPROACH #1: Maintaining the interfaces for each part ID.

```
*ALE_MULTI-MATERIAL_GROUP
        11          1
        22          1
        33          1
        44          1
        55          1
        66          1
        77          1
```

Then, the interface of each part (11-77) will be tracked. This is, however, expensive due to the additional interface tracking computations, and not necessarily more accurate. As the same physical fluid , say fluid 2 from parts 2 and 3, flow into the same element, they behave as a single material. Thus tracking their interfaces may not be necessary.

ALE: Multi-Material Interfaces Tracking

APROACH #2: If we group the physical materials together.

```
*SET_PART
         1
        11
*SET_PART
         2
        22          33
*SET_PART
         3
        44          55          66
*SET_PART
         4
        77
*ALE_MULTI-MATERIAL_GROUP
         1          0
         2          0
         3          0
         4          0
```

AMMGSID=1
AMMGSID=2
AMMGSID=3
AMMGSID=4

Then, the interfaces of the 4 physical materials will be tracked.

ALE: Multi-Material Interfaces Tracking

NOTE:

- It is important to distinguish between the **material ID** or **part ID** from the **"physical" material**: the former is used for identification purpose and the latter is used in the physical computation.
- The same physical fluids coming from different part ID's into an empty element will "coagulate" and have no physical interfaces. However, the same physical solids coming from different part ID's into an empty element will not mix and will have interfaces. The ***ALE_MULTI-MATERIAL_GROUP** command should be used based on the physics of the problem.
- For example, if we want to model multiple solid particles of the same physical material, which collide with each other (powder compaction), we want to maintain the interfaces among the solid particles. Thus, APPROACH #1 in the previous slide should be used in this case.

ALE: Mesh Translation/Rotation/Expansion control

```
*ALE_REFERENCE_SYSTEM_GROUP
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        SID       STYPE      PRTYPE        PRID      BCTRAN       BCEXP       BCROT      ICOORD
         XC          YC          ZC      EXPLIM       DELAY
```

SID Set ID

STYPE Set type (0=Part_Set, 1=Part, 2=Node_Set, 3=Segment_Set).

PRTYPE Reference system type (More details on next slide ➔).

PRID ID of switch list, node group or curve group.

BCTRAN Mesh **translational** constraints.

BCEXP Mesh **expansion** constraints.

BCROT Mesh **rotational** constraints.

0=free	7=fixed x&y&z
1=fixed x	2=fixed y
3=fixed z	4=fixed x&y
5=fixed y&z	6=fixed z&x

ICOORD Flag for the definition of the center of mesh expansion and rotation
0: center of gravity **1**: at coordinate (XC,YC,ZC)

XC
YC
ZC
Coordinate defining center of mesh expansion and rotation.

EXPLIM Limit ratio for mesh expansion or shrinkage.

ALE: Mesh Translation/Rotation/Expansion control

PRTYPE

0 Eulerian.

1 Lagrangian.

2 Classical or normal ALE mesh smoothing
(see ***CONTROL_ALE** and ***ALE_SMOOTHING**).

3 Prescribed motion following ***load curves***
(see ***ALE_REFERENCE_SYSTEM_CURVE**).

4 Automatic mesh motion following ***mass weighted average velocity*** in ALE mesh.

5 Automatic mesh motion following coordinate system defined by ***three user-defined nodes***
(see ***ALE_REFERENCE_SYSTEM_NODE**).

6 ***Time-Switching*** control among different reference system types (Eulerian, ALE, etc.)
(see ***ALE_REFERENCE_SYSTEM_SWITCH**).

7 Automatic ***mesh expansion*** in order to enclose up to twelve user defined nodes
(see ***ALE_REFERENCE_SYSTEM_NODE**).

8 ***Delayed-ALE*** to control how much the mesh is to be moved during the remap step ➔
This option requires the definition of the 5th parameter in the 2nd card (called **DELAY**%).

ALE: Mesh Translation/Rotation/Expansion control

PRTYPE = 8 = DELAYED-ALE ***(this is still experimental!)***

For pure Eulerian method, the nodes of the deformed mesh (Lagrangian) always get remapped all the way back to the original Eulerian mesh position ➔ i.e. no delay in mesh remapping! ➔

DELAY is the fraction of mesh motion to be remapped to the original Eulerian mesh per dt.

DELAY =1= Pure Eulerian. **DELAY** =0= Pure Lagrangian.

Xmeshnew = XmeshLagrangian - **DELAY** *d

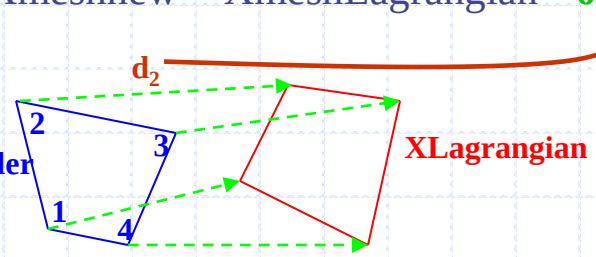
Remapping distance

Eulerian = **DELAY** =1.0 ➔ Xmeshnew = XmeshLagrangian - **1.00***d

ALE = **DELAY** =0.2 ➔ Xmeshnew = XmeshLagrangian - **0.20***d

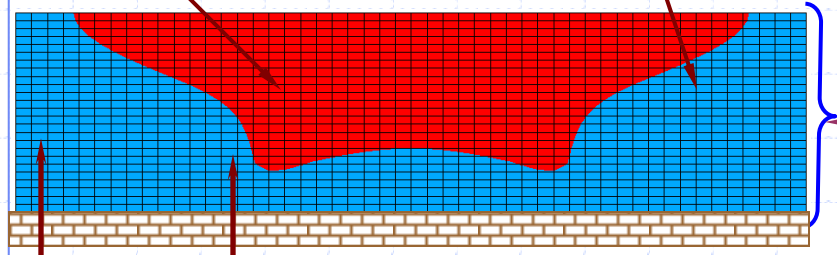
Lagrangian = **DELAY** =0.0➔ Xmeshnew = XmeshLagrangian - **0.00***d

$$d = \min(v_{\max}\Delta t, D * DELAY)$$




```
*ALE_REFERENCE_SYSTEM_GROUP
$       SID       STYPE      PRTYPE       PRID      BCTRAN       BCEXP       BCROT      ICOORD
                                  8
$        XC          YC          ZC     EXPLIM       DELAY
                                                      0.20
```

(5) Multi-Material ALE Formulation (varying mesh)



Control interfaces tracking between material groups

*ALE_MULTI-MATERIAL_GROUP

PSID 1

```
*SET_PART
         1
         1                    2
```

Magnitude of mesh expansion

```
$ ASSIGN REF SYS TYPE TO SETID=1; SETTYPE=0=PSID; PRTYPE=4=REFSYSTYP=mesh_ave_v
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
*ALE_REFERENCE_SYSTEM_GROUP
$       SID     STYPE     PRTYP      PRID    BCTRAN     BCEXP     BCROT    ICOORD
          1         0         4         0         3         6         7         0
$        XC        YC        ZC    EXPLIM
         0         0         0       1.3
```

PRTYP=4=Auto mesh motion following Ave mass

BCTRAN=3=No mesh translation in z-dir

BCEXP=6=No mesh expansion in z&x dir

BCROT=7=No mesh rotation

ALE: Mesh Translation Control

```
*ALE_REFERENCE_SYSTEM_NODE
      NSID
      NID1      NID2      NID3      NID4      NID5      NID6      NID7      NID8
      NID9     NID10     NID11     NID12
```

ID — Node_Set ID

NID1… NID12 — User specified nodes

To define a Node-Group that controls the motion of an ALE mesh (to be used with *ALE_REFERENCE_SYSTEM_GROUP)

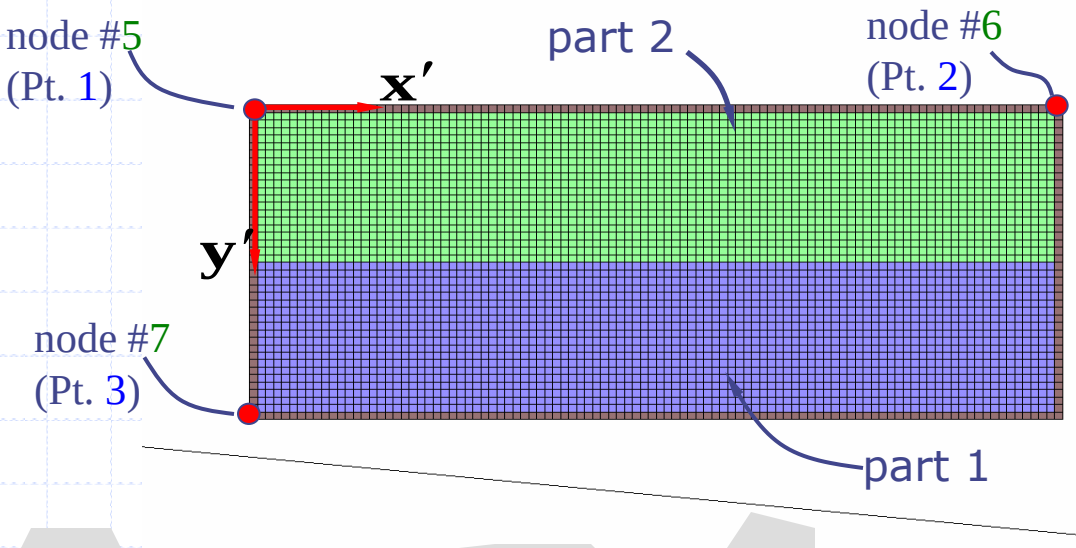
For PRTYPE=5 the ALE mesh is forced to follow the motion of a coordinate system, which is defined by three nodes (NID1,NID2,NID3). These nodes are located at, x_1, x_2, x_3, respectively. The axes of the coordinate system are defined as:

$$\mathbf{x}' = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|} \qquad \mathbf{z}' = \frac{\mathbf{x}' \times (\mathbf{x}_3 - \mathbf{x}_1)}{|\mathbf{x}' \times (\mathbf{x}_3 - \mathbf{x}_1)|} \qquad \mathbf{y}' = \mathbf{z}' \times \mathbf{x}'$$

For PRTYPE=7, the ALE mesh is forced to move and expand, so as to enclose up to twelve user defined nodes (NID1…NID12).

ALE: Reference System Moves with 3-nodes

ALE mesh motion following a coordinate system defined by 3 use-defined nodes



$$\mathbf{x}' = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\,\mathbf{x}_2 - \mathbf{x}_1\,|}$$

$$\mathbf{z}' = \frac{\mathbf{x}' \times (\mathbf{x}_3 - \mathbf{x}_1)}{|\,\mathbf{x}' \times (\mathbf{x}_3 - \mathbf{x}_1)\,|}$$

(z' is an axis pointing into-paper)

$$\mathbf{y}' = \mathbf{z}' \times \mathbf{x}'$$

```
*ALE_REFERENCE_SYSTEM_GROUP
         1          0          5          1
*SET_PART_LIST
         1
         1          2
*ALE_REFERENCE_SYSTEM_NODE
         1
         5          6          7
```

PRTYPE=5: SID follows 3-nodes

PRID=1: System-Node-Group number

PSID 1 contains PID 1 & 2

NSID of the 3-nodes ref. System which contains these 3 nodes.

ALE: Mesh Translation Control

```
*ALE_REFERENCE_SYSTEM_CURVE
$ CURVESID
$    LCID1     LCID2     LCID3     LCID4     LCID5     LCID6     LCID7     LCID8
$    LCID9    LCID10    LCID11    LCID12
```

ID — Curve set ID

LC1 = f_1
:
LC12=f_{12} — Load curve ID's

To prescribe the motion of an ALE mesh (used with *ALE_REFERENCE_SYSTEM_GROUP)

The velocity of a node at coordinate (x_1, y_2, z_3) is defined as:

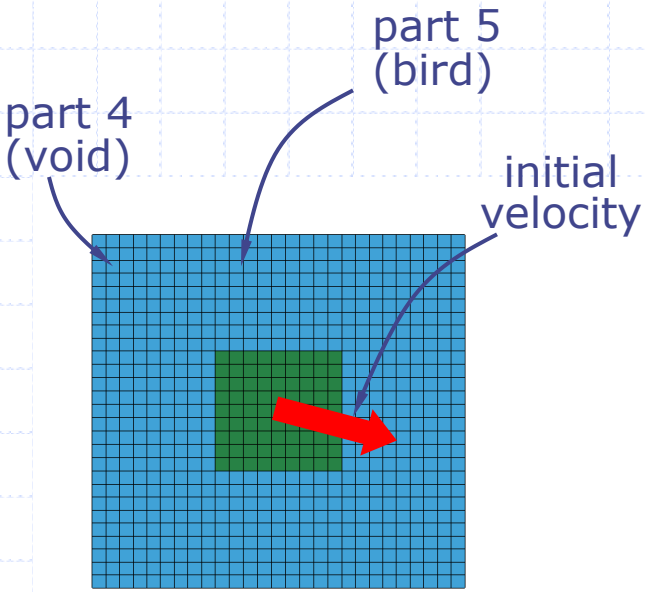
Translation Rotation

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_5 \\ f_9 \end{Bmatrix} + \begin{bmatrix} f_2 & f_3 & f_4 \\ f_6 & f_7 & f_8 \\ f_{10} & f_{11} & f_{12} \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

$f_1(t)$ is the value of load curve **LC1** at time t, etc.

ALE: Mesh Translation Control

Example of prescribed ALE mesh motion following a set of pre-defined load curves




```
*ALE_REFERENCE_SYSTEM_GROUP
$       SID       STYPE       PRTYP        PRID
                      1           0           3           1
$        XC          YC          ZC     EXPLIM
                      0           0           0           0
*ALE_REFERENCE_SYSTEM_CURVE
$ CURVESID
                      1
$     LCID1       LCID2       LCID3       LCID4       LCID5
                      1           0           0           0           2

*SET_PART_LIST
1
4       5
*DEFINE_CURVE
1
0.0,   5.0
0.15,  5.0
0.16,  0.0
5.0,   0.0
*DEFINE_CURVE
2
0.0, -1.0
0.15,-1.0
0.16,  0.0
5.0,   0.0
```

curve ID for $f_1 = V_x$

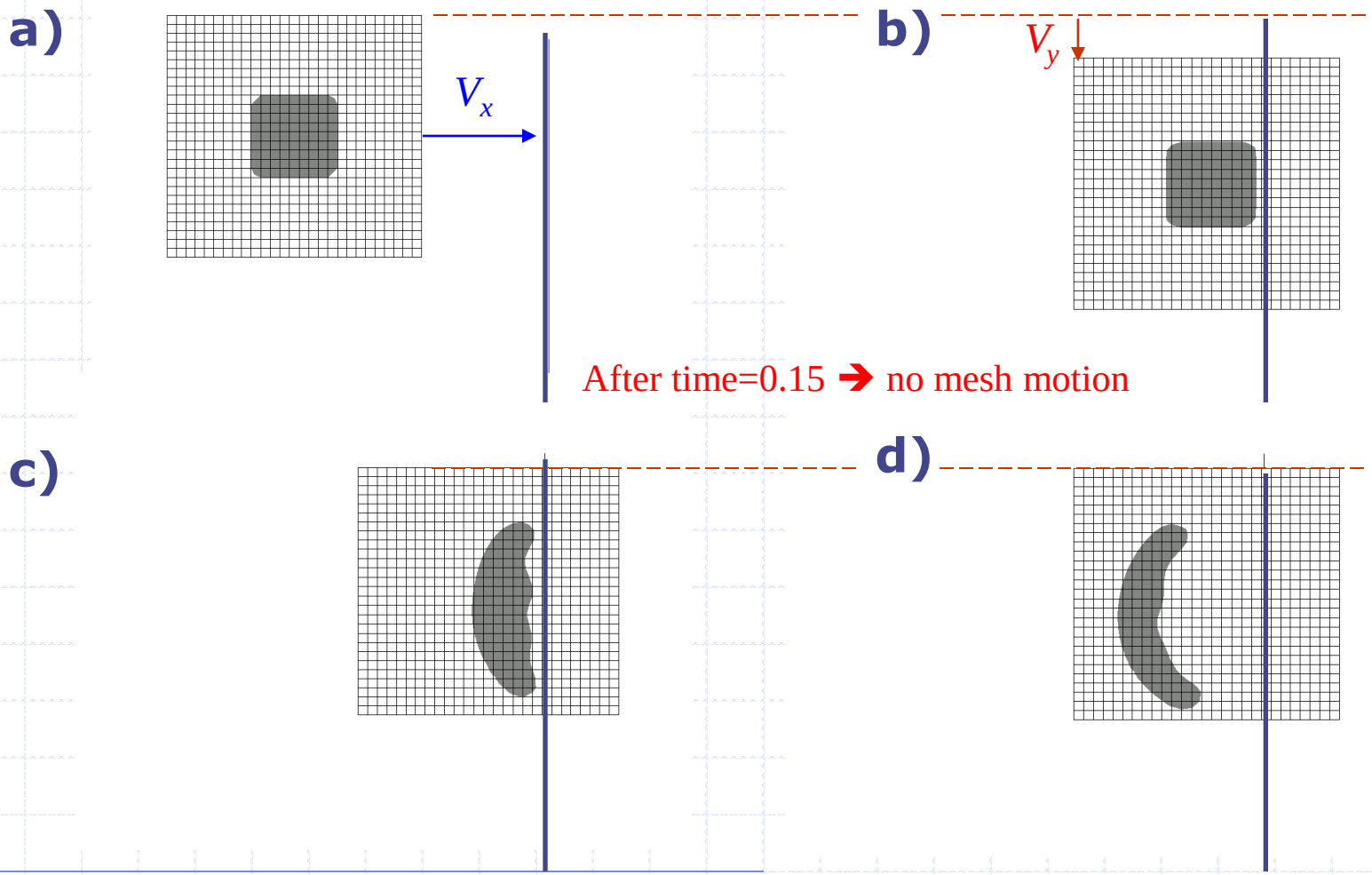
curve ID for $f_5 = V_y$

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_5 \\ 0 \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

After time=0.15 ➔ no motion

ALE: Mesh Translation Control

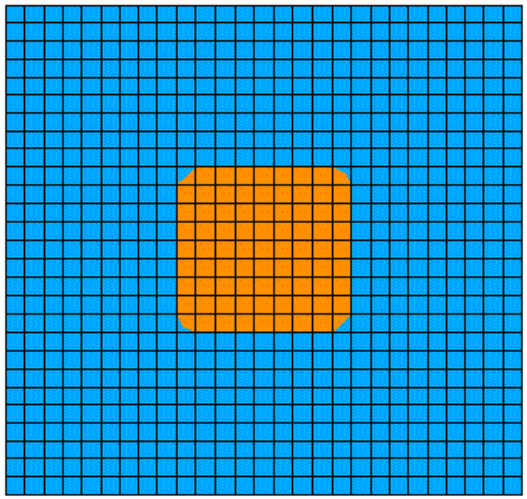
Example of prescribed ALE mesh motion following a set of pre-defined load curves



ALE: Mesh Translation Control

Example of prescribed ALE mesh motion following a set of pre-defined load curves

Double-Click on the picture …➔



ALE: Reference System Time-Switching Control

```
*ALE_REFERENCE_SYSTEM_SWITCH
 SWITCH_ID
        T1         T2         T3         T4         T5         T6         T7
  SYSTYPE1   SYSTYPE2   SYSTYPE3   SYSTYPE4   SYSTYPE5   SYSTYPE6   SYSTYPE7   SYSTYPE8
NDCURVID1  NDCURVID2  NDCURVID3  NDCURVID4  NDCURVID5  NDCURVID6  NDCURVID7  NDCURVID8
```

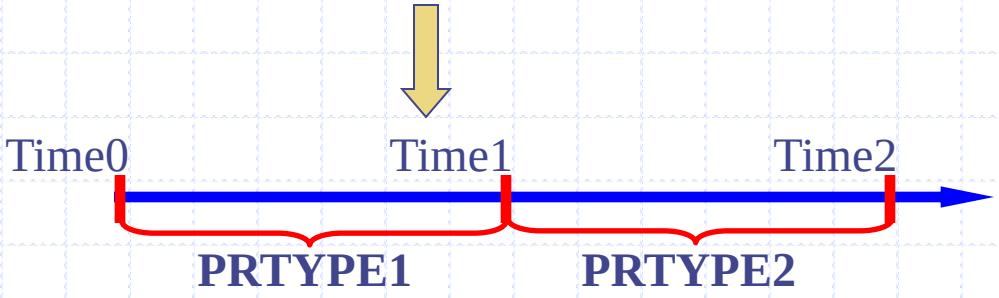
ID — Switch list ID

T1 . . . T7 — Times for switching reference system type

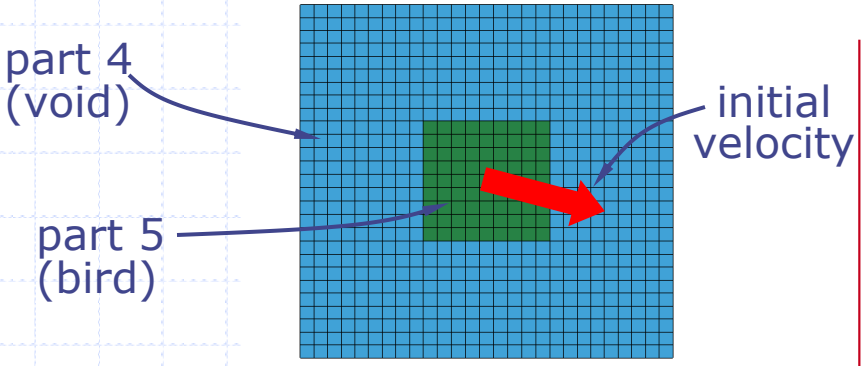
TYPE1 . . . TYPE8 — Reference system types

ID1 . . . ID8 — ID of node or curve group (**PRTYPE=3,5** or **7**)

At time **T1** the reference system type is switched from **PRTYPE1** to **PRTYPE2** etc. (Please see ***ALE_REFERENCE_SYSTEM_GROUP** for information about the different reference system types, i.e. **PRTYPE** parameter.)



ALE: SwitchingMotion of Reference System (a) Moves with Average Mass Flow & (b) Fixed Mesh



PRTYPE=6 ➔
Reference System is switched from one type to another according to a time table provided by the load curve PRID.

Example of switching between reference system types:

```
*ALE_REFERENCE_SYSTEM_GROUP
1         0          6        1234          0          0          7

*ALE_REFERENCE_SYSTEM_SWITCH
1234
0.1      0.15       0.25       0.30       10.0
4          0          4          0          4

*SET_PART_LIST
1
4          5
```

PRID Translate Expand Rotate

Time

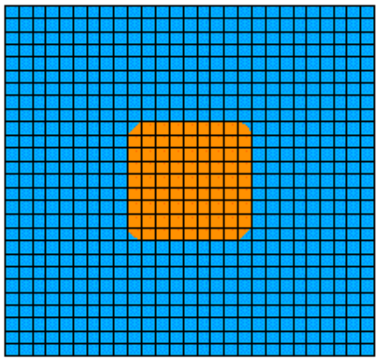
System type

Mesh motion following Average mass flow (PRTYPE=4)

Eulerian description of motion: fixed mesh (PRTYPE=0)

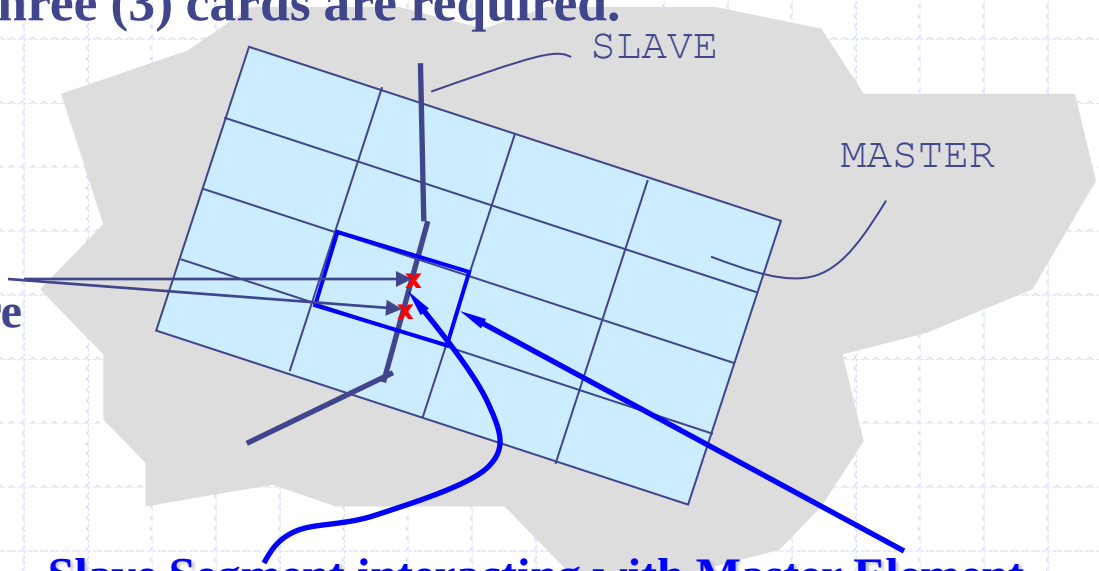
ALE: SwitchingMotion of Reference System
(a) Moves with Average Mass Flow & (b) Fixed Mesh

Double-Click on
the picture …➔



```
*CONSTRAINED_LAGRANGE_IN_SOLID
     SLAVE     MASTER     SSTYPE     MSTYPE      NQUAD      CTYPE      DIREC      MCOUP
   START        END       PFAC       FRIC    FRACMIN       NORM  ISEGNORM      XDAMP
      Cq       HMIN       HMAX      ILEAK      ILEAK      PLEAK      LCPOR
```

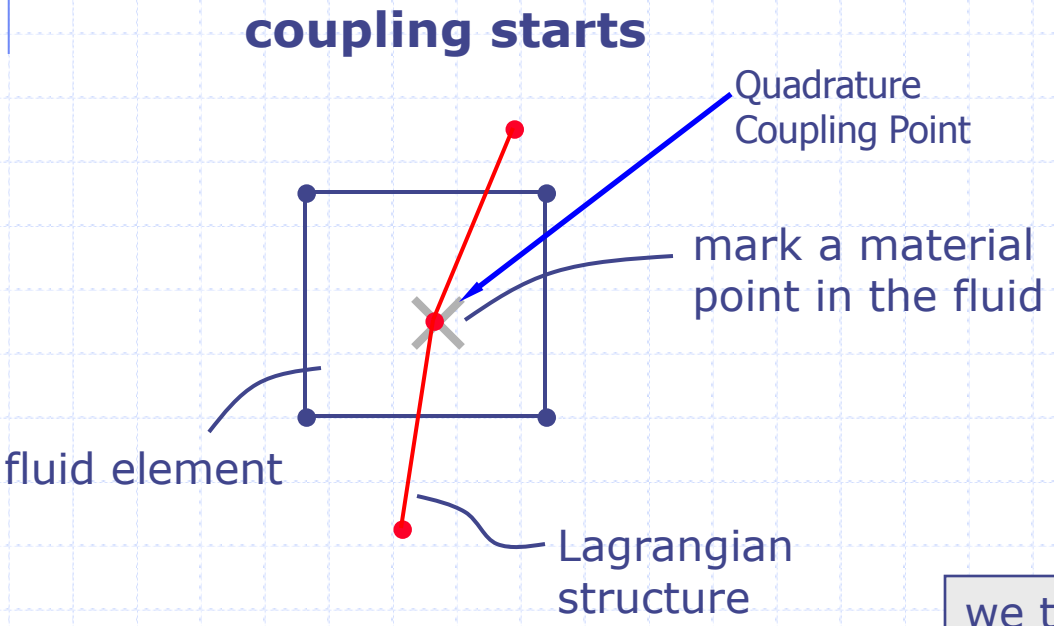
This is the main command to couple the Lagrangian structures (SLAVE) to Eulerian or ALE structures (MASTER). Currently Lagrangian shell, beam and solid elements can be coupled. A collection of shells can be grouped into a Segment_Set and the SGSID can be used in the coupling. Currently three (3) cards are required.

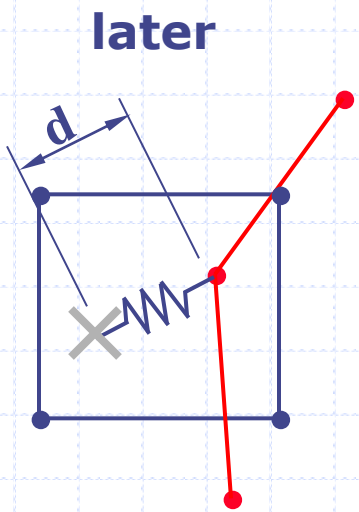


Slave Segment interacting with Master Element

Penalty-Based Coupling Formulation

- The penalty based algorithm tracks the relative displacement between fluid and the structure.
- Nodal forces, proportional to the magnitude of the relative displacements, are applied forcing the fluid(s) and structure(s) to provide the interactions.
- The method conserves energy but it is not as stable as the constraint-based method.





we trace the material point, ×, and apply a coupling force proportional to **d.**

```
*CONSTRAINED_LAGRANGE_IN_SOLID
     SLAVE     MASTER     SSTYPE     MSTYPE      NQUAD      CTYPE      DIREC      MCOUP
     START       END       PFAC       FRIC    FRACMIN       NORM  ISEGNORM      XDAMP
        Cq      HMIN       HMAX      ILEAK      ILEAK      PLEAK      LCPOR
```

CARD 1:

SLAVE Slave ID.

MASTER Master ID.

SSTYP Slave ID type : 0=PSID; 1=PID; 2=SGSID.

MSTYP Master ID type : 0=PSID; 1=PID.

NQUAD Number of quadrature coupling points on a Lagrangian segment.

CTYPE Coupling type:

1: Constrained acceleration

2: Constrained acceleration and velocity (default)

3: Constrained acceleration and velocity normal direction only.

4: Penalty coupling for Lagrangian **shell** & **solid** elements.

5: Penalty coupling for Lagrangian **solid** elements, with erosion.

DIREC=1 ← **6**: Special penalty coupling for airbag application (new, less robust).

```
*CONSTRAINED_LAGRANGE_IN_SOLID
     SLAVE     MASTER     SSTYPE     MSTYPE      NQUAD      CTYPE      DIREC      MCOUP
     START       END       PFAC       FRIC    FRACMIN       NORM   ISEGNORM      XDAMP
        Cq       HMIN       HMAX      ILEAK      ILEAK      PLEAK      LCPOR
```

CARD 1 (cont.):

DIREC Coupling direction choice:

1: Coupling in normal direction - in compression & tension (default).

2: Coupling in normal direction - in compression only (robust).

3: Coupling in all directions.

MCOUP Multi-material groups coupling:

=0: Coupling with all material groups ➔ **ILEAK** is turned off.

=1: Coupling with material with highest density only (more robust).

<0: If this is a negative integer (example: **–112**) then an

AMMSID =**112** will contain the master fluid(s) to be coupled to. This SET-ID **AMMSID** is defined by a

***SET_MULTI-MATERIAL_GROUP_LIST** card.

```
*CONSTRAINED_LAGRANGE_IN_SOLID
     SLAVE     MASTER     SSTYPE     MSTYPE      NQUAD      CTYPE      DIREC      MCOUP
     START       END       PFAC       FRIC    FRACMIN       NORM  ISEGNORM      XDAMP
        Cq      HMIN       HMAX      ILEAK      ILEAK      PLEAK      LCPOR
```

CARD 2:

START Time to start coupling.

END Time to end coupling.

PFAC Penalty factor (for CTYPE=4, 5, 6), default=0.1.

If this is a **possitive integer** ➔ it is the % of estimated critical stiffness.

If this is a **negative integer** (example: **–112**)➔ then 112 is a load curve of **P VS. penetration_depth** is given (LCID=**112**) for coupling force calculation: abscissa=x= penetration_depth, ordinate=y=maximum pressure on Lagrangian segments.

FRIC Coefficient of friction (for DIREC=2 only).

FRCMIN Minimum volume fraction of a fluid in an element to start coupling (MCOUP=1 only, 0.3 may be a good value, default is 0.5).

***CONSTRAINED_LAGRANGE_IN_SOLID**

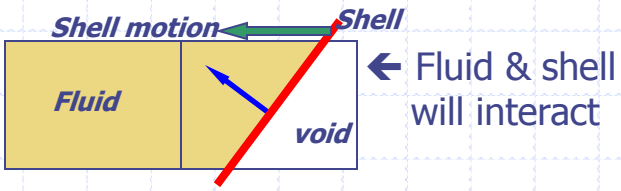
SLAVE	MASTER	SSTYPE	MSTYPE	NQUAD	CTYPE	DIREC	MCOUP
START	END	PFAC	FRIC	FRACMIN	NORM	ISEGNORM	XDAMP
Cq	HMIN	HMAX	ILEAK	ILEAK	PLEAK	LCPOR	

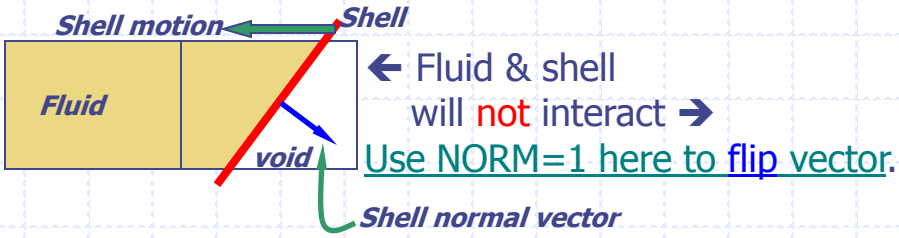
CARD 2 (cont.):

NORM Shell element and segment normal vector flag:

0: Use right-hand rule to find surface normal vector (default).

1: Use left-hand rule to find surface normal vector .





ISEGNORM Coupling direction flag: =0= interpolation based on **nodal normals**; =1=interpolation of coupling direction based on **segment normals**.

XDAMP Coupling damping frequency in terms of % of critical system frequency(testing phase).

```
*CONSTRAINED_LAGRANGE_IN_SOLID
     SLAVE     MASTER     SSTYPE     MSTYPE      NQUAD      CTYPE      DIREC      MCOUP
     START       END       PFAC       FRIC    FRACMIN       NORM  ISEGNORM      XDAMP
        Cq      HMIN       HMAX      ILEAK      ILEAK      PLEAK      LCPOR
```

CARD 3:

CQ Equivalent heat transfer coefficient.

HMIN Minimum air gap in heat transfer.

HMAX Maximum air gap in heat transfer.

ILEAK Coupling leakage control flag. 0: Off (default).
1: Leakage control is turned off if volfrac > FRCMIN+0.1 (**FRCMIN=0.3**)
2: Leakage control is turned off if volfrac > FRCMIN+0.3 (**FRCMIN=0.3**)

PLEAK How much to additionally stretch the coupling "spring" to prevent leakage. (% of coupling distance to fluid surface from the end of coupling "springs", 0 < PLEAK < 0.2).

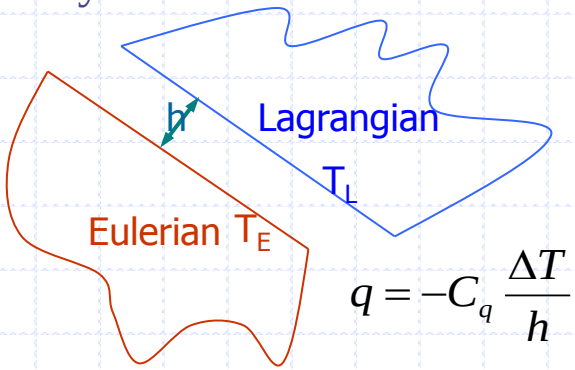
LCPOR Load curve ID for porous flow through coupling segment:
abscissa=x=coupling-P, ordinate=y=porous_fluid_velocity.

***CONSTRAINED_LAGRANGE_IN_SOLID**

SLAVE	MASTER	SSTYPE	MSTYPE	NQUAD	CTYPE	DIREC	MCOUP
START	END	PFAC	FRIC	FRACMIN	NORM	ISEGNORM	XDAMP
Cq	HMIN	HMAX	ILEAK	ILEAK	PLEAK	LCPOR	

CARD 3 (notes):

CQ, **HMIN, HMAX** are defined for a heat transfer interface between the Eulerian master part(s) and the Lagrangian slave part(s) assuming there is a gap at their intereface. This is a preliminary means for heat transfer **"coupling"** effect.



$$q = -C_q \frac{\Delta T}{h}$$

LCPOR: The coupling pressure is defined as

$$P_{coupling} \equiv (P_{in} - P_{out}) = \frac{F_{coupling}}{Area_{coupling}}$$

where P_{in} and P_{out} are the pressures inside and outside of the coupling segment.

```
*CONSTRAINED_LAGRANGE_IN_SOLID
     SLAVE     MASTER     SSTYPE     MSTYPE      NQUAD      CTYPE      DIREC      MCOUP
     START       END       PFAC       FRIC    FRACMIN       NORM  ISEGNORM      XDAMP
        Cq       HMIN       HMAX      ILEAK      ILEAK      PLEAK      LCPOR
```

SOME TIPS:

- **NQUAD** is defined based on the relative mesh sizes between the Lagrangian & Eulerian meshes. Use **NQUAD**=3 per Eulerian element.
- **CTYPE**=4 works with Lagrangian shell and solid; **CTYPE**=5 to couple with Lagrangian solid with erosion; **CTYPE**=6 for airbag only (DIREC=1).
- Most robust: **DIREC**=2 ➔ coupling in compression only.
- For a start, use default values for **PFAC** and **FRIC** ➔ they can be "tweaked" later in data validation phase.
- **FRACMIN** is the minimum fluid volume fraction in an element when the coupling interaction between the fluid and the structure is turned on. The lower it is, the earlier it is turned on.
- **Always** check the normal vectors of the slave shell elements to **make sure** that **all** their normal vectors point into the interacting fluid. Otherwise, turn **NORM**=1.
- When turning on leakage control: **ILEAK**=1, use **FRCMIN**=0.3.

ALE: GROUPING PHYSICAL MATERIAL

```
*SET_MULTI-MATERIAL_GROUP_LIST
12345678901234567890123456789012345678901234567890123456789012345678901234567890
   AMMGSID
   AMMGID1    AMMGID2    AMMGID3    AMMGID4   . . .
```

AMMGSID ALE MULTI-MATERIAL GROUP SET ID (integer)

AMMGID# ALE MULTI-MATERIAL GROUP ID #

These AMMGID#'s are defined by the

***ALE_MULTI-MATERIAL_GROUP** card ➔

NOT JUST PART ID!

This card allows for the selection of particular ALE-Multi-Material(s) as master material(s) to be used in the coupling.

AMMGSID is a positive integer ➔ [-AMMGSID] = negative integer is used in MCOUP in the 1st card of the *CONSTRAINED_LAGRANGE_IN_SOLID card.

ALE: GROUPING PHYSICAL MATERIAL

Example: Consider a fuel tank sloshing example …

```
*SET_PART_LIST                          PIDs
  PSID→10
        11              32              43
*ALE_MULTI-MATERIAL_GROUP
        11               1          ← AMMGID=1 = gas      inside  a tank
        32               1          ← AMMGID=2 = gasoline inside  a tank (liquid)
        43               1          ← AMMGID=3 = air      outside a tank
*SET_MULTI-MATERIAL_GROUP_LIST
$  AMMGSID
        12
$  AMMGID1    AMMGID2    AMMGID3    AMMGID4    AMMGID5    AMMGID6    AMMGID7    AMMGID8
         1          2
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE     MASTER    SLVSTYP      MSTYP      NQUAD       CTYP        DIR      MCOUP
         9         10          0          0          4          4          2        -12
$     START       END       PFAC       FRIC      FRCMN       NORM
         0         0        0.1        0.0        0.3          0
$        CQ      HMIN       HMAX      ILEAK      PLEAK      LCPOR
         0         0          0          1          0          0
```

ALE: GROUPING PHYSICAL MATERIAL

Example: Consider a fuel tank sloshing example …

Where PID 11 & 32 are the gas & liquid inside the tank, respectively, and PID 43 is the air outside the tank.

PID's 11, 32, 43 are AMMGID's 1, 2, 3, respectively.

In the coupling card, the "MASTER" SETID is PSID 10. This PSID includes the *geometrical space* of all 3 parts. It tells LS-DYNA to search for coupling in all of these mesh spaces.

The MCOUP is set to "–12" where 12 is the ALE-multi-material-SETID (AMMSID) which includes the 1st and 2nd ALE_multi-material-ID's (AMMGID) only. This tells LS-DYNA to consider coupling only to those materials excluding the 3rd AMMGID which is the air outside.

ALE: GROUPING PHYSICAL MATERIAL

```
*CONTROL_ALE
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      DCT      NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
    START       END     AAFAC      VFAC     VLIM→PRIT       EBC  PAMBIENT   EBCXNSID
```

Modification:

The obsolete VLIMIT (card 2, colume 5) is replaced by PRIT. PRIT is a flag for pressure equilibrium iterations in mixed multi-material elements. By default, all materials inside a multi-material cell are (per definition) exposed to the same volumetric strain rates. This can create problems when mixing materials with very different bulk stiffnesses.

Setting PRIT=1, the code switches from equal volumetric strain rates to equal pressure. That is, with PRIT=1 all materials inside an element are assumed to have the same pressure.

PRIT - Pressure iteration flag

- **Eq.0 - Not activated (uniform volumetric strain rate inside multi-material elements)**
- **Eq.1 - Activated (uniform pressure inside multi-material elements)**

ALE: GROUPING PHYSICAL MATERIAL

```
*CONTROL_ALE
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      DCT       NADV       METH       AFAC       BFAC       CFAC       DFAC       EFAC
    START        END      AAFAC       VFAC      VLIM       PRIT        EBC   PAMBIENT  EBCXNSID
```

Addition:

The **PAMBIENT** parameter has been added to define the nominal $P_{AMBIENT}$

PAMBIENT – Ambient Pressure value (Pascal, psi, etc.)

The effect of $P_{AMBIENT}$ is taken out of the total stress tensor ➔ $\hat{\sigma}^{-}{}_{ij} = \sigma^{-}{}_{ij} - P_{AMBIENT} * \delta_{ij}$

The model is integrated in time ➔ $\hat{\sigma}^{-}{}_{ij} \Rightarrow \hat{\sigma}^{+}{}_{ij}$

Then the $P_{AMBIENT}$ is added back to the updated stresses ➔ $\sigma^{+}{}_{ij} = \hat{\sigma}^{+}{}_{ij} + P_{AMBIENT} * \delta_{ij}$

This takes out the dynamic effects due to $P_{AMBIENT}$.

EBC = Global Eulerian BC's: 0=off; 1=on with stick; 2=on with slip BC.

If EBC<0 ➔ when the nodal normals of the adjacent segments have angles larger than the specified acos(abs(-EBC)) then the node BC is applied.

EBCXNSID = NSID containing nodes to be ***excluded*** from the EBC Eulerian BC's.

ALE: GROUPING PHYSICAL MATERIAL

```
* INITIAL_VOLUME_FRACTION
$      EID       VF1       VF2       VF3       VF4       VF5       VF6       VF7
```

This card defines initial volume fractions of different materials in multi-material ALE element formulation. It used to allow only 2 AMMGID (ALE multi-material group ID) ➔

Addition:

The number of VF-columns are extended from 2 to 7. Thus, we now can handle more than 3 multi-material groups.

[VI] EXAMPLES

Illustrative Examples

In the following, the overall LS-DYNA modeling steps will be given in details in the first few examples. Subsequently, we will focus on the input definitions which control and/or affect the fluid structure interactions (FSI) processes.

Note:

- **The following examples are not meant to be (1) numerically accurate, (2) comparable to each other.**
- **They are used only to illustrate the steps in FSI model constructions.**
- **Some may have non-physical material properties (adjusted to make the computations faster.)**
- **The goal is only to convey the methodology used.**

The approach taken here asks the following questions:

1. What is the physical problem we are trying to simulate?
2. What are the LS-DYNA commands which can be used to describe this physical system?

We may divide the input file into the following sections:

- Define the basic geometry of the system.
- Define/organize the parts which build up the total model.
- Define the material properties that control deformations.
- Define the initial & boundary conditions.
- If dealing with multi-materials (multi-fluids), define the ALE characteristics of the subsystems involved – Think of this as the rules for treating the ALE|Eulerian material systems.
- Define the interaction constraints among the interacting fluids (ALE or Eulerian, master parts) and structures (Lagrangian, slave parts).

Modeling Procedure ⟺ LS-DYNA Commands

- Set up the control parameters for Execution & Computation options (*CONTROL_).
- Set up the control parameters for Output options (*DATABASE_).
- Set up the Basic Geometry – This is a Pre-Processor function (*NODE, *ELEMENT_SOLID, *ELEMENT_SHELL, etc.) .
- Organize the Components of the model for interactions:
  Define a *PART with its corresponding *SECTION_ properties, Constitutive, EOS, and Hourglass models, etc.
  Define the *SECTION_ properties (element formulation, etc.) .
  Define the material behavior associated with each part using *MAT_.
- Organize the parts into part-sets (*SET_PART ➔ PSID) ; nodes into node-sets (*SET_NODE ➔ NSID); segments into segment-sets (*SET_SEGMENT ➔ SGSID), etc., as necessary for interaction constraint definitions.

Modeling Procedure ←→ LS-DYNA Commands

- Define Initial (IC), Boundary (BC) & Load conditions (*INITIAL_, *BOUNDARY_, *LOAD_, etc.).
- Define interaction constraints or *CONTACT_ among Lagrangian parts.
- Define interaction constraints among the Master|ALE|Eulerian|Fluid and Slave|Lagrangian|Structure parts. This is done with the command *CONSTRAINED_LAGRANGE_IN_SOLID.
- Define ALE controls for the various set-IDs: PID|PSDI|SGSID|NSID … These controls can include: [1] multi-material mixing in an ALE element (*ALE_MULTI-MATERIAL_GROUP); [2] SID|mesh motion & expansion (*ALE_REFERENCE_SYSTEM_); and [3] SID|mesh smoothing (*ALE_SMOOTHING).
- Define user-input forcing functions or load curve definitions (LCID), as necessary (*DEFINE_CURVE) for the above commands.

Some Abbreviations

SID	= Set ID
NSID	= Node set ID
PID	= Part ID
PSID	= Part set ID
SGSID	= Segment set ID
SHSID	= Shell element set ID
SOSID	= Solid element set ID
STP	= Standard Temperature & Pressure ~(298K, 1 atm)

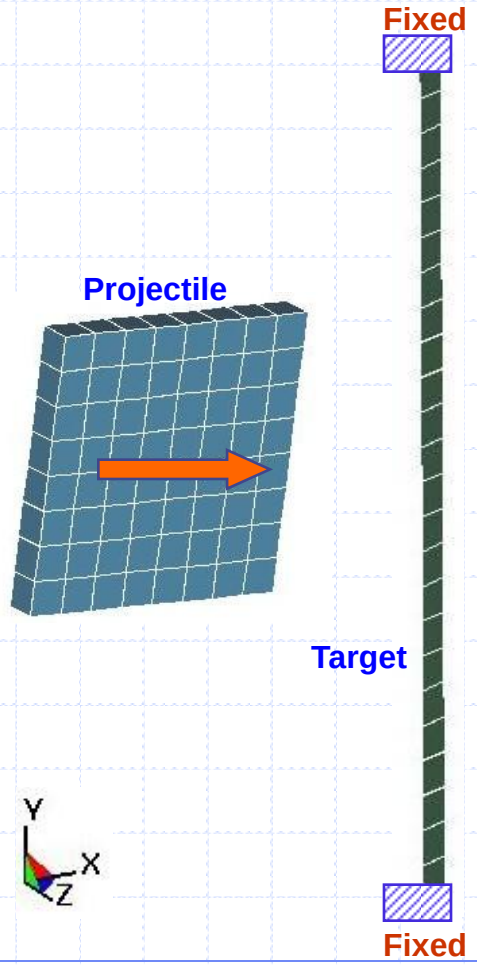
Fluid-Structure-Interaction Modeling Capabilities under Development

- Improving methods for minimizing "leakages" due to the (1) **FSI** and (2) advection methods.
- Material **temperatures** & **phase states** tracking throughout the whole calculation. This is necessary for modeling thermally sensitive materials and processes (injection molding, welding processes, etc.).
- Consistent energy calculations for thermally dependent processes.
- Adding material-interface contour plotting capability into **LSPOST**.
- Adding parameter contour plotting capability (superimposed on top of material plotting.)
- Non-Newtonian flow modeling capability.
- Improve dt calculation procedure to enhance robustness.
- Improve advection method to prevent negative advection volumes.

Lagrangian Model
Lagrangian Projectile hitting Lagrangian target

Pure Lagrangian Model: Projectile Hitting Target

Step 1: Understanding the physics of the problem



- The projectile is modeled as a deformable Lagrangian solid structure moving at constant velocity striking the target.
- The target is modeled as a deformable shell structure (also Lagrangian) with its 2 ends fixed.
- Since both structures are Lagrangian, the impact is governed by the *CONTACT_ algorithms in LS-DYNA.
- The meshes follow and deform with the materials.

Relating the physical set-up of the problem to LS-DYNA KEYWORD commands … ➔

Lagrangian Model: Projectile Hitting Target

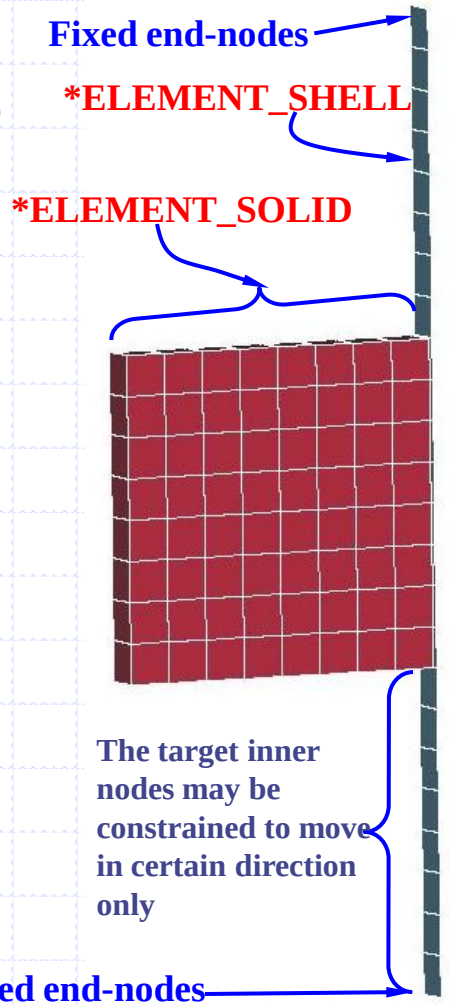
Step 2: Define the basic geometry of the problem

Basic geometry definitions & nodal boundary conditions are typically defined by commands such as:

***NODE** ~ NODE #,
nodal coordinate (x,y,z),
nodal motion constraints
(translational & rotational).

***ELEMENT_SOLID** ~ ELEMENT #,
the PART # this element belongs to, the node #'s of the nodes make up this element.

***ELEMENT_SHELL** ~ ELEMENT #,
the PART # this element belongs to, the node #'s of the nodes make up this element.



Lagrangian Model: Projectile Hitting Target

Step 3: Define characteristics of basic geometrical components

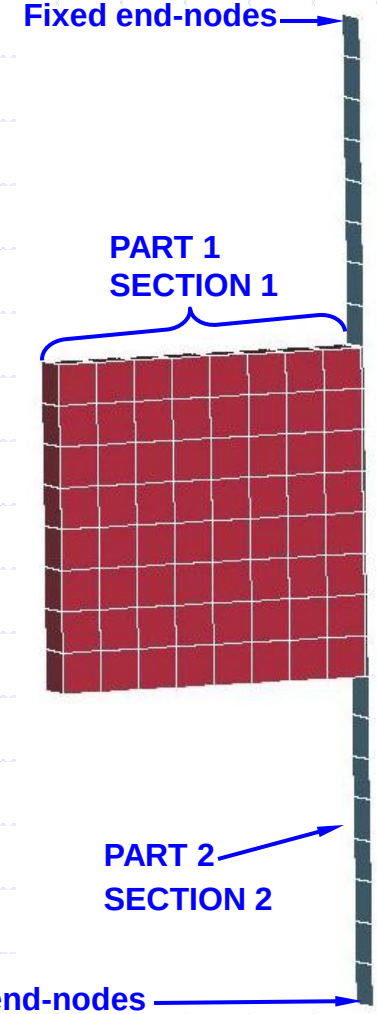
Organize the components in this model (projectile and target) into part(s) by the *PART command.

Each part is associated with an *element formulation* (types of solid or shell elements, etc.) defined by the commands *SECTION_SOLID or *SECTION_SHELL. Generally one can use:

*SECTION_SHELL Defines a SECTION ID, *shell element formulation* type, integration options, and other parameters required for shell element calculation.

*SECTION_SOLID Defines a SECTION ID, *solid element formulation* type, smoothing weight factors & times, and ALE advection vector.

*PART Defines a PART ID associated with a SECTION ID, its material and/or equation of state models, hourglass ID, gravitational initialization, mesh adaptivity, and thermal material ID.



Lagrangian Model: Projectile Hitting Target

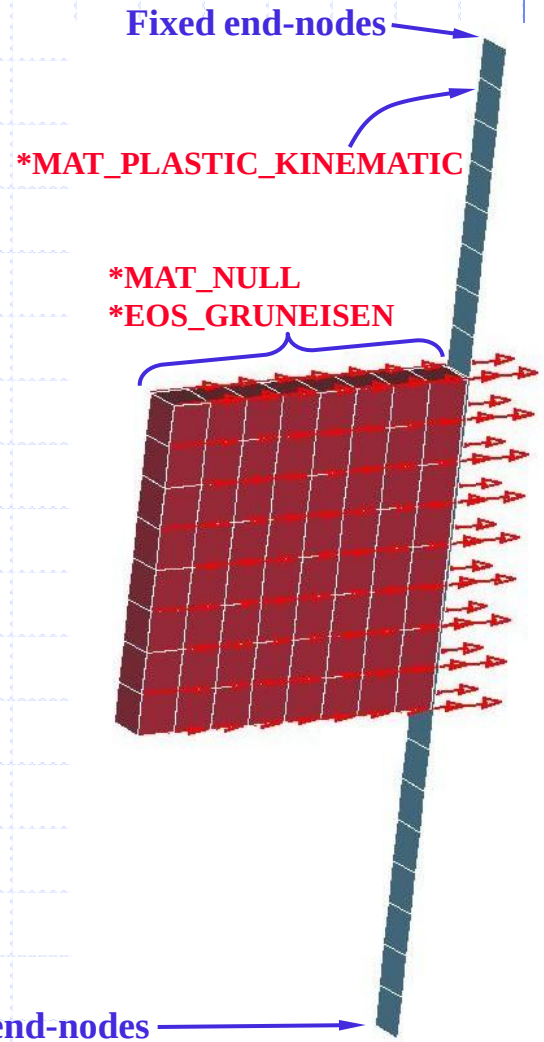
Step 4: Define material deformation behaviors

Each PART is assigned a Material or Constitutive Model (CM) and/or an Equation of State (EOS) model by the *MAT_ & *EOS_ commands such as:

*MAT_NULL ~ CM model for the projectile.

*EOS_GRUNEISEN ~ EOS model for the projectile (required by *MAT_NULL).

*MAT_PLASTIC_KINEMATIC ~ CM for the target (no EOS model required for this CM).



Lagrangian Model: Projectile Hitting Target

Step 5: Define the initial and|or boundary conditions.

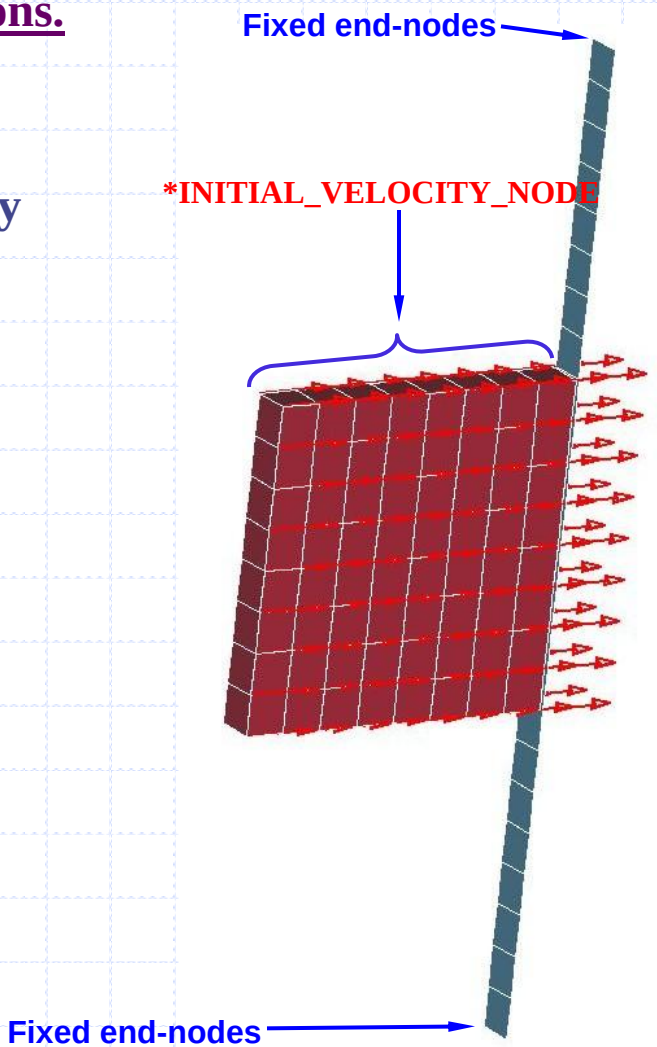
INITIAL CONDITION:

The initial velocity of any PART can be defined by the commands:

*INITIAL_VELOCITY_NODE which imposes a velocity on each NODE

or

*INITIAL_VELOCITY_GENERATION which imposes a velocity on each PART (created by the *PART command), PART_SET (created by *SET_PART), or NODE_SET (created by *SET_NODE).



Lagrangian Model: Projectile Hitting Target

Step 6: Define parts-interaction behavior

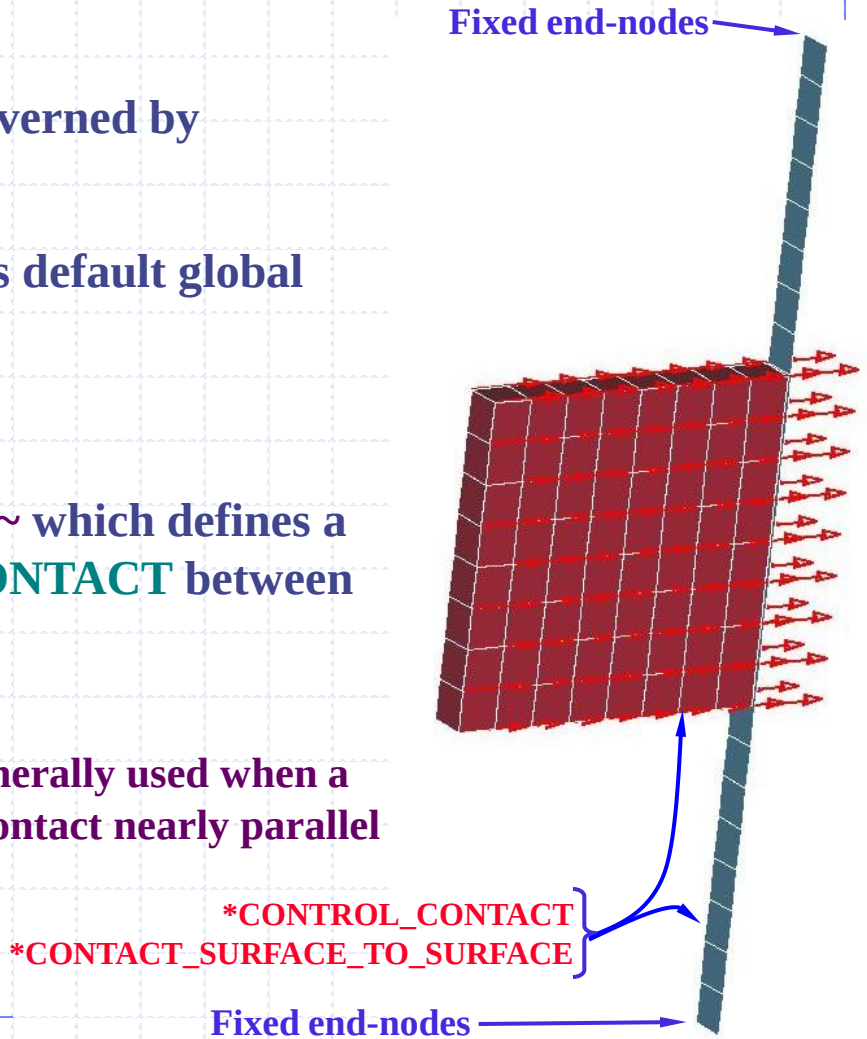
Impact/contact of Lagrangian bodies is governed by CONTACT algorithm via the commands:

*CONTROL_CONTACT ~ which controls default global CONTACT parameters;

and

*CONTACT_SURFACE_TO_SURFACE ~ which defines a method for detecting and resolving the CONTACT between the two Lagrangian parts.

NOTE:
*CONTACT_SURFACE_TO_SURFACE is generally used when a collection of surfaces (or segments) come into contact nearly parallel with another set of surfaces.



Lagrangian Model: Projectile Hitting Target

COMMAND SUMMARY: **INTERACTIONS**

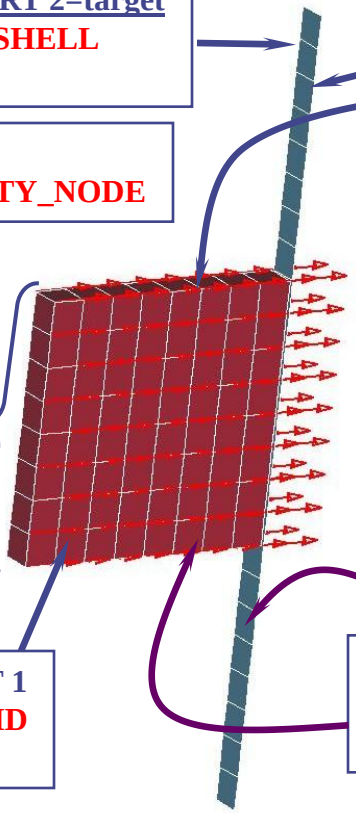
Projectile = Lagrangian SOLID ← CONTACT → Target = Lagrangian SHELL

[2] Define PART 2=target
*SECTION_SHELL
*PART

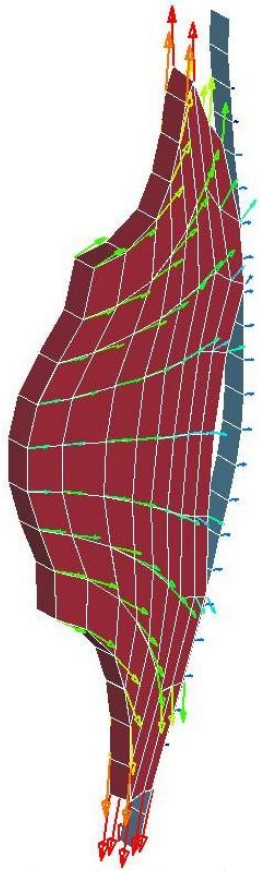
[5] = Define material properties:
*MAT_PLASTIC_KINEMATIC
*MAT_NULL
*EOS_GRUNEISEN

[3] Define IC's
*INITIAL_VELOCITY_NODE

[1] = Basic geometry definitions + BC's:
*NODE (anchor 2 ends of target)
*ELEMENT_SOLID
*ELEMENT_SHELL



Reasonable trend but
Severe element distortion



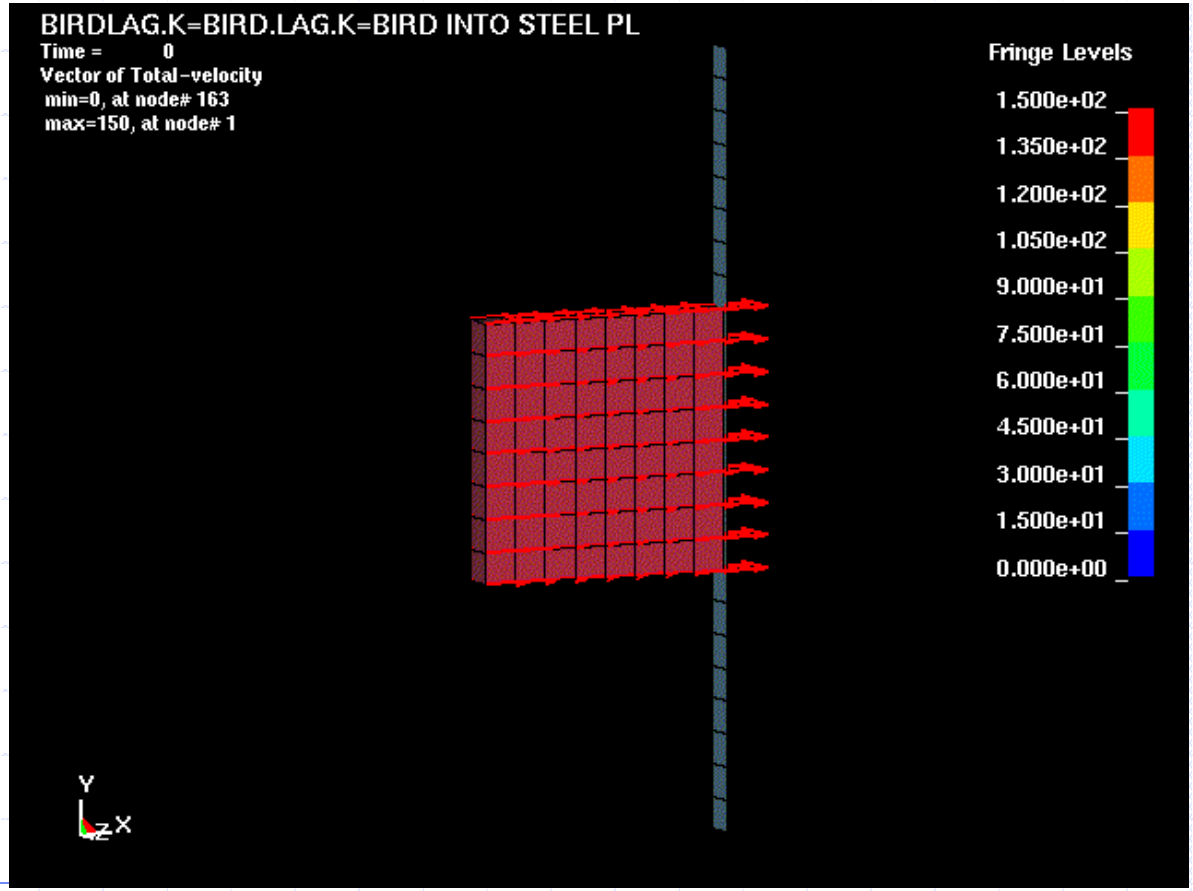
[2] = Define PART 1
*SECTION_SOLID
*PART

[4] = Structure Interactions - LAGRANGIAN:
*CONTROL_CONTACT
*CONTACT_SURFACE_TO_SURFACE

Lagrangian Model: Projectile Hitting Target

RESULTS (double-click on picture below):

OK for small deformation, but can lead to highly distorted elements and thus inaccurate result.



Lagrangian Model: Projectile Hitting Target

Details of the actual input commands …

Basic headings for LS-DYNA input deck (please consult the LS-DYNA user's manual):

Comment symbol = "$": LSDYNA ignores anything following a "$" at column 1.

Specifies that this input deck is of "*KEYWORD" format

```
$---------------------------------------------------------------------
*KEYWORD
*TITLE
Bird hitting steel plate (birdlag.k)
$---------------------------------------------------------------------
```

Specifies the title of this model (input filename, unit system,…)

Lagrangian Model: Projectile Hitting Target

Set up the CONTROL parameters for Execution & Computation options. (Only some typical CONTROL options are shown here. For more details, please consult the user's manual).

ENDTIM = Stopping time

Time or cycle number to stop calculation

Controls for energy Dissipation option

Controls for computing Shell response

```
$---------------------------------------------------------------------
*CONTROL_TERMINATION
$    ENDTIM    ENDCYC     DTMIN    ENDENG    ENDMAS
 2.5000E-3         0 0.0000000         0 0.0000000
$---------------------------------------------------------------------
*CONTROL_ENERGY
$      HGEN      RWEN     SLNTEN      RYLEN
         1         2          1          1
$---------------------------------------------------------------------
*CONTROL_SHELL
$    WRPANG     ITRIST      IRNXX     ISTUPD     THEORY        BWC      MITER      PROJ
 0.5000000          0          0          0          0          0          0
$---------------------------------------------------------------------
```

Energy control options:
HGEN ~ Hourglass
RWEN ~ Rigid-Wall
SLNTEN ~ Sliding-Interface
RYLEN ~ Rayleigh

Lagrangian Model: Projectile Hitting Target

Set up the CONTROL parameters for Output Options (please consult the user's manual for more details):

Specifies the time increment, DT, for the *complete* binary output

```
$---------------------------------------------------------------------
*DATABASE_BINARY_D3PLOT
$        DT       LCDT
 5.0000E-5          0
*DATABASE_BINARY_D3THDT
$        DT       LCDT
 1.0000001
$---------------------------------------------------------------------
```

Specifies dt for the *history* output

Lagrangian Model: Projectile Hitting Target

Define Basic Geometry, Nodes + Boundary Conditions and Elements :

Nodal ID

Nodal Coordinate

Translational Constraint

Rotational Constraint

```
$---------------------------------------------------------------------------
*NODE
$     NID               X               Y               Z    TRANSC     ROTC
       1 1.000000015E-01 1.000000015E-01 0.000000000E+00         3         0
       2 1.125000045E-01 1.000000015E-01 0.000000000E+00         3         0
...
     211 2.010000050E-01 2.875000238E-01 1.250000019E-02         3         4
     212 2.010000050E-01 3.000000119E-01 1.250000019E-02         7         7
*ELEMENT_SOLID
$     EID     PID      N1      N2      N3      N4      N5      N6      N7      N8
       1       1       1       2      11      10      82      83      92      91
       2       1       2       3      12      11      83      84      93      92
...
      63       1      70      71      80      79     151     152     161     160
      64       1      71      72      81      80     152     153     162     161
*ELEMENT_SHELL
$     EID     PID      N1      N2      N3      N4
       1       2     163     188     189     164
       2       2     164     189     190     165
...
      23       2     185     210     211     186
      24       2     186     211     212     187
$---------------------------------------------------------------------------
```

PART ID = 1

Define for each *solid* ELEMENT its ID, its associated PART, and its 8 node numbers (N1-N8).

Define for each *shell* ELEMENT its ID, its associated PART, and its 4 node numbers (N1-N4).

PART ID = 2

ELEMENT ID #'s

Lagrangian Model: Projectile Hitting Target

Define SECTIONs, PARTs, PART_LISTs, NODE_LISTs, etc.:

First, define SECTION properties (using *SECTION_SOLID or *SECTION_SHELL).
- Element formulation
- Smoothing options

(and in case of a SHELL,
- Shear factor
- Shell thicknesses

etc.)

```
$---------------------------------------------------------------------
*SECTION_SOLID
$     SECID    ELFORM       AET
          1         1
$      AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
$
*SECTION_SHELL
$     SECID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP     SETYP
          2         0 0.0000000 3.0000000 0.0000000 0.0000000         0
$       TH1       TH2       TH3       TH4      NLOC
 0.0100000 0.0100000 0.0100000 0.0100000 0.0000000
$
*PART
BIRD = SOLID ELM, ELFORM=1=CONST-STRESS-ELM
$       PID       SID       MID     EOSID      HGID      GRAV    ADPOPT    THERMID
          1         1         1         1         1         0         0          0
*PART
BLADE = SHELL ELM, ELFORM=2=DEFLT=BELYT-TSAY
          2         2         2         0         0         0         0          0
$---------------------------------------------------------------------
```

Second, define each PART with *PART:
- The associated *SECTION ID
- CM & EOS models
- Hourglass options, etc.

Lastly, to treat multiple PARTs or SEGMENTs as an entity the GROUPING commands may be used:

*SET_PART groups one or more PARTs (or PID's) under one PART_SET ID (or PSID);

*SET_SEGMENT groups one or more SEGMENTs (or surfaces, SGID's) under one SEGMENT_SET ID (or SGSID);

These PID, PSID, or SGSID, etc. can be used to define their interactions later.

Lagrangian Model: Projectile Hitting Target

Define Initial Conditions and Boundary Conditions:

In this case, all the nodes on the projectile body is given an initial translational velocity. Each node can be given a unique velocity.

```
*INITIAL_VELOCITY_NODE
$      NID        VX          VY          VZ        VRX        VRY        VRZ
         1 150.00000 0.0000000 0.0000000
         2 150.00000 0.0000000 0.0000000
$...
       211 0.0000000 0.0000000 0.0000000
       212 0.0000000 0.0000000 0.0000000
```

Nodal ID

Translational Velocity

Rotational Velocity (radian/unit-time)

NOTE:
we can also use another command to give an initial velocity to a PID or PSID or NSID:
***INITIAL_VELOCITY_GENERATION**

Lagrangian Model: Projectile Hitting Target

Define a Lagrangian CONTACT option between any 2 contacting *ENTITIES*:

Lagrangian CONTACT card is required.
Without this card the 2 PARTS will not be able to "see" each other, and the projectile will just fly right through the target.

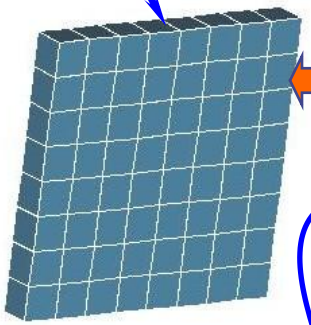
Lagrangian Shell Element as target

MSID = Master entity ID
(Projectile, Solid, Lagr.)

SSTYPE = Slave Set types:
0=SGSID 1=SHSID
2=PSID 3=PID
4=NSID 5=single-surf
6=exempted PSID

MSTYPE = Master Set types:
0=SGSID 1=SHSID
2=PSID 3=PID

Lagrangian Solid Element as projectile



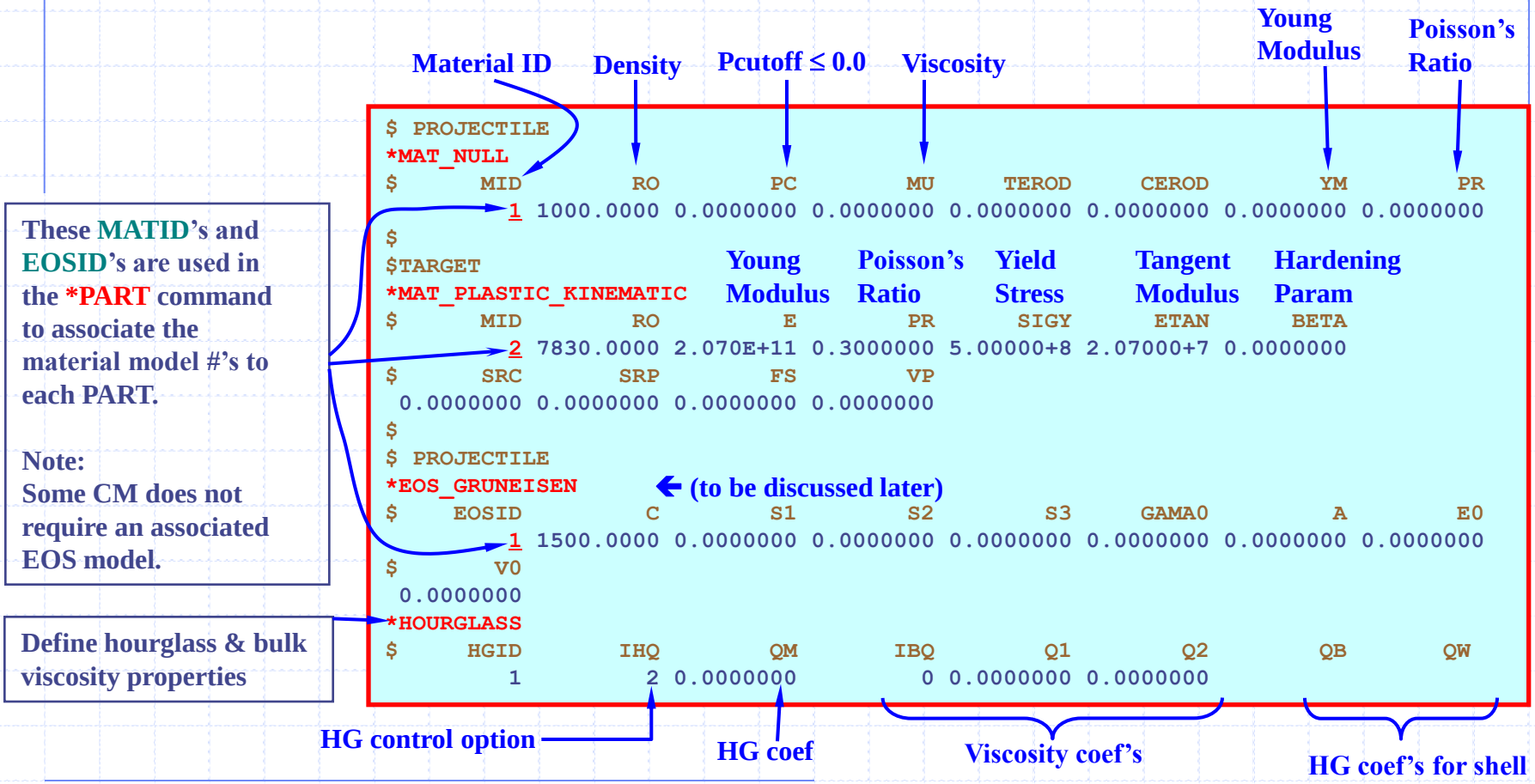
```
$ CONTACT CARDS: FIRST 3 CARDS ARE MANDATORY. IF NOT USED, MUST LEAVE A BLANK LINE.
*CONTACT_SURFACE_TO_SURFACE
$      SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
         2         1         3         3         0         0         0         0
$        FS        FD        DC        VC       VDC    PENCHK        BT        DT
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000         0 0.0000000 0.0000000
$       SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF

$ The blank line above is required for card # 3 for this type of contact.
```

SSID = Slave entity ID
(Target, Shell, Lagr.)

SBOXID = include only SLAVE nodes & segments w/i specified box
MBOXID = include only MASTER segments w/i specified box

Lagrangian Model: Projectile Hitting Target

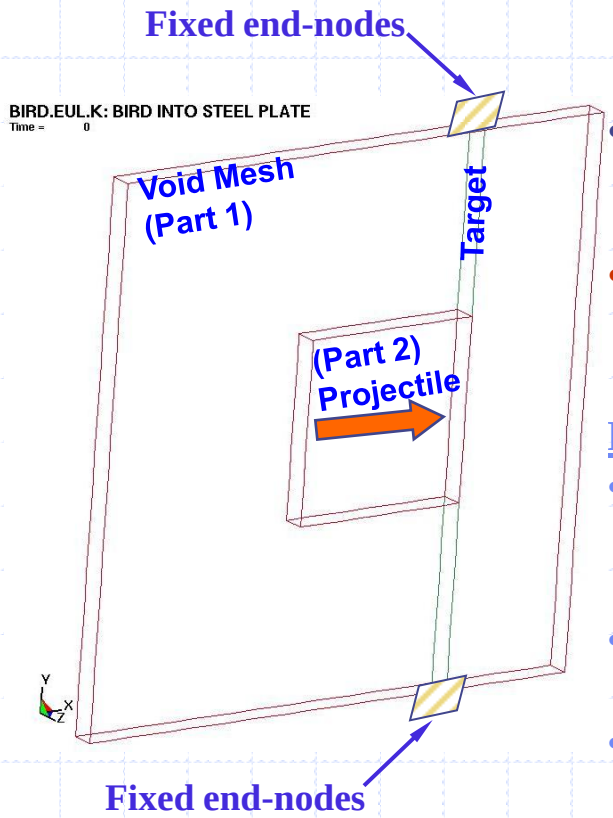
Define the MATERIAL deformation response properties
Consitutive & EOS models



Eulerian Model
ALE Projectile hitting Lagrangian target

Eulerian projectile hitting Lagrangian target

Another Impact Model



- The projectile is modeled as a deformable Eulerian solid structure moving at constant velocity striking the target.
- The target is modeled as deformable Lagrangian shell structure with its 2 ends fixed.
- Eulerian projectile hitting Lagrangian target is treated by COUPLING algorithms in LS-DYNA.

Features different from previous pure-Lagrange example:

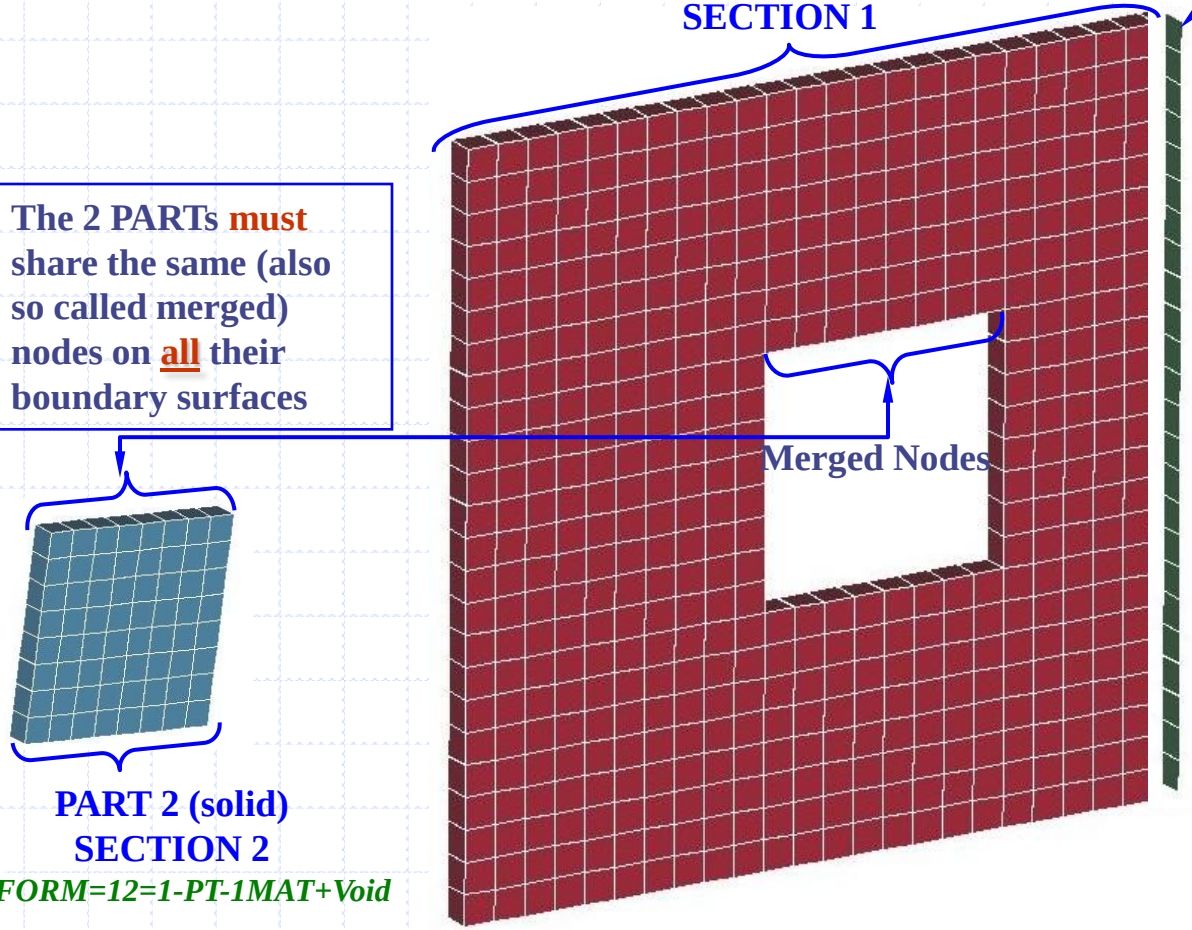
- The PARTs are constructed in a different way:
  PART 1=VOID & PART 2=Projectile: 1-MAT+VOID
  PART 3=TARGET=Shell Element Structure
- An Eulerian mesh containing the projectile and overlaps with the target Lagrangian mesh.
- COUPLING mechanism (not CONTACT) handles the interaction between the 2 impacting objects.

Eulerian projectile hitting Lagrangian target

PART CONSTRUCTION

ELFORM=12=1-PT-1MAT+Void
PART 1 (solid)
SECTION 1

ELFORM=0=Belytschko-Tsay shell
PART 3 (shell)
SECTION 3



The 2 PARTs must share the same (also so called merged) nodes on all their boundary surfaces

PART 2 (solid)
SECTION 2
ELFORM=12=1-PT-1MAT+Void

- Create PID 1, the overall computational domain.
- Delete the space occupied by PID 2 (projectile) from it, getting PID 1 mesh.
- Create PID 2.
- Merge these 2 parts so they share the same nodes on their common boundary surfaces.
- Create the target, PID 3.

- *Assign the same ELFORM, MATID & EOSID to both parts 1 & 2 (properties of the projectile).*
- *Turn PID 1 into void using the command* ***INITIAL_VOID_PART.***
- *Give PID 2 initial velocity.*

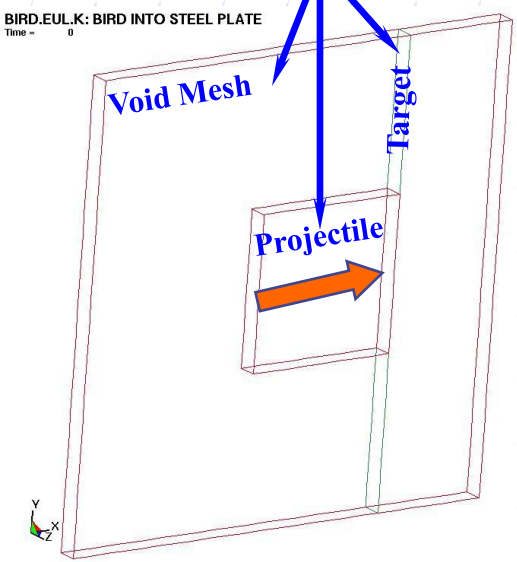
Eulerian projectile hitting Lagrangian target

GEOMETRY DEFINITIONS:

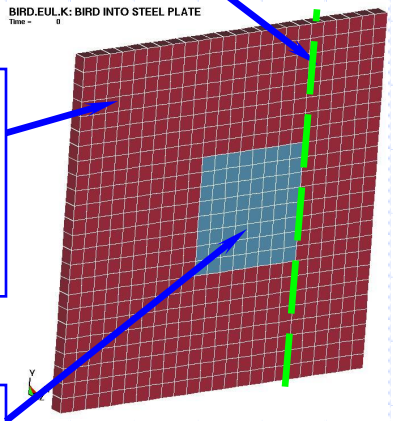
Basic geometry definitions & nodal boundary conditions are typically defined by commands such as:

Basic geometry definitions + BC's:
***NODE (anchor 2 ends of target)**
***ELEMENT_SOLID**
***ELEMENT_SHELL**

Target=PID 3=Lagrangian SHELL overlaps solid mesh (hidden)
***ELEMENT_SHELL**



Void=PID 1
Solid (ALE) element
ELFORM=12=1-PT-1-MAT+Void
***ELEMENT_SOLID**



Projectile=PID 2
Solid (ALE) element
ELFORM=12=1-PT-1-MAT+Void
***ELEMENT_SOLID**

Eulerian projectile hitting Lagrangian target

MODEL ORGANIZATION:

Define each PART in the model.

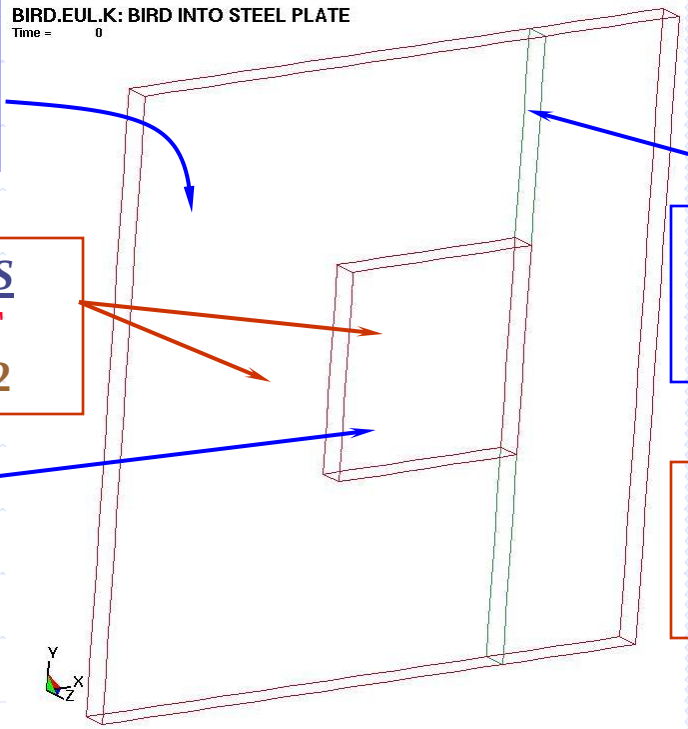
Grouping PARTs (PID) into PART_SET (PSID):

PSID 1:{PID 3} ← ***CONSTRAIN-COUPLING*** → ***PSID 2:{PID 1 & PID 2}***

PART 1=Void=ALE Solid
***SECTION_SOLID**
***PART**

GROUPING PARTS
***SET_PART_LIST**
PSID 2: Parts 1 & 2

PART 2=ALE Projectile
***SECTION_SOLID**
***PART**



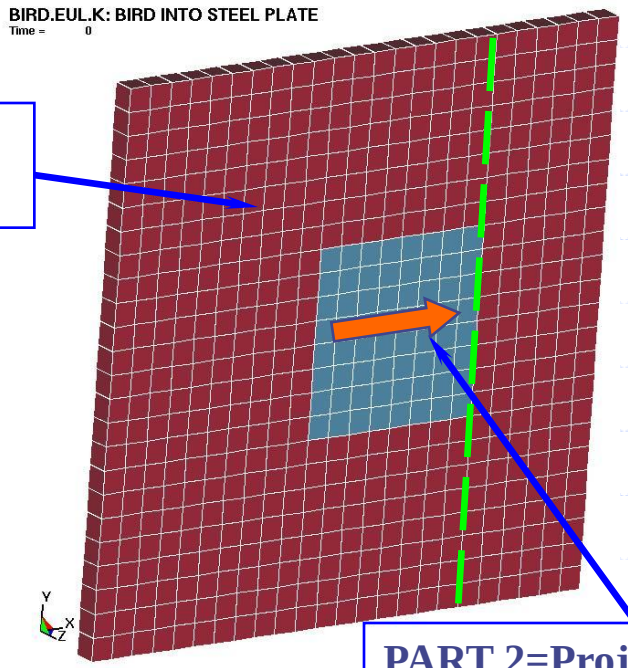
PART 3=Target
***SECTION_SHELL**
***PART**

GROUPING PARTS
***SET_PART_LIST**
PSID 1: {PID 3}

Eulerian projectile hitting Lagrangian target

INITIAL CONDITIONS:

- Initially in *PART, PID 1 has MATID=2 & EOSID=2, the properties of PID 2 (projectile). To turn PID 1 into VOID, use the *INITIAL_VOID_PART command.
- To give the PID 2 its initial velocity, use *INITIAL_VELOCITY_GENERATION



PART 1 = Void
*INITIAL_VOID

PART 2=Projectile
*INITIAL_VELOCITY_GENERATION

Eulerian projectile hitting Lagrangian target

DEFORMATION BEHAVIOR: (Similar to previous example)

*MAT_NULL ~ CM model for the projectile.
*EOS_GRUNEISEN ~ EOS model for the projectile (required by *MAT_NULL).
*MAT_PLASTIC_KINEMATIC ~ CM for the target (no EOS model required).

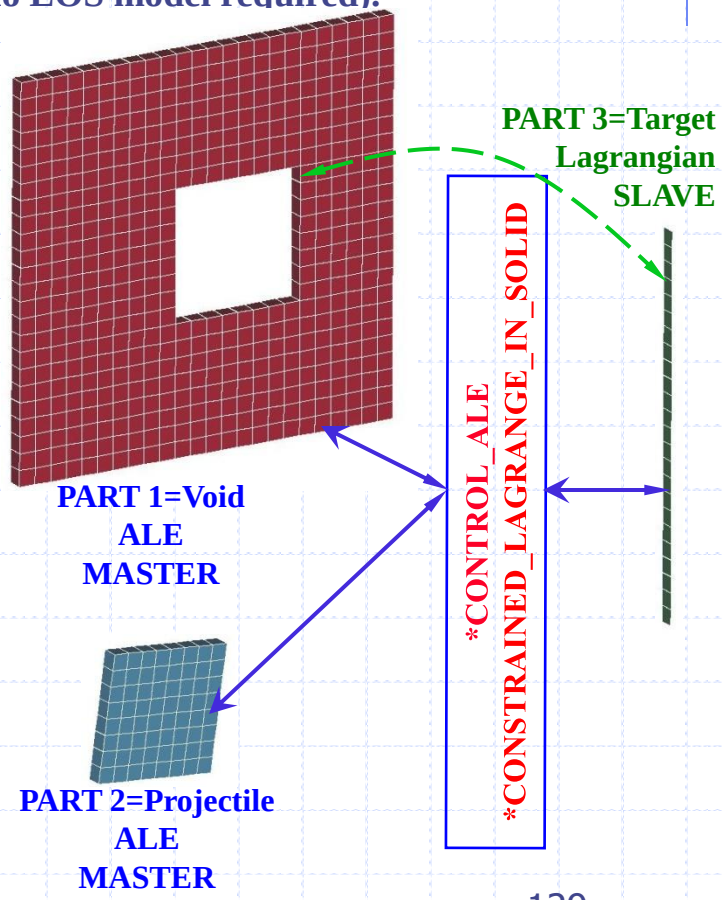
PARTS-INTERACTION BEHAVIOR:

Interaction of ALE VS. Lagrangian bodies is governed by COUPLING algorithm via the commands:

*CONTROL_ALE ~ controls default ALE & Eulerian parameters for advection and smoothing.

&

*CONSTRAINED_LAGRANGE_IN_SOLID ~ Couple a Lagrangian mesh (slave) of shells|solid|beams to the material points of an Eulerian (master) flow.



Eulerian projectile hitting Lagrangian target

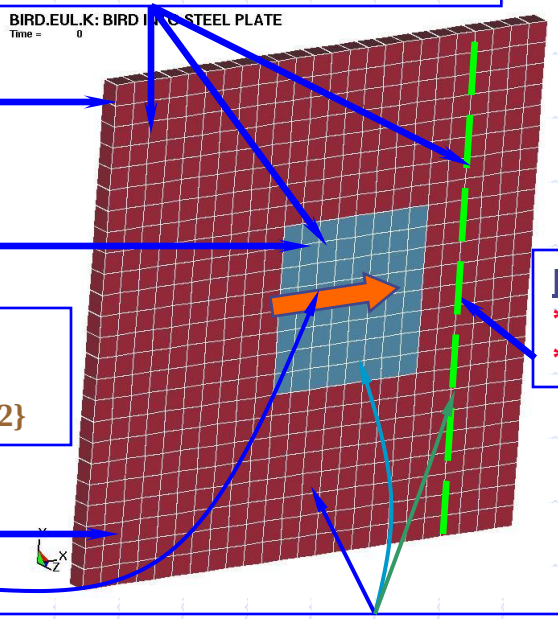
COMMAND SUMMARY: INTERACTIONS

Projectile = ALE SOLIDS ← COUPLING → Target = Lagrangian SHELL

[1] Basic geometry definitions + BC's:
*NODE (anchor 2 ends of target)
*ELEMENT_SOLID
*ELEMENT_SHELL

[6] Define material properties:
*MAT_PLASTIC_KINEMATIC
*MAT_NULL
*EOS_GRUNEISEN

[2] Define PART 1=Void=ALE Solid
*SECTION_SOLID
*PART (Master)



[2] Define PART 2 =Projectile=ALE Solid
*SECTION_SOLID
*PART (Master)

[2] Define PART 3=target=Lagr. Shell
*SECTION_SHELL
*PART (Slave)

[3] Organize model:
*SET_PART_LIST defines PSID 1: {PART 3}
*SET_PART_LIST defines PSID 2: {PARTs 1 & 2}

[4] BC's & constraints:
*INITIAL_VOID
*INITIAL_VELOCITY_GENERATION

PSID 2: {PARTs 1 & 2} Master

[5] Fluid-Structure Interactions ALE VS. LAGRANGIAN
*CONTROL_ALE
*CONSTRAINT_LAGRANGE_IN_SOLID

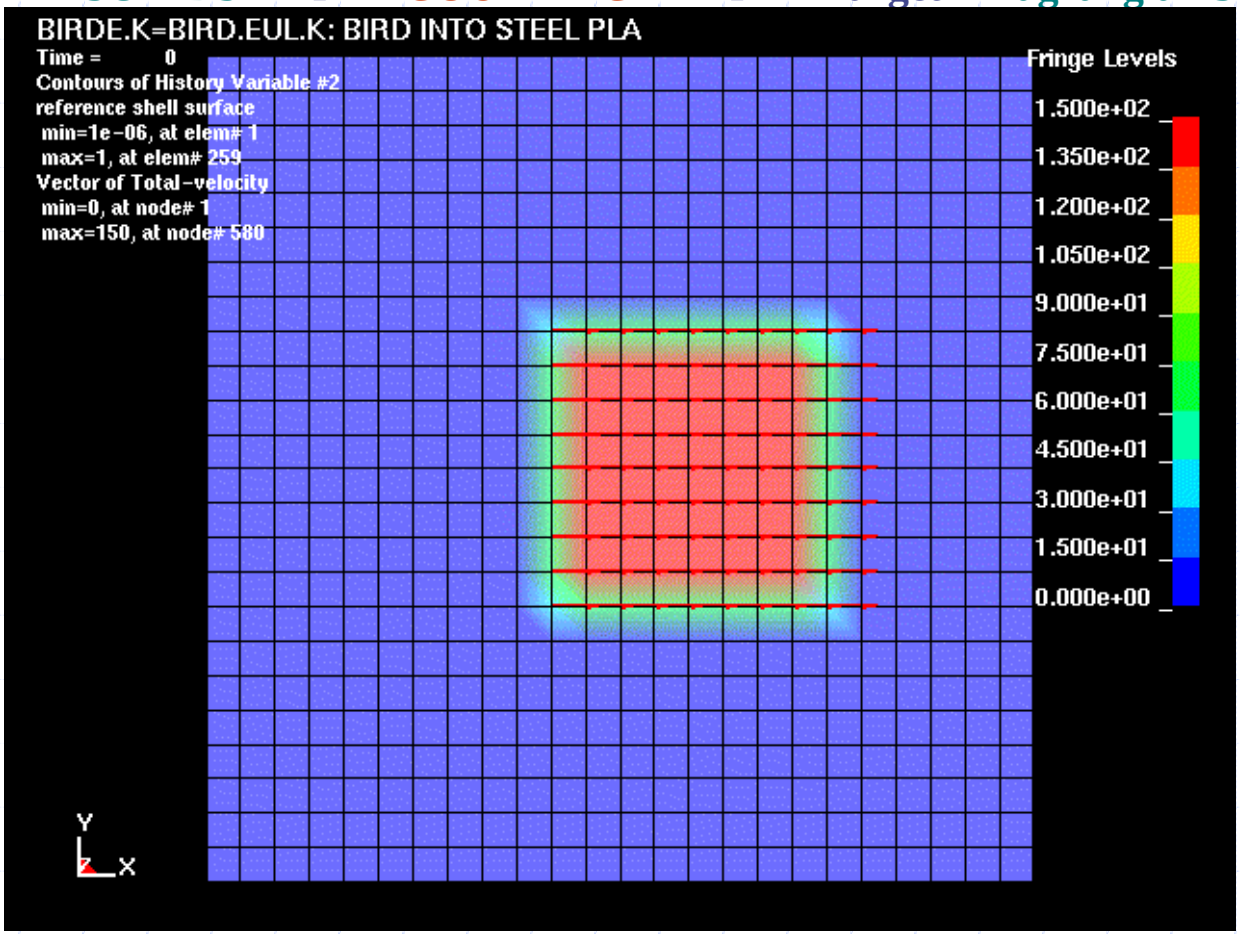
PSID 1: {PART 3} Slave

Eulerian projectile hitting Lagrangian target

RESULTS:

INTERACTIONS

Projectile = ALE SOLIDS ← COUPLING → Target = Lagrangian SHELL



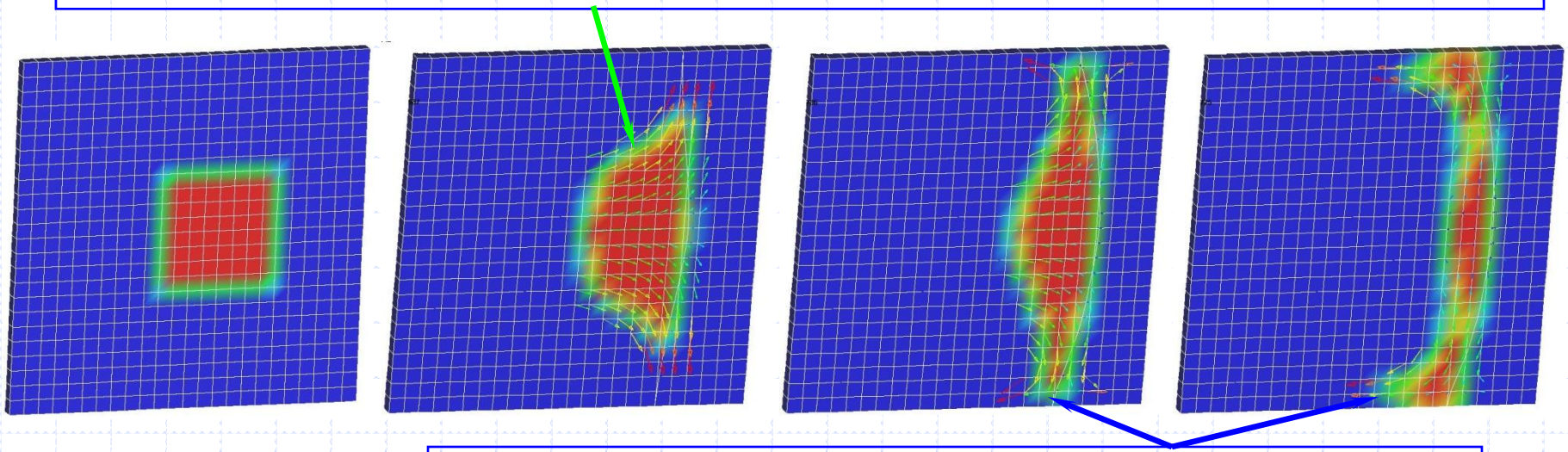
Double-click on the picture ➔

Eulerian projectile hitting Lagrangian target

NOTES ON RESULTS: **INTERACTIONS**

Projectile = ALE SOLIDS ← COUPLING → Target = Lagrangian SHELL

Eulerian projectile ("flowing" through the Eulerian mesh):
Its shape is tracked by the volume fraction (vf) of the material in an element.
The edge contour variation reflects the vf magnitude & depends on the mesh resolution.
To get good interface ➔ use higher mesh resolution.



NO OUTFLOW AT BOUNDARIES

The motion constraints set with the *NODE command does not allow any outflow out of the Eulerian void mesh in this case.

Eulerian projectile hitting Lagrangian target

BASIC GEOMETRY: Nodes + Boundary Conditions and Elements …

Nodal ID

Nodal coordinate

Translational Constraint

Rotational Constraint

```
*NODE
$     NID               X               Y               Z   TRANSC    ROTC
       1 0.000000000E+00 0.000000000E+00 0.000000000E+00        7       0
...
    1300 2.250000089E-01 3.000000119E-01 1.250000019E-02        7       7
$
*ELEMENT_SOLID
$     EID     PID      N1      N2      N3      N4      N5      N6      N7      N8
       1       1       1       2      10       9      65      66      74      73
...
     259       2     580     581     597     596     588     589     661     660
...
     576       1    1193    1194    1201    1200    1242    1243    1250    1249
$
*ELEMENT_SHELL
$     EID     PID      N1      N2      N3      N4
       1       3    1251    1276    1277    1252
...
      24       3    1274    1299    1300    1275
```

SOLID: 8 node numbers

SHELL: 4 node numbers

PART 1=Void

PART 2=Projectile

Each SOLID ELM:
Its associated PART,
Its 8 nodes.

Each SHELL ELM:
Its associated PART,
Its 4 nodes.

PART 3=Target

Eulerian projectile hitting Lagrangian target

MODEL ORGANIZATION: SECTIONs, PARTs, PART_LISTs, etc.

For solid: ELFORM=12=1-PT-INTEG-1MAT+VOID
For shell: ELFORM=0=Default=2=Belytschko-Tsay

***SECTION_SOLID_ALE or *SECTION_SHELL** define: Element formulation, smoothing options (and in case of SHELL, shear factor, shell thicknesses, etc.)

***PART** defines:
Its ***SECTION** ID, MATID, EOSID, Hour-Glass-ID, Gravity, Mesh Adapivity, Thermal MATID.

***SET_PART_LIST** defines:
PSID's, each contains 1 or more PID's.
PSID's can be used

```
*SECTION_SOLID_ALE                                                    <-- Void
$       SID     ELFORM       AET
         1         12
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
*SECTION_SOLID_ALE                                                    <-- Projectile
         2         12
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
*SECTION_SHELL                                                        <-- Target
$     SECID    ELFORM      SHRF       NIP      PROPT    QR/IRID      ICOMP      SETYP
         3         0 0.0000000 3.0000000 0.0000000 0.0000000
$       TH1       TH2       TH3       TH4      NLOC
 0.0100000 0.0100000 0.0100000 0.0100000 0.0000000
*PART
VOID = SOLID ELM, ALE MULMAT
$       PID       SID       MID     EOSID      HGID      GRAV    ADPOPT    THERMID
         1         1         2         2         0         0         0          0
*PART
PROJECTILE = SOLID ELM, ALE MULMAT, MATSER
         2         2         2         2         0         0         0          0
*PART
TARGET = SHELL ELM, LAGR MAT, SLAVE
         3         3         3         0         0         0         0          0
*SET_PART_LIST
$       SID       DA1       DA2       DA3       DA4
         1                                                            <-- PSID 1 contains PID 3
         3
*SET_PART_LIST
         2                                                            <-- PSID 2 contains PID 1&2
         1         2
```

Eulerian projectile hitting Lagrangian target

INITIAL & BOUNDARY CONDITIONS:

Initialize-by-PART:
*INITIAL_VELOCITY_GENERATION
gives PID 2 (projectile) its initial velocity.

Initialize-by-PART:
*INITIAL_VOID_PART makes
PID 1 "VOID" initially.

```
$ MAKE PART 1 VOID
*INITIAL_VOID_PART
$ PSID/PID
         1
$-------------------------------------------------------------------
$ GIVE PROJECTILE, PART 2, ITS INITIAL VEL
*INITIAL_VELOCITY_GENERATION
$PSID|PID|NSID SETYP      OMEGA       VX          VY          VZ
         2         2          0    150.           0           0
$       XC          YC         ZC       NX          NY          NZ      PHASE

$ Keep *INITIAL_VELOCITY_GENERATION card #2 BLANK
```

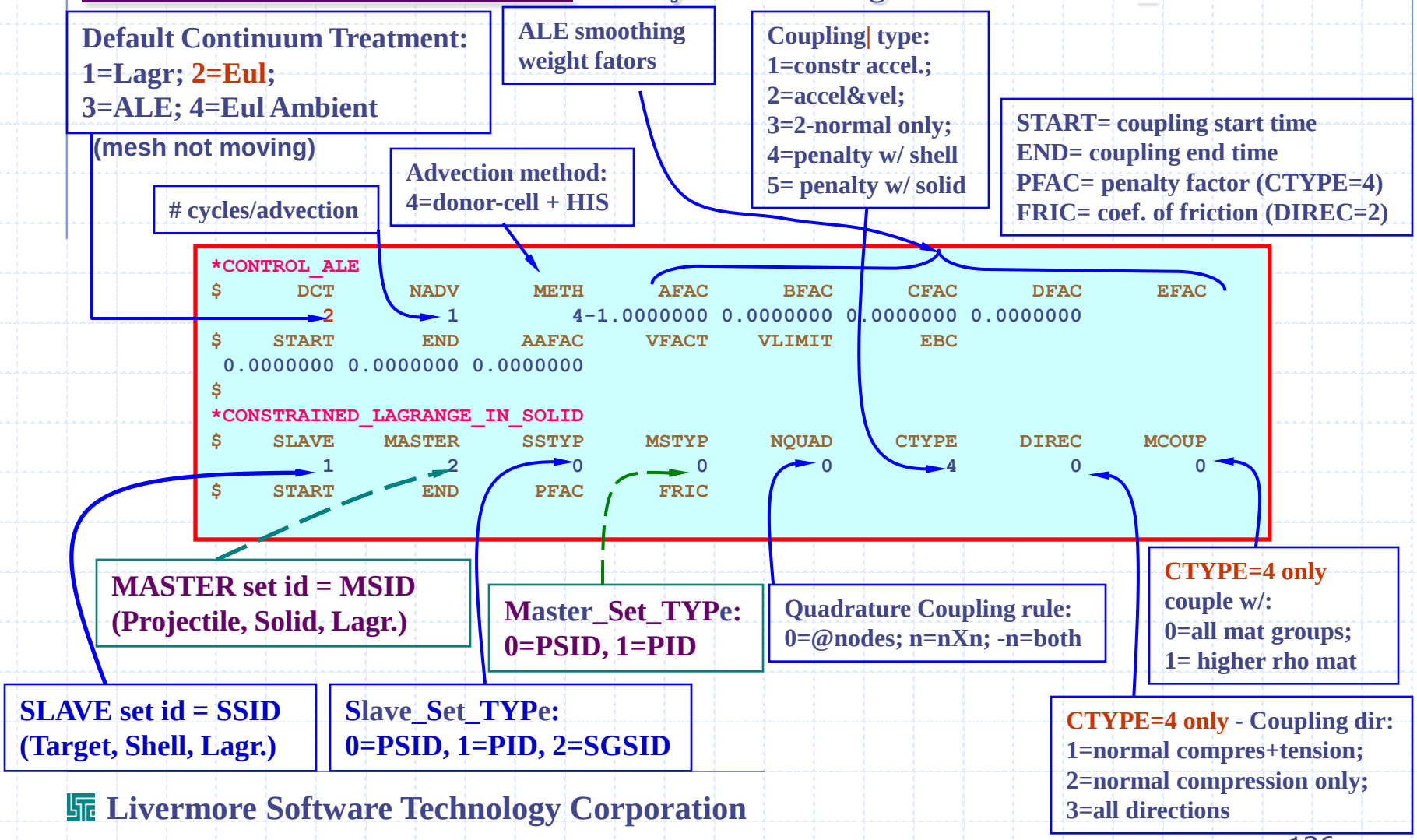
Velocity

1=PSID
2=PID
3=NSID

Angular Velocity about rotation axis is defined on the 2nd card:
{Xc,Yc,Zc} = a coordinate on the rotational axis
{Nx,Ny,Nz} = directional cosines of the rotational axis
PHASE = 0: apply velocity immediately
= 1: apply velocity after dynamic relaxation

Eulerian projectile hitting Lagrangian target

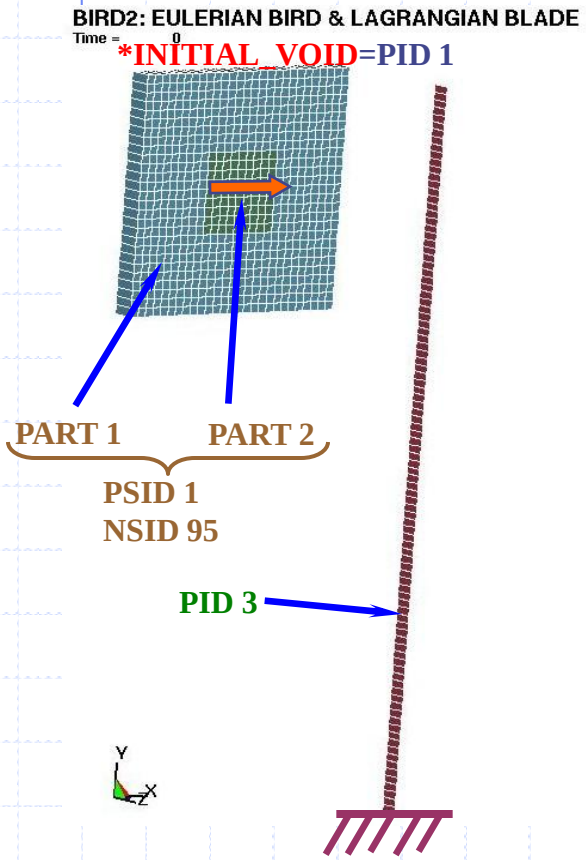
INTERACTION COUPLING: for any 2 interacting PARTs or PART_LISTs :



ALE Model (Target has 1 free-end, moving mesh)
ALE Projectile hitting Lagrangian target

ALE projectile hitting Lagrangian target (moving mesh)

OVERVIEW: The projectile is modeled as an ALE object surrounded by the ALE void space. The ALE mesh (containing the projectile & void but does not cover the whole computational domain) is to move with the projectile's average velocity.



INTERACTION

- Group the 3D solid elements of PART 1 (void) & PART 2 (projectile) into a PSID 1 (or PART_SET 1) .
- Let PSID 1 (an ALE system) interact with PART 3 (or PID 3, a Lagrangian structure).

INITIAL VELOCITY

- Group a set of NODES including both PIDs 1 & 2 into a NODE_SET (NSID 95).
- In the initial condition, give this NODE_SET an initial velocity.

(Note: Only nodes with nonzero mass can move with the prescribed velocity. So for the void nodes to move we need additional constraint ➔ below)

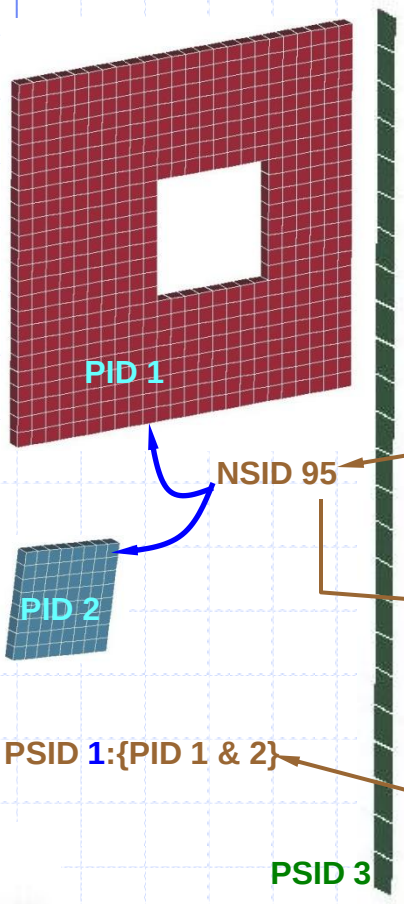
ALE MESH MOTION

- Define PSID 1 as an *ALE_REFERENCE_SYSTEM_GROUP for to allow for the ALE mesh to move with the projectile.

ALE projectile hitting Lagrangian target (moving mesh)

INITIAL VELOCITY & ALE MESH MOTION:

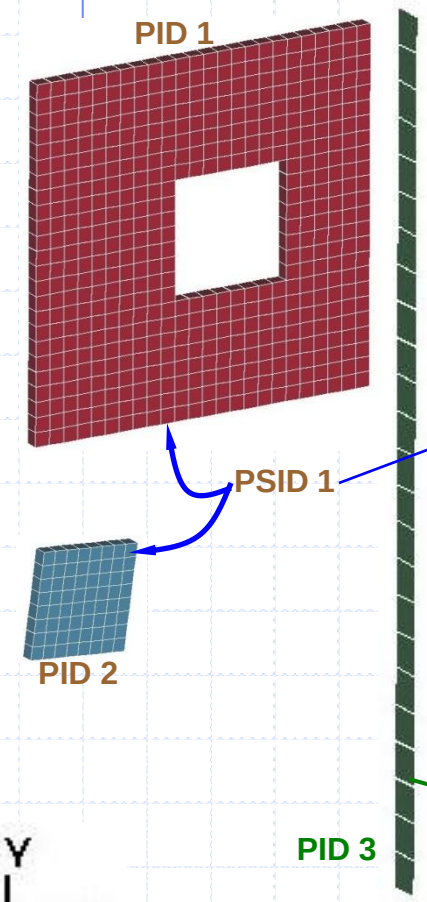
- A NODE_SET containing all nodes in the ALE mesh is defined, NSID 95.
- NSID 95 is given an initial velocity using the command *INITIAL_VELOCITY.
- Since only nodes with nonzero mass can move with this prescribed velocity, to specify the ALE mesh movement, we can use the command *ALE_REFERENCE_SYSTEM_GROUP.




```
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
$ For ALE mesh motion, create NSID 95: {all nodes in 3D solid elements}
*SET_NODE_LIST_GENERATE
95
1569,3136
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
*INITIAL_VELOCITY                    ← This give initial vel. To all NODES in set.
$     NSID    NSIDEX     BOXID       ← If node has no mass → ignore this vel.
        95
$       VX        VY        VZ       VXR       VYR       VZR
       5.0      -1.0       0.0       0.0       0.0      20.0
$ ASSIGN REF SYS TYPE TO SETID=1; SETTYPE=0=PSID; PRTYPE=4=REFSYSTYP=mesh_ave_v
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
*ALE_REFERENCE_SYSTEM_GROUP
$       SID     STYPE     PRTYP       PRID    BCTRAN     BCEXP     BCROT    ICOORD
          1         0         4          0         3         3         4         0
$        XC        YC        ZC    EXPLIM
         0         0         0       1.3          (More details on this later …)
```

ALE projectile hitting Lagrangian target (moving mesh)

INTERACTION COUPLING: The fluid-structure-coupling is similar to the last example.



FSC:

The ALE-Lagrangian interaction is defined similarly as before …
i.e., via the commands:

*CONTROL_ALE (usually used with ELFORM: 5,6,7,11,12)

*CONSTRAINED _LAGRANGE_IN_SOLID

Advection method:
4=Donor Cell + HIS

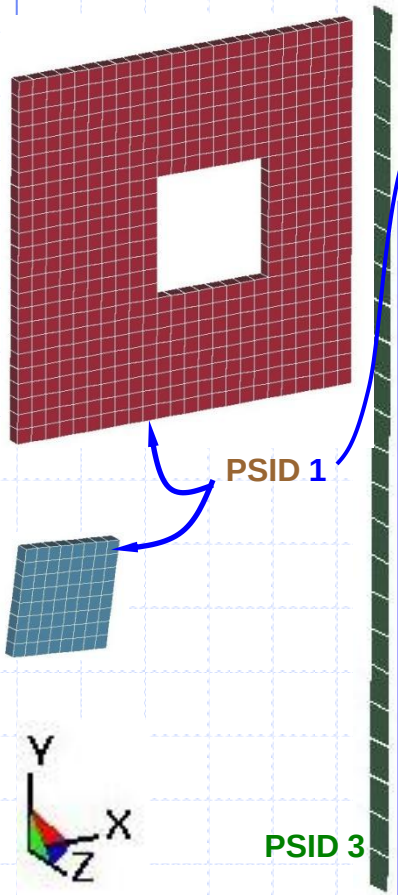
Turn off smoothing

```
$ For FSC: Create PSID 1: {PID 1(void) & 2(projectile)}
*SET_PART_LIST
1
1,2
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
$ DCT=2=EUL; METH=4=donor+HIS
*CONTROL_ALE
$       DCT      NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
          2         1         4-1.0000000 0.0000000 0.0000000 0.0000000
$     START       END     AAFAC     VFACT    VLIMIT       EBC
 0.0000000 0.0000000 0.0000000
$ SSTYP=1=PID;MSTYP=0=PSID;NQUAD=0=@NODES;CTYPE=4=PENALTY;DIREC=2=COMPRES,1=HI RHO
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
         3         1         1         0         0         4         2         1
$     START       END      PFAC      FRIC
       0.0       0.0       0.1
```

PID PSID Penalty coupling Normal compression

ALE projectile hitting Lagrangian target (moving mesh)

Some more details of the command *ALE_REFERENCE_SYSTEM_GROUP …




```
$ ASSIGN REF SYS TYPE TO SETID=1; SETTYPE=0=PSID; PRTYPE=4=REFSYSTYP=mesh_ave_v
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
*ALE_REFERENCE_SYSTEM_GROUP
$       SID     STYPE     PRTYP       PRID    BCTRAN     BCEXP     BCROT    ICOORD
          1         0         4          0         3         3         4         0
$        XC        YC        ZC    EXPLIM
         0         0         0       1.3
```

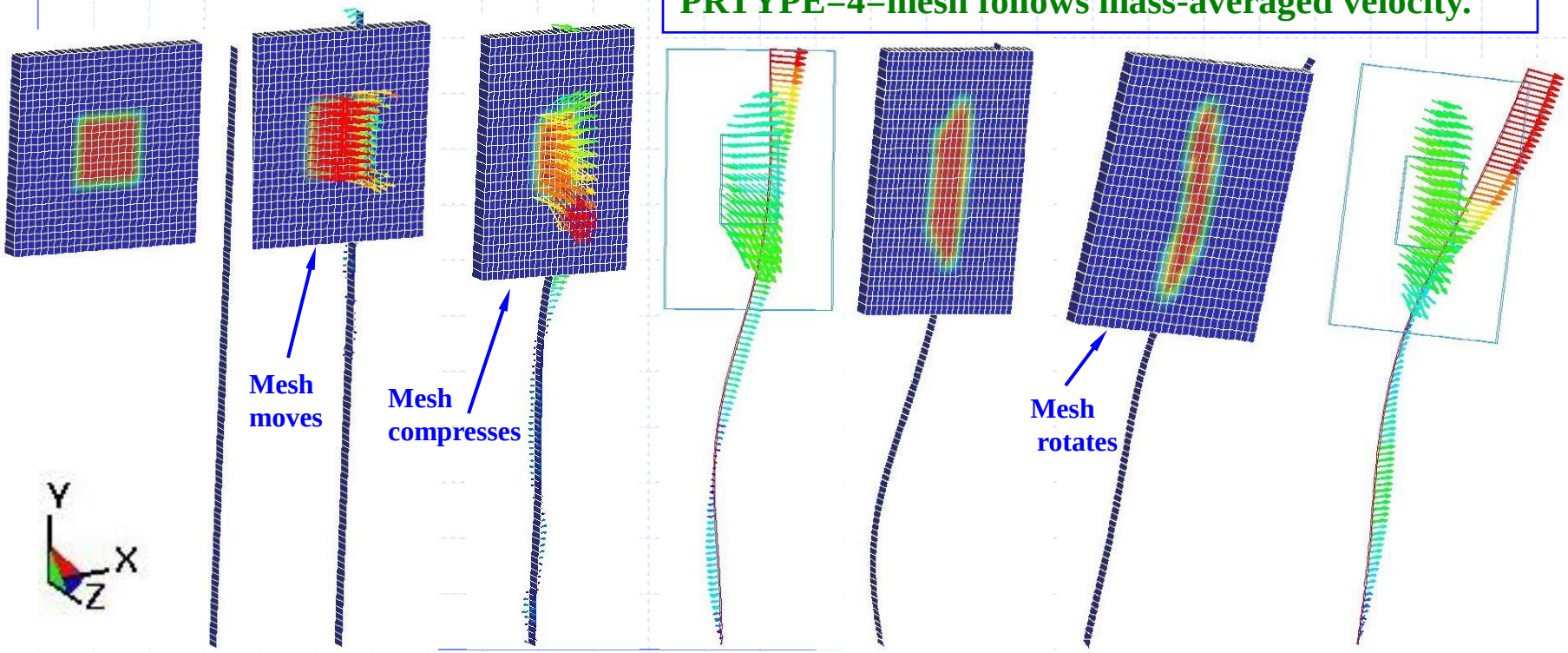
```
$*ALE_REFERENCE_SYSTEM_GROUP: assign ref sys types (LAGR|EUL|ALE) to a *_SET
$ CARD 1 ------------------------------------------------------------------------
$ SID    = set id
$ STYPE = SET TYPE:  0=part-set;   1=part;   2=node-set;   3=segment-set
$ PRTYPE= REFERENCE SYSTEM TYPE:
$         0=Eulerian;    1=Lagrangian;    2=normal ALE smoothing;
$         3=prescribed motion using load curve: *ALE_REFERENCE_SYSTEM_CURVE;
$         4=auto mesh motion follow mass-averaged velocity of the ALE mesh;
$         5=auto mesh motion following 3-NODES: *ALE_REFERENCE_SYSTEM_NODE;
$         6=switch in time |.| ref sys types: *ALE_REFERENCE_SYSTEM_SWITCH;
$         7=auto mesh expansion to enclose: *ALE_REFERENCE_SYSTEM_NODE;
$ PRID  = ID of switch list (node group or curve group (for PRTYPE:3,4,5,6,7)
$ BCTRAN= TRANSLATIONAL constraints (PRTYPE 3,4,5 & 7)
$         0=none;      1=fix X;      2=fix Y;      3=fix Z;
$                      4=fix XY;     5=fix YZ;     6=fix ZX;      7=fix XYZ
$ BCEXP = mesh EXPANSION  constraints (PRTYPE 3,4,5 & 7):similar to BCTRAN
$ BCROT = mesh ROTATIONAL constraints (PRTYPE 3,4,5 & 7): :similar to BCTRAN
$ ICOORD= center of mesh expansion or rotation: 0=CG; 1=@ given xc-yc-zc
$ CARD 2 ------------------------------------------------------------------------
$ XC,YC,ZC = center of mesh expansion or rotation
$ EXPLIM   = limiting ratio for mesh expansion: max expan=explim; min=1/explim
```

ALE projectile hitting Lagrangian target (moving mesh)

RESULT SNAP SHOTS:

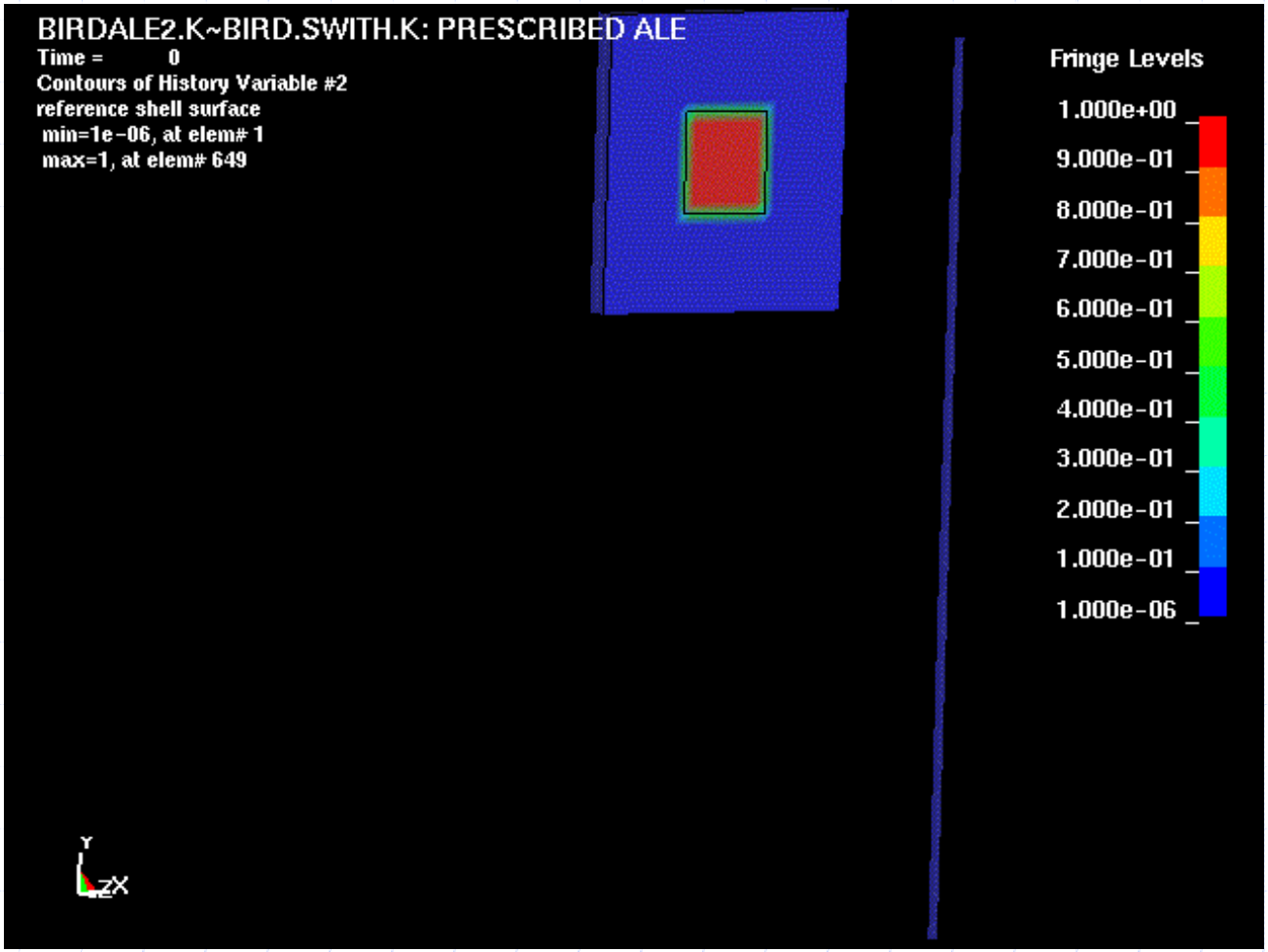
- The mesh moves and compresses in the X-direction proportionally to the X-deformation of the ALE material (BCTRAN=3=fixed-in-z, thickness direction; BCEXP=3=fixed-in-z)
- It also rotates as this non-symmetrical impact results in the rotation of the projectile (BCROT=4=allow mesh to rotate-about-z-only).

PRTYPE=4=mesh follows mass-averaged velocity.



ALE projectile hitting Lagrangian target (moving mesh)

RESULT AVI (double-click on picture)



NOTE:
Free top and bottom boundaries on the ALE mesh allows the "bird" material to flow out of the mesh.

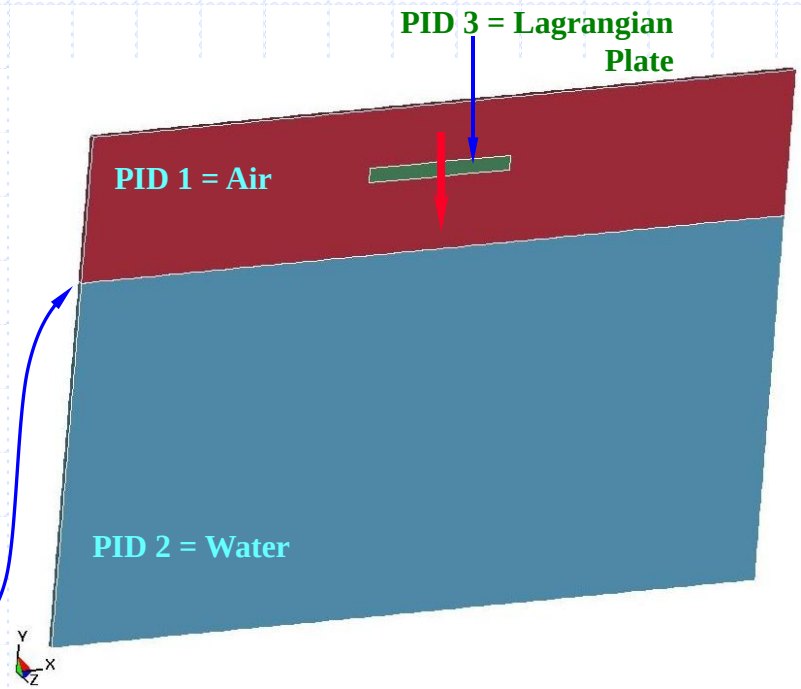
Lagrangian Plate hitting ALE Multi-Material Fluids

Lagrangian Plate hitting ALE Multi-Material Fluids

OVERVIEW:

A Lagrangian plate moves with "-y" velocity through air, then hits water.

- The Air and Water are defined as with ALE-Multi-Materials. Such formulation allows the mixing of the two material within each element.
- The Steel Plate is defined as Lagrangian.
- The Lagrangian mesh can overlap the ALE mesh.
- The ALE-Multi-Material meshes must have merged nodes on their shared boundaries (they share the same nodes).

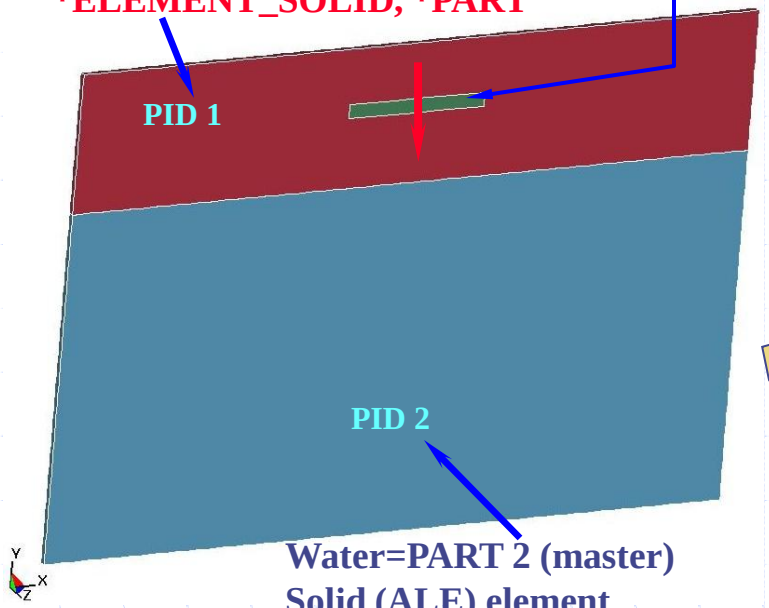


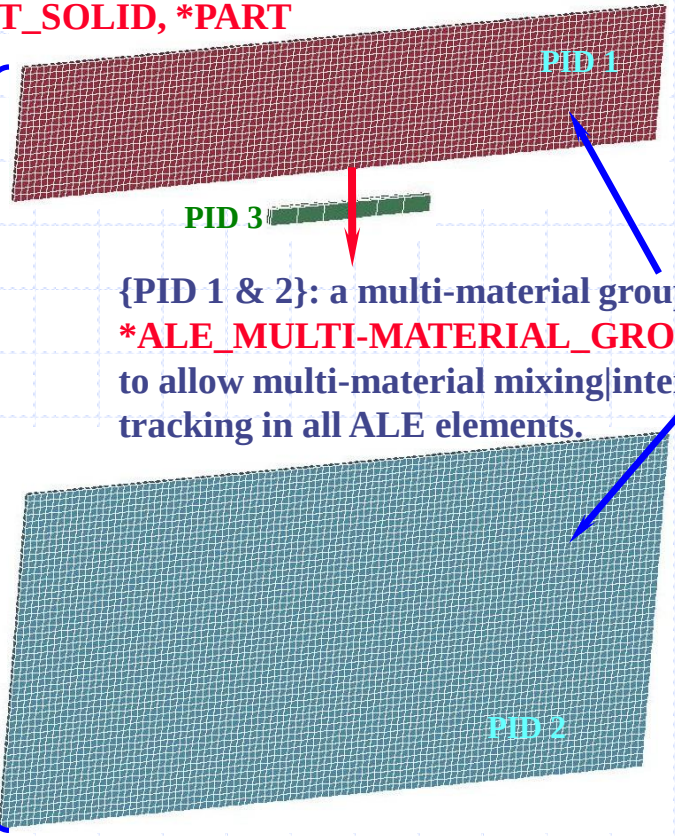
Lagrangian Plate hitting ALE Multi-Material Fluids

GEOMETRY DEFINITIONS:

Air=PART 1 (master)
Solid (ALE) element
ELFORM=11=1-PT-ALE-MUL-MAT
*ELEMENT_SOLID, *PART

Steel Plate=PART 3 (slave)
Solid (Lagrangian) element
ELFORM=1=Constant-Stress
*ELEMENT_SOLID, *PART





{PID 1 & 2}: a multi-material group
*ALE_MULTI-MATERIAL_GROUP
to allow multi-material mixing|interface tracking in all ALE elements.

Water=PART 2 (master)
Solid (ALE) element
ELFORM=11=1-PT-ALE-MUL-MAT
*ELEMENT_SOLID, *PART

Lagrangian Plate hitting ALE Multi-Material Fluids

MODEL ORGANIZATION & INITIAL CONDITION:

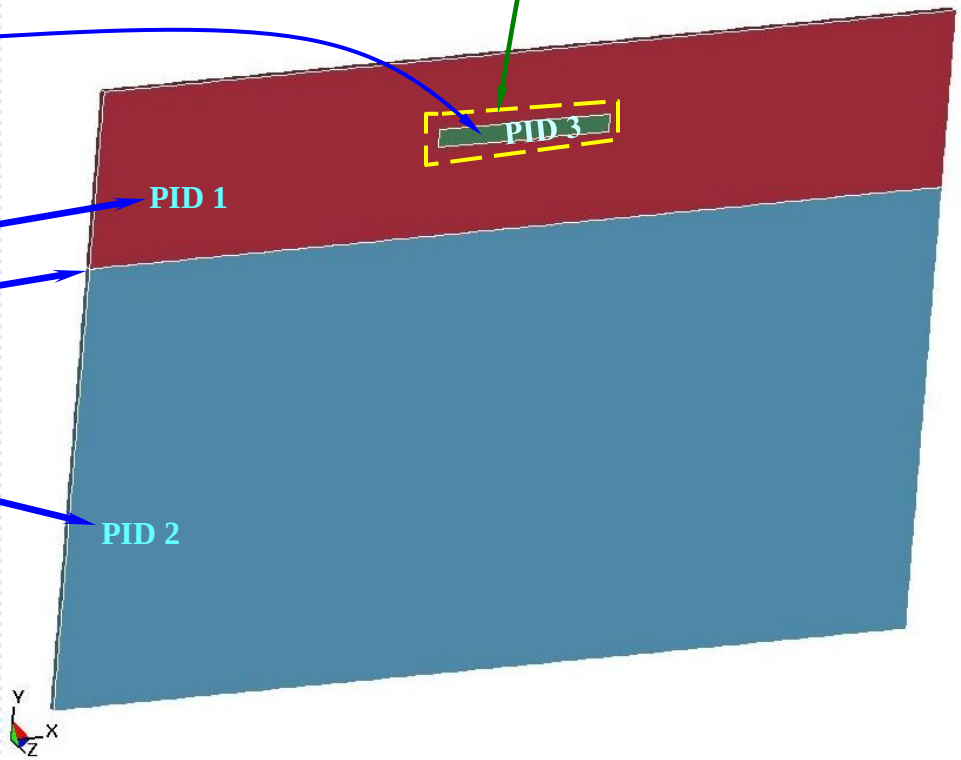
Grouping PID 1 & 2 into PSID 1(master)

Grouping all surface segments of PID 3 into SGSID 1(slave)

Defining initial velocity for PID 3.

*SET_SEGMENT
SGSID 1
All surfaces of PID 3

To give PID 3 its initial velocity use:
*INITIAL_VELOCITY_GENERATION

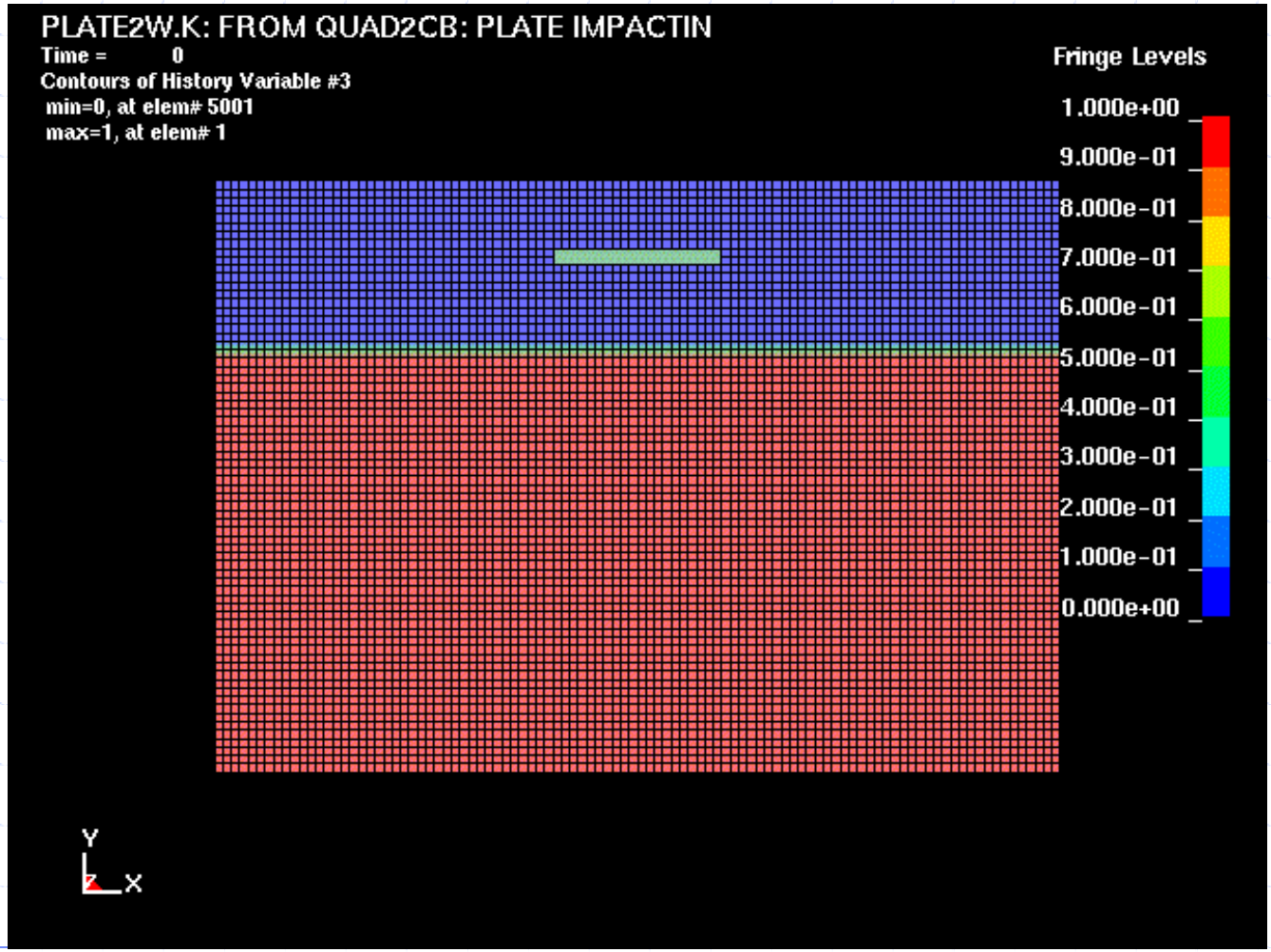


*SET_PART_LIST
PSID 1:{PID 1&2}

Merged nodes
on boundary

Lagrangian Plate hitting ALE Multi-Material Fluids

RESULTS: (double-click on picture below)



Lagrangian Plate hitting ALE Multi-Material Fluids

MODEL ORGANIZATION: SECTIONs, PARTs

```
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
$ Air & Water = Section 1
*SECTION_SOLID
$       SID    ELFORM       AET
         1        11
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
$ Plate = Section 2
*SECTION_SOLID
         2         1
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
*PART
FLUID, AIR, SURROUNDING, MASTER, 3D SOLID ELFORM=1PT-INTEG-MUL-MAT ELM
$       PID       SID       MID     EOSID      HGID      GRAV    ADPOPT    THERMID
         1         1         1         1
*PART
FLUID, WATER, TARGET, MASTER, 3D SOLID ELFORM=1PT-INTEG-MUL-MAT ELM
         2         1         2         2
*PART
PLATE, PROJECTILE, SLAVE, ELFORM=1=3D SOLID LAGRANGIAN FULL-INTEG-S/R ELM
         3         3         3         0         0         0         0         0
```

Air & Water: Eulerian Solid Multi-material

Projectile = Plate: Lagrangian Solid

For solid:
ELFORM=11=1-PT-ALE-MULTI-MAT
ELFORM=1=Default=Constant stress solid elm

Lagrangian Plate hitting ALE Multi-Material Fluids

MODEL ORGANIZATION: PART_SET, SEGMENT_SET

PSID 1 contains PID 1&2

SGSID 1 contains all surface segments of PID 3 (Plate)

```
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
$ DEFINE PSID's FOR FS COUPLING
*SET_PART_LIST
$       SID       DA1       DA2       DA3       DA4
         1
$      PID1      PID2      PID3      PID4      PID5      PID6      PID7      PID8
           1         2
$ SEGMENT LIST FOR FSC: DEFINE THE SURFACE ON THE LAGRANGIAN
*SET_SEGMENT
$       SID       DA1       DA2       DA3       DA4
         1
$        N1        N2        N3        N4        A1        A2        A3        A4
     14344     14358     14357     14343
     14358     14372     14371     14357
     14345     14359     14358     14344
     14359     14373     14372     14358
     14346     14360     14359     14345
     14360     14374     14373     14359
     14347     14361     14360     14346
     14361     14375     14374     14360
     14348     14362     14361     14347
     14362     14376     14375     14361
     14349     14363     14362     14348
     14363     14377     14376     14362
```

Surface of the impacting plate:
Each segment is a surface of a 3D solid elm. Each surface is 4-sided quad made up by 4 nodes.

Lagrangian Plate hitting ALE Multi-Material Fluids

PART INTERACTIONS:

PSID 1:{PID 1 & 2}← CONSTRAIN-COUPLING →SGSID 1:{PID 3 surfaces}.

```
*CONTROL_ALE
$      DCT      NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
         2         1         2        -1.
$    START       END     AAFAC     VFACT    VLIMIT       EBC
$ SGSID=1=slave; PSID=1=master=PID 1&2
*CONSTRAINED_LAGRANGE_IN_SOLID
$    SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
         1         1         2         0         3         4         2         1
$    START       END      PFAC      FRIC
         0         0       1.0
```

3X3 quadrature

Penalty

Normal compression

Couple w/ higher rho

SGSID 1 (Segment set ID) SSTYP=2

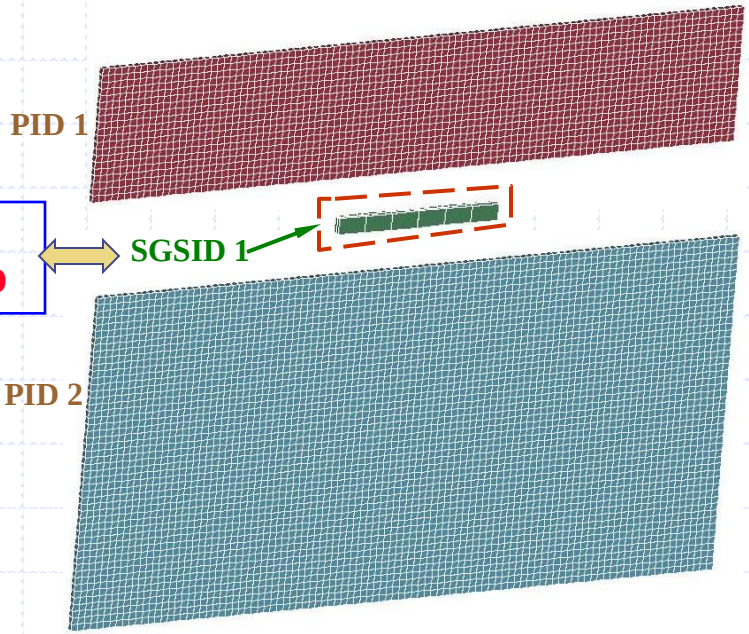
PSID 1: {PID 1&2} MSTYP=0

Penalty factor

Advection method: 2=Van Leer + HIS

PSID 1: {PID 1&2}

*CONTROL_ALE
*CONSTAINED_LAGRANGE_IN_SOLID



Lagrangian Plate hitting ALE Multi-Material Fluids

MATERIAL CONSTITUTIVE & EOS MODELS:

Use NULL constitutive model & Mie-Gruneisen EOS for Air and Water:

Pcutoff ≤ 0.0

vol/vol0 for tension erosion

vol/vol0 for compression erosion

Density

Viscosity

Young Modulus

Poisson's Ratio

sound speed

```
$ AIR: ALT=0M,T=0C:VISC=17.456E-6N*S/M^2; KINVISC=13.942E-6M^2/S,CP=1011J/(KG*K)
$      V_sound (M/S) = 331.5 + 0.6*T_C => @20C V_sound=343.7M/S
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
         1 1.2520000       0.0 17.456E-6       0.0       0.0       0.0       0.0
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMA0         A        E0
         1  343.7000       0.0       0.0       0.0      1.40       0.0       0.0
$       V0
       0.0
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
$ H2O:T=20C:VISC=1.002E3N*S/M^2;RHO=998.21KG/M^3;CP=4.1818E3J/(KG*K)
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
         2    998.21       0.0  1.002E+3       0.0       0.0       0.0       0.0
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMA0         A        E0
         2   1.647E3     1.921    -0.096       0.0     0.350       0.0       0.0
$       V0
       0.0
```

More details on the Mie-Gruneisen EOS follows …

Lagrangian Plate hitting ALE Multi-Material Fluids

MATERIAL CONSTITUTIVE & EOS MODELS:

Use Johnson-Cook constitutive model & Mie-Gruneisen EOS for Steel Plate.

Density
Shear Modulus
Young Modulus
Poisson's Ratio
Min dt for elm delete
Rate effect
Effective plastic strain rate

```
$ 304 STAINLESS STEEL IMPACTOR (Tensile strength ~ 900.0 MPa)  ~ kg-m-s
*MAT_JOHNSON_COOK
$      MID       RHO         G          E         PR        DTF        VP
         3    7840.0    80.0E+9  210.0e+9        0.3        0.0       0.0
$        A         B         n          C         m      Tmelt     Troom      EPSO
  792.19e6  509.51e6      0.26      0.014      1.03   1793.15    298.15         1
$       Cp        PC     spall         IT        D1         D2        D3        D4
       477    -5.2e8       0.0        0.0   -8.0e-1    2.1e00   -5.0e-1   2.0e-3
$       D5
    6.1e-1
$---5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
*EOS_GRUNEISEN
         3   4.570E3      1.49       0.0       0.0     1.930       0.0       0.0
       0.0
```

specific heat

Failure stress or P_cutoff

Spall model

Plastic strain iteration option

A, B, n, C, m, D_1-D_5 are defined in the following …

Lagrangian Plate hitting ALE Multi-Material Fluids

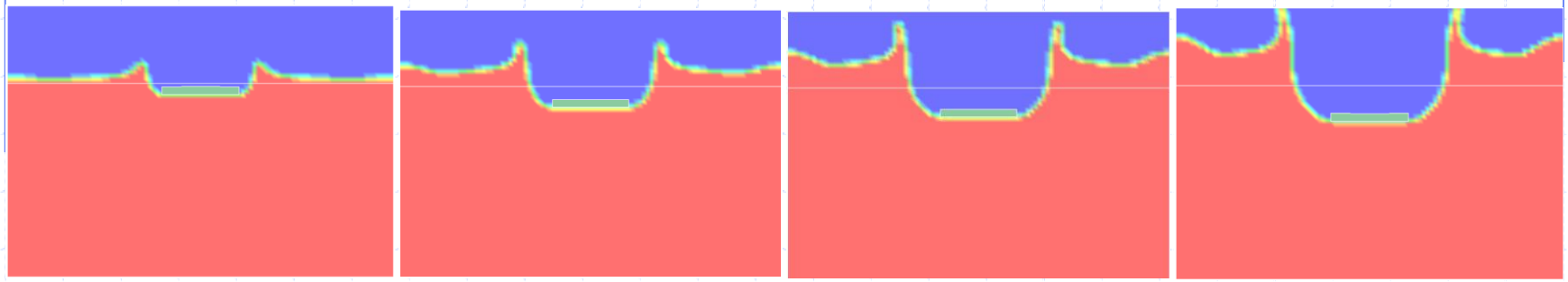
EXAMPLE SUMMARY:

- Void space is substituted by "Air" in this simulation,thus necessitate the use of *ALE_MULTI-MATERIAL-GROUP to mix the 2 Eulerian materials, air and water, in the ALE mesh.
- Interactions between the plate (slave) and the fluids (master) is modeled with a coupling to the surface-segment-set of the plate instead of its PART NUMBER.
- Real material properties and more sophisticated constitutive models are used in this example instead of scaled-and-simplified data and models. This increases computation time significantly.

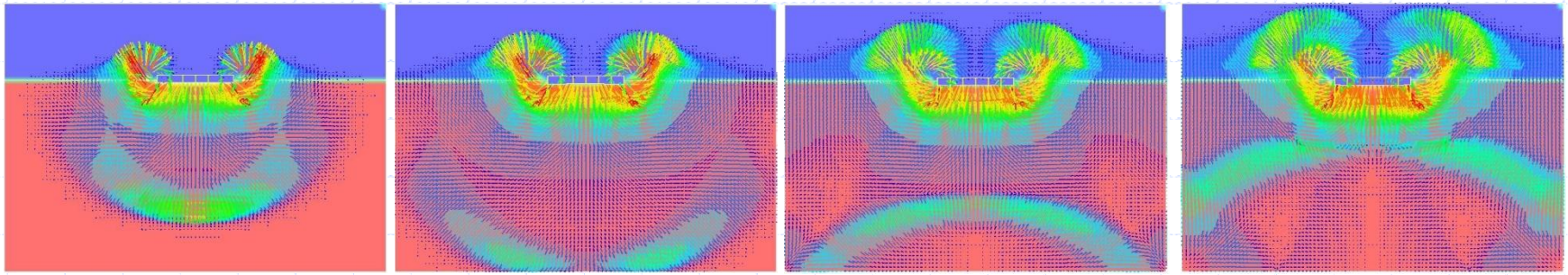
Lagrangian Plate hitting ALE Multi-Material Fluids

SOME RESULTS:

Water interface profile … (Note that the lateral walls do not allow outflow).



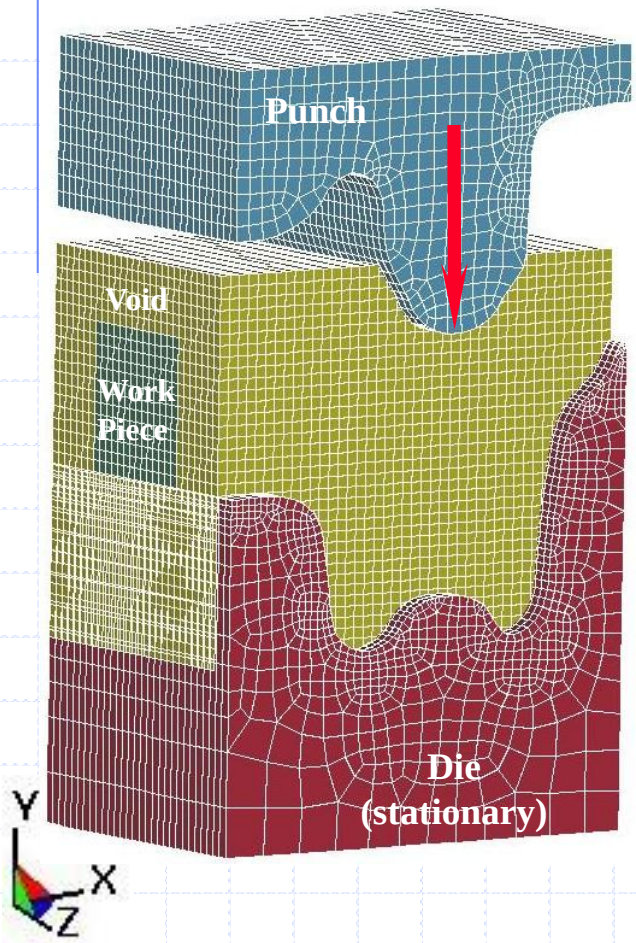
Early velocity vector profile showing the shock wave reflection in the water from the bottom boundary. Note the very fast wave reflection due to high acoustic velocity in water.



3D FORGING
Rigid Punch-Die System & ALE Work Piece

3D Forging: Rigid Tools & Deformable Work-Piece

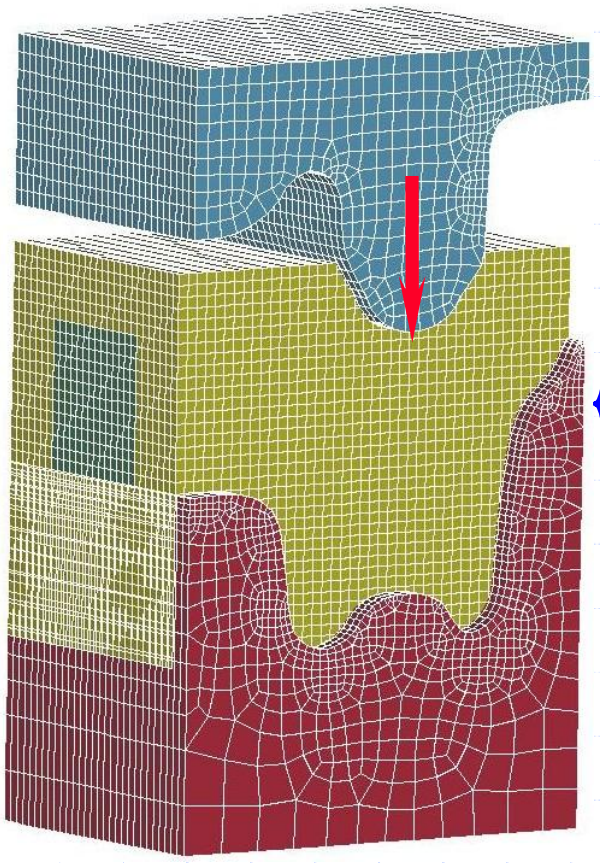
OVERVIEW:



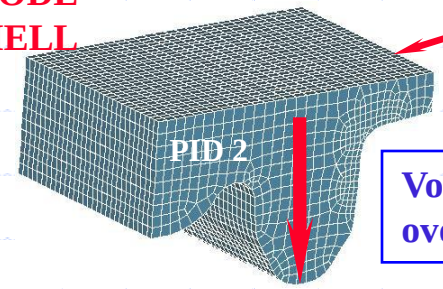
- The punch (top tool piece) moves down and press the work piece into the stationary die (bottom tool piece).
- Both tool pieces, punch and die, are modeled as Lagrangian *rigid shell* structures (we only care for their surface topologies).
- The work piece is modeled as *solid* ALE material which is allowed to deform flow into surrounding void space.
- A space with exactly the shape of the work piece is delete from the initial void mesh. Then the work piece is inserted into this space. The meshes of the void and work piece share the same node ID's on their common boundaries (*merged nodes*).
- The ALE void mesh can overlap with the Lagrangian rigid tool structures.

3D Forging: Rigid Tools & Deformable Work-Piece

GEOMETRY:



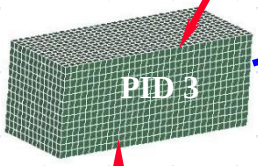
*NODE
*ELEMENT_SHELL

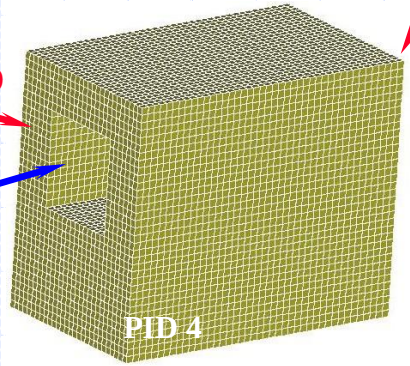


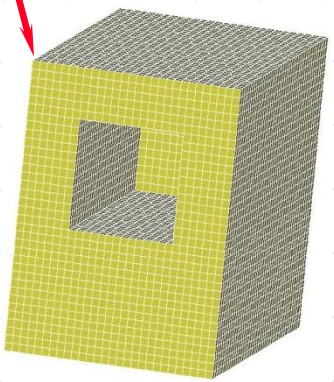
Rigid punch (moving down)

Void mesh (static & overlapping tools)

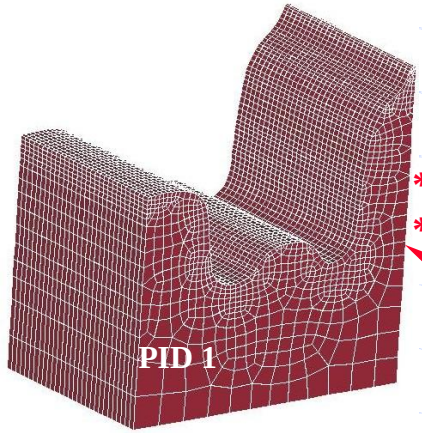
*NODE
*ELEMENT_SOLID







Work piece



*NODE
*ELEMENT_SHELL

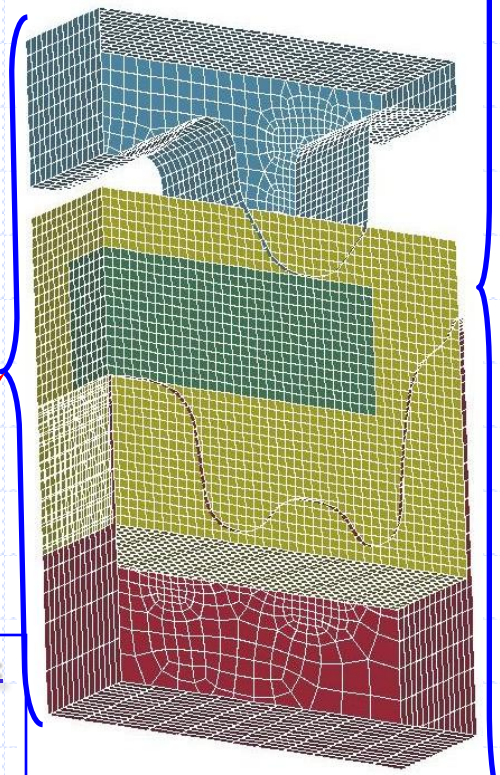
Rigid die (fixed)

3D Forging: Rigid Tools & Deformable Work-Piece

ORGANIZE MODEL

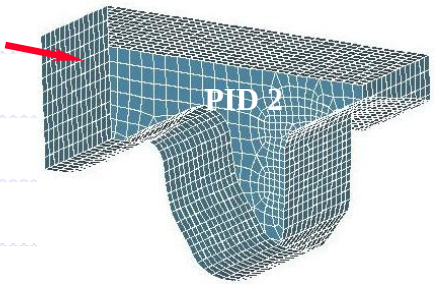
As the mesh for each part is created, appropriate boundary conditions can be defined for the boundary nodes of the meshes. (This can be handled in the preprocessing step.)

Cross-Section of Assembly

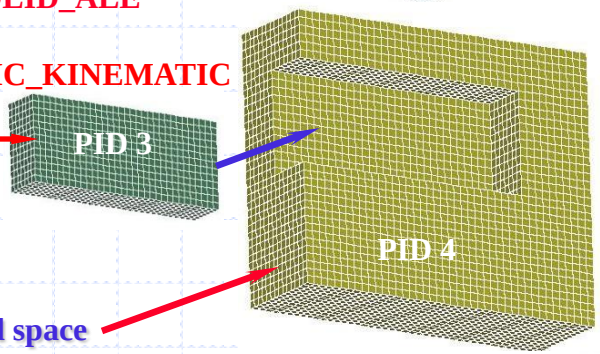


Basic geometry defs + BC's:
*NODE
*ELEMENT_SOLID
*ELEMENT_SHELL

PART 2: Rigid Shell
*SECTION_SHELL
*PART
*MAT_RIGID

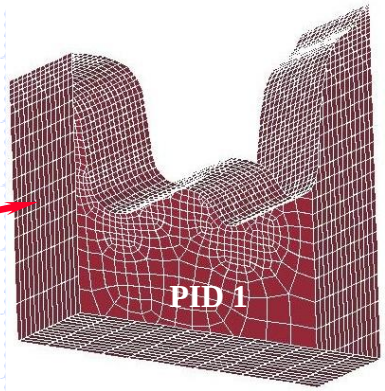


PART 3: Work piece
*SECTION_SOLID_ALE
*PART
*MAT_PLASTIC_KINEMATIC



PART 4: Void space
*SECTION_SOLID_ALE
*PART
*INITIAL_VOID

PART 1: Rigid Shell
*SECTION_SHELL
*PART
*MAT_RIGID



3D Forging: Rigid Tools & Deformable Work-Piece

IC's & BC's :

*BOUNDARY_PRESCRIBED_MOTION_RIGID
(moving)

```
*SET_PART_LIST
         2
         3          4
```

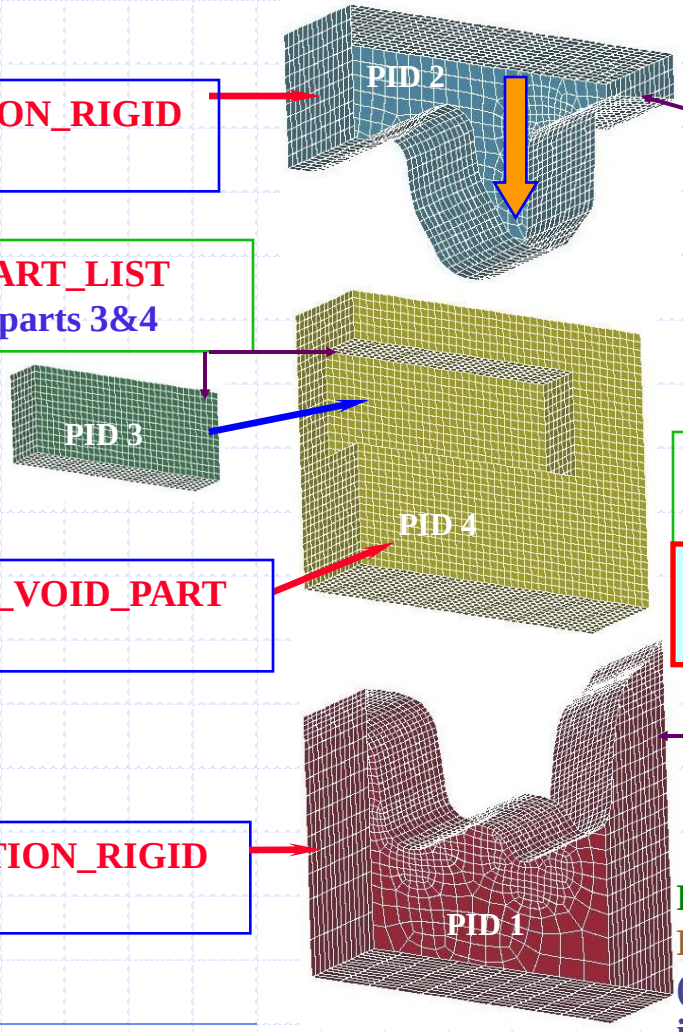
*SET_PART_LIST
PSID=2:parts 3&4

*INITIAL_VOID_PART
4

*SET_PART_LIST
PSID=1:parts 1&2

```
*SET_PART_LIST
         1
         1          2
```

*BOUNDARY_PRESCRIBED_MOTION_RIGID
(fixed)



PSID=1: Lagrangian group.
PSID=2: ALE group.
(These will be used for interactions and coupling modeling).

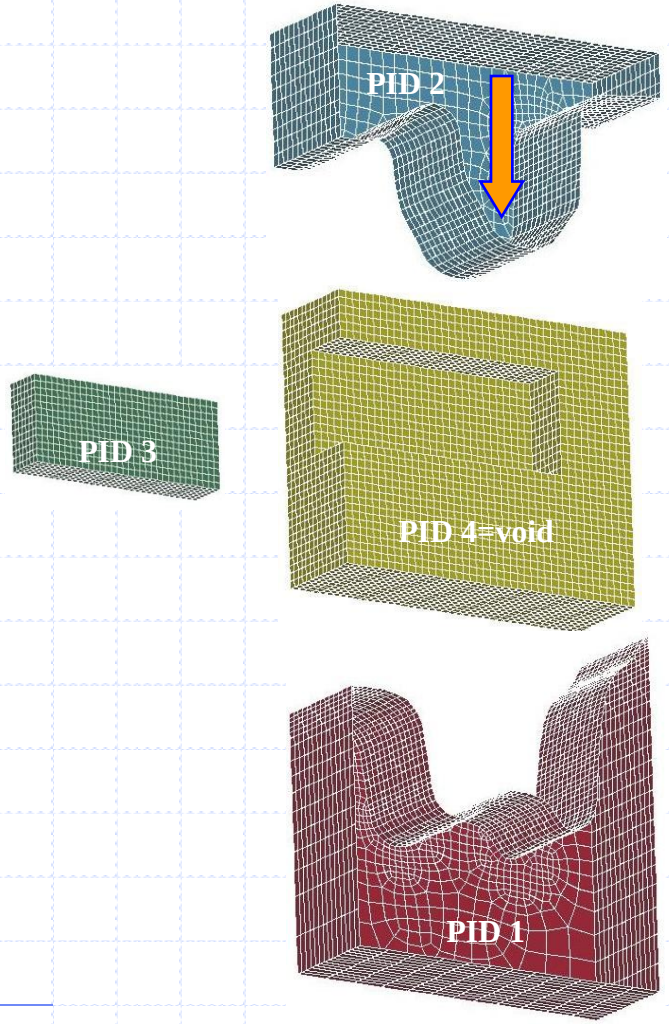
3D Forging: Rigid Tools & Deformable Work-Piece

COUPLING DEFINITIONS:

Master
PSID=2:parts 3&4

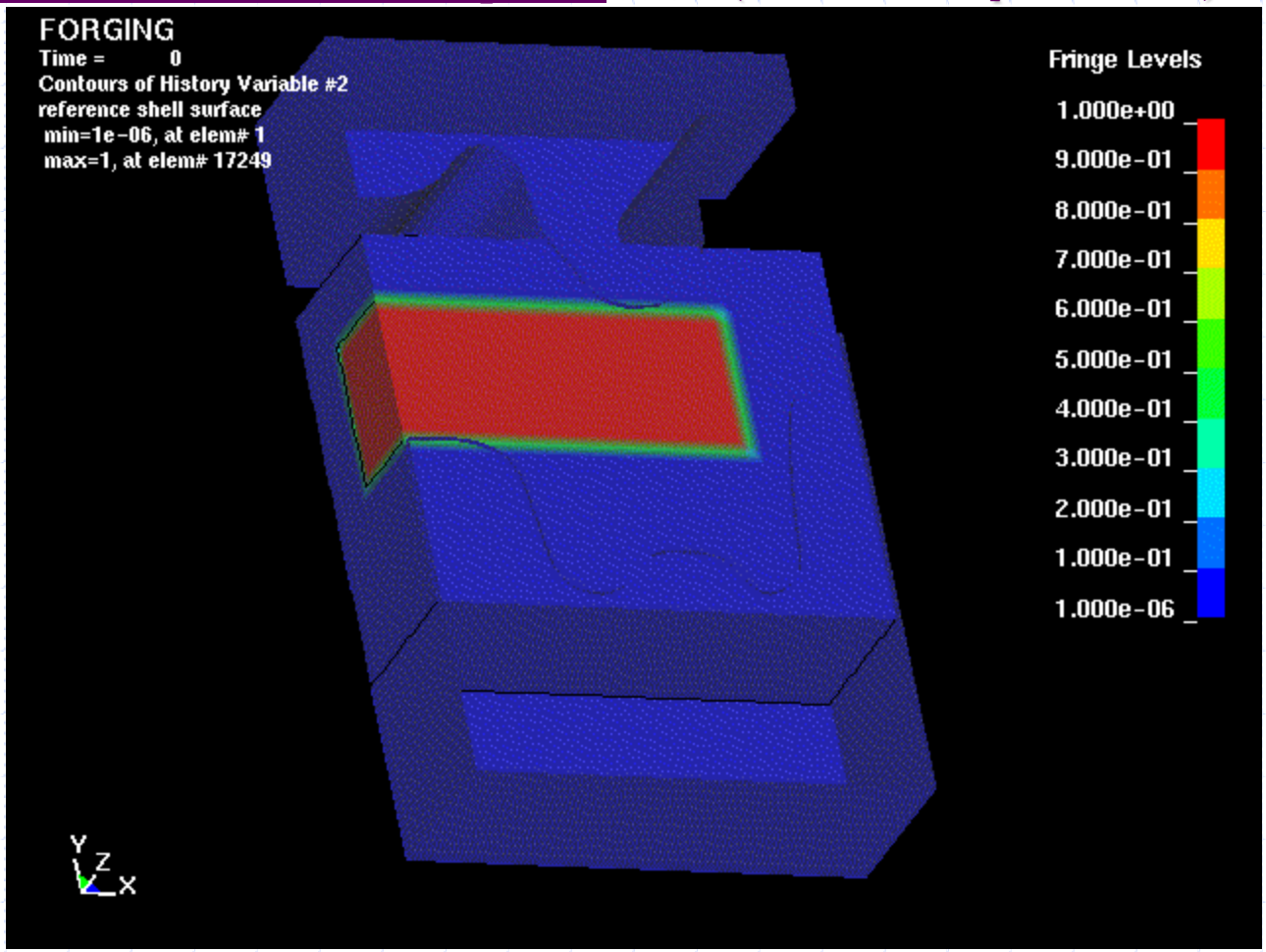
*CONTROL_ALE
*CONSTAINED_LAGRANGE_IN_SOLID

Slave
PSID=1:parts 1&2



3D Forging: Rigid Tools & Deformable Work-Piece

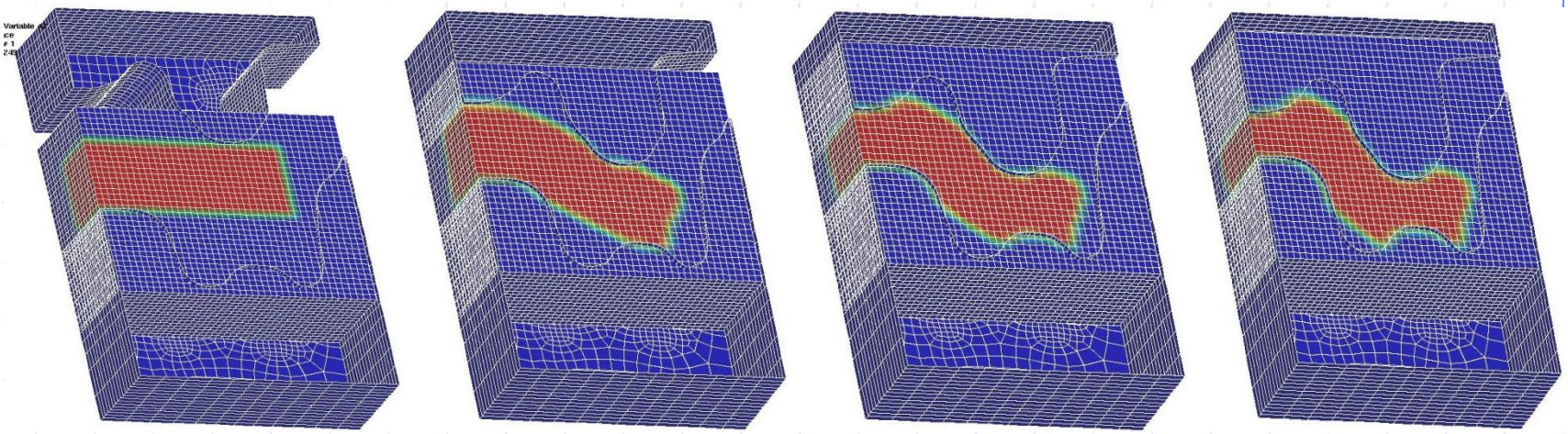
Result viewed at a cross-section plane: AVI (double-click on picture below)



3D Forging: Rigid Tools & Deformable Work-Piece

SOME RESULTS (viewed at a cross-section plane) :

Some volume-fraction plots of the work-piece showing the forging process.



3D Forging: Rigid Tools & Deformable Work-Piece

MODEL DEFINITION: SECTIONs, PARTs

For shell: ELFORM=0=Default=2=Belytschko-Tsay

Punch & Die (Rigid Shell)

Work piece & void (Solid)

```
*PART
Bottom die RIGID SHELL SURFACE (fixed)
$      PID       SID       MID     EOSID      HGID      GRAV    ADPOPT    THERMID
         1         1         1         0         0         0         0          0
*PART
Top punch RIGID SHELL SURFACE (moving down -y)
         2         1         1         0         0         0         0          0
*SECTION_SHELL
$     SECID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP     SETYP
         1         0
$       TH1       TH2       TH3       TH4      NLOC
       0.1       0.1       0.1       0.1
*PART
Work piece at core of 3D solid elm mesh
         3         2         2         0         0         0         0          0
*PART
Void mesh surrounding the work piece
         4         2         2         0         0         0         0          0
*INITIAL_VOID_PART
         4
*SECTION_SOLID_ALE
$     SECID    ELFORM       AET
         2        12
$      AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
```

For solid: ELFORM=12=1-PT-INTEG-1MAT+VOID

3D Forging: Rigid Tools & Deformable Work-Piece

IC's & BC's :

For DOF=4 or 8: Vector ID for motion defined by *DEFINE_VECTOR

Impose nodal motion (u,v or a) on a NID,NSID or PID.

```
$ Fix the die and give the punch is constant velocity using load curves
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$ NID|*PID        DOF        VAD       LCID         SF        VID      DEATH      BIRTH
         1          2          0          2        1.0
*BOUNDARY_PRESCRIBED_MOTION_RIGID
         2          2          0          1        1.0
```

Y-dir motion only

Load curve ID's

Scale factor for load curve

VAD=motion type

For nodes:

0=Vel

1=Accel

2=Displacement

For rigid bodies:

0=Vel

2= Displacement

3=Vel-VS-Displ.

4=relative Displ.

```
$ Define the load #1: MOTION FOR THE TOP PUNCH
*DEFINE_CURVE
$      LCID      STDR       SFO      OFFA      OFFO    DATTYP
         1
$          time          velocity
             0.0            -500.0
             1.0            -500.0
$ Define the load #2: NO MOTION FOR THE BOTTOM DIE
*DEFINE_CURVE
         2
             0.0             0.0
             1.0             0.0
```

3D Forging: Rigid Tools & Deformable Work-Piece

PART INTERACTIONS:

{PID 1 & 2}:PSID 1 ← CONSTRAIN-COUPLING → PSID 2:{PID 3 & 4}.

Advection method:
4=Donor + HIS
(1st order)

No-smoothing

```
*CONTROL_ALE
$       DCT       NADV       METH       AFAC       BFAC       CFAC       DFAC       EFAC
          2          1          4       -1.0
$     START        END      AAFAC      VFACT     VLIMIT        EBC
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE     MASTER     SSTYP      MSTYP      NQUAD      CTYPE      DIREC      MCOUP
          1          2          0          0          1          4          2          1
$     START        END       PFAC       FRIC
        0.0        0.0        0.1        0.2
```

1 quad pt

Penalty

Normal compression

Couple w/ higher rho

PSID 1 (PID 1&2) SSTYP=0

PSID 2: {PID 3&4} MSTYP=0

Penalty factor

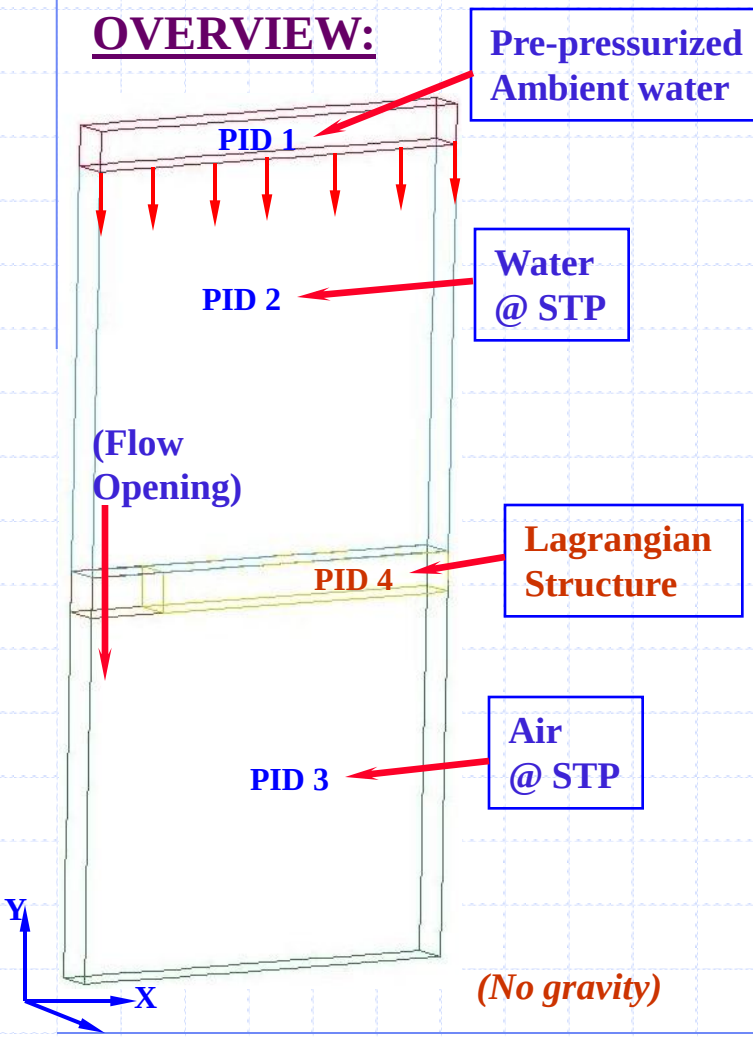
Friction factor

PSID 2: {PID 3&4} ↔ *CONTROL_ALE *CONSTAINED_LAGRANGE_IN_SOLID ↔ PSID 1: {PID 3&4}

PURGING FLOW THROUGH ORIFICE
Fluid VS. Orifice Interaction

Purging Flow Interaction with Orifice

OVERVIEW:



(No gravity)

- **Part 1**, top row, consists of ambient elements of type "Pressure-Inflow". This reservoir inlet water is pre-pressurized to $P_0 \sim (\gamma-1)*e_{v0} \sim$ 3.0E-5 Mbar.
- **Part 2** consists of water at STP.
- **Part 3** consists of air at STP.
- All 3 parts (PIDs 1, 2 & 3) have element formulation type **1-Point-ALE-Multi-Material (ELFORM=11)**.
- **Part 4** is a Lagrangian structure deflected by the flowing fluid. It is made of rubber.

***PID 4** mesh overlaps with **PID 3** mesh.*

***PID 4** (slave) interacts with all 3 fluid parts (master: **PIDs 1, 2 & 3**).*

Purging Flow Interaction with Orifice

GEOMETRY & ORGANIZATION:

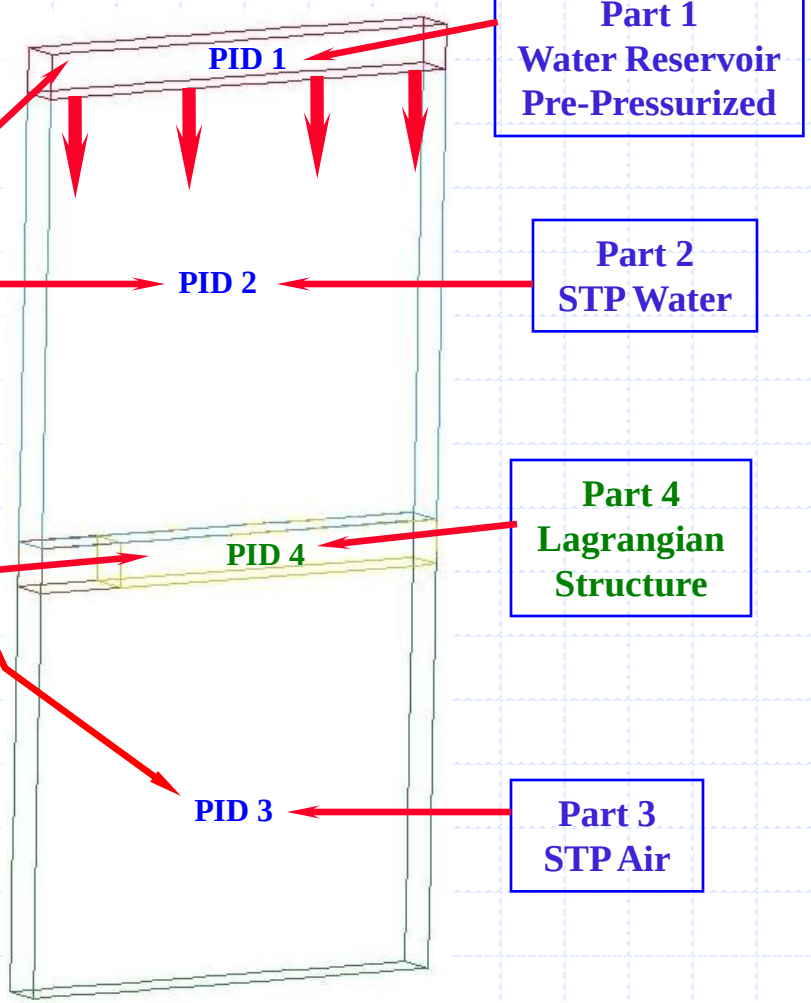
All part geometry are defined with
*NODE
*ELEMENT_SOLID

(ALE-Multi-Mat=Master)
*SECTION_SOLID_ALE
ELFORM=11=1-Pt-ALE-Multi-Mat
AET=4=Ambient elm type (for PID 1 ONLY)
*PART

(Lagrangian=Slave)
*SECTION_SOLID
ELFORM=0
*PART

PSID 2 contains the 3 fluids PIDs 1, 2 & 3:

```
*SET_PART_LIST
     2
     1       2       3
```



Purging Flow Interaction with Orifice

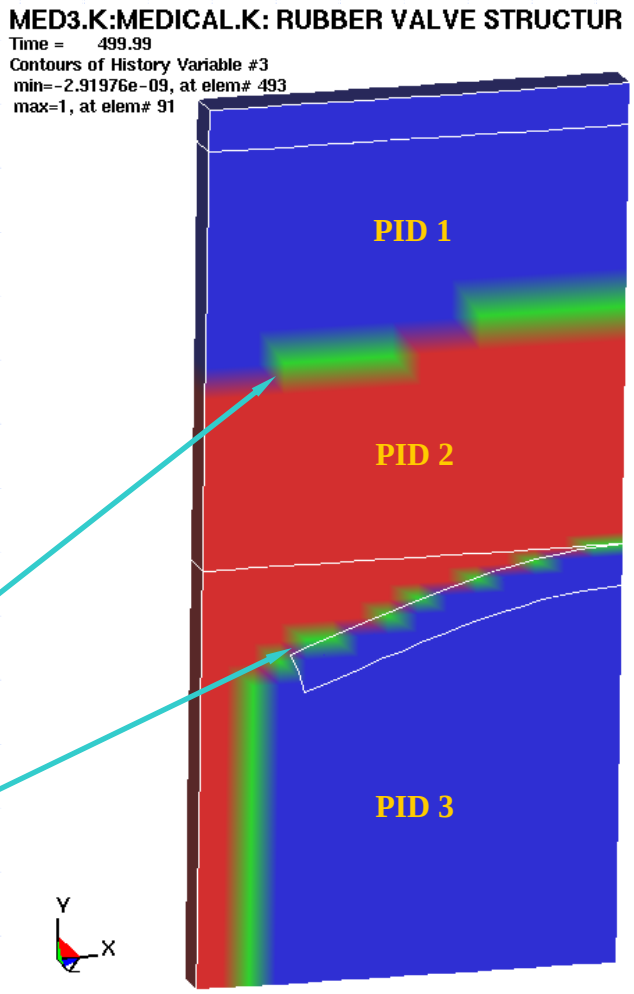
BC's & IC's & ALE SYSTEM DEFINITIONS:

The *ALE_MULTI-MATERIAL_GROUP command to turn on the interface tracking option for multi-materials 1, 2 & 3 in 1 single element.

```
*ALE_MULTI-MATERIAL_GROUP
      1       1
      2       1
      3       1
```

Interface of PID 2 with respect to PID 1

Interface of PID 2 with respect to PID 3

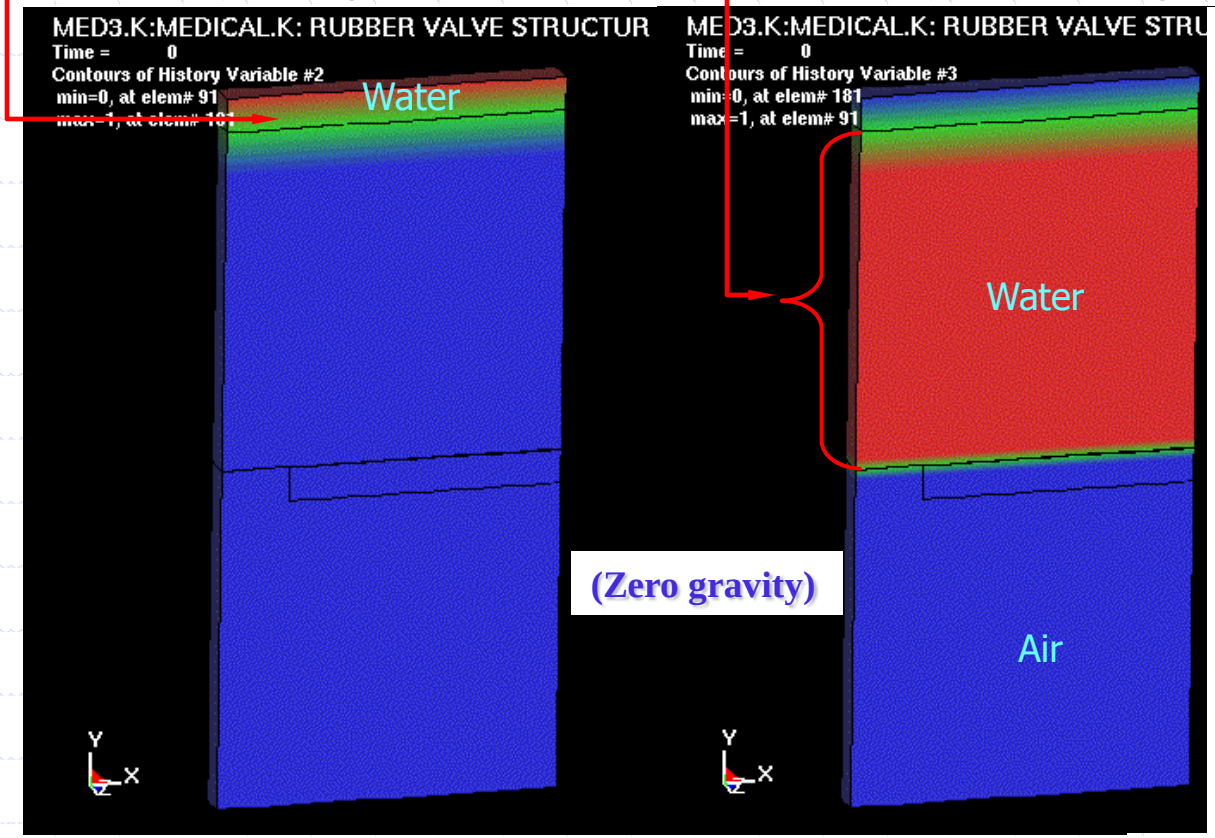


Purging Flow Through An Orifice (multi-fluids)

AVI's (double-click on pictures below)

RESULTS: Priming Fluid Reservoir (Initially Pressurized Fluid)

Fluid Being Purged (Fluid initially at ambient condition).



Priming fluid flow

Purging fluid flow

NOTE:

Very coarse mesh is being used (just for demonstration purpose).

Purging Flow Interaction with Orifice

PRESSURIZED RESERVOIR = PID 1:

For solid: ELFORM=11=Multi-Material

```
*PART
[1] Water - pressurized reservoir: material type # 9 = null material
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         1         0         0         0         0
*SECTION_SOLID_ALE
$    SECID    ELFORM       AET
         1        11         4
$     AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
  .0000000  .0000000  .0000000  .0000000  .0000000  .0000000
$ ---------------------------------------------------------------------------
*MAT_NULL
$      MID       RHO        PC        MU     TEROD     CEROD        YM        PR
         1   0.99820  .0000000  9.982e-9  .0000000  .0000000  .0000000  .0000000
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMMA         A       EV0
         1     0.165  1.920000  .0000000  .0000000  0.100000  .0000000 3.0000e-4
$       V0
  .0000000
```

AET=4=Ambient element simulating a reservoir

The Gruneisen EOS gives $P = P_c + P_T = A(\mu) + B(\mu) \cdot E$

The initial pressure is, with $v/v_0=1$ ➔ $\mu=0$, and "a"=0 and $B(\mu) = (\gamma_0 + a\mu)$

$$P_0 = \gamma_0 e_0$$

Internal Energy / Initial Volume

Purging Flow Interaction with Orifice

FLUID TO BE PURGED = PID 2:

For solid: ELFORM=11=Multi-Material

```
*PART
[2] Water @ STP: material type # 9 = null material
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         2         2         0         0         0         0
*SECTION_SOLID_ALE
$    SECID    ELFORM       AET
         2        11
$     AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
  .0000000  .0000000  .0000000  .0000000  .0000000  .0000000
$ ---------------------------------------------------------------------------
*MAT_NULL
$      MID       RHO        PC        MU     TEROD     CEROD        YM        PR
         2 0.9982e00  .0000000  9.982e-9  .0000000  .0000000  .0000000  .0000000
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMMA         A       EV0
         2     0.165  1.920000  .0000000  .0000000  0.100000  .0000000       0.0
$       V0
  .0000000
```

The initial pressure is $P_0 = 0$

Typically entered material data.

Purging Flow Interaction with Orifice

AMBIENT FLUID = PID 3:

```
*PART
External air receiving the outflow = NULL = mat type #9.
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         3         3         3         3         0         0         0         0
*SECTION_SOLID_ALE
$    SECID    ELFORM       AET
         3        11
$     AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
  .0000000  .0000000  .0000000  .0000000  .0000000  .0000000
$ *EOS_LINEAR_POLYNOMIAL: air c4=c5=γ-1=1.4-1=0.4  for air
*MAT_NULL
$      MID       RHO        PC        MU     TEROD     CEROD        YM        PR
         3 1.293e-3  .0000000 1.850e-10 .0000000  .0000000  .0000000  .0000000
*EOS_LINEAR_POLYNOMIAL
$    EOSID        C0        C1        C2        C3        C4        C5        C6
         3 .0000000  .0000000  .0000000  .0000000  .4000000  .4000000  .0000000
$      EV0        V0
  .0000000  .0000000
```

$$e_{\upsilon 0} = 0$$

The initial pressure is $$P_0 = (\gamma - 1)\frac{\rho}{\rho_0}\left[e_{\upsilon 0}\right]_0 = 0$$

Typically entered material data.

Purging Flow Interaction with Orifice

LAGRANGIAN STRUCTURE = PID 4:

```
*PART
Material Type # 27 (Lagrangian valve)
$      PID      SECID        MID      EOSID       HGID       GRAV    ADPOPT       TMID
         4          4          4          0          0          0         0          0
*SECTION_SOLID
$    SECID    ELFORM
         4          1
$---------------------------------------------------------------------------------
$ POLYRUBBER: g-cm-mcrs-K-Mbar;
*MAT_MOONEY-RIVLIN_RUBBER
$      MID        RHO         PR          A          B        REF
$        4       1.01      0.499    0.13292     0.0263        0.0
         4       1.01      0.499   0.013292    0.00263        0.0
$      SGL         SW         ST       LCID
```

For solid: ELFORM=1=Constant stress solid element

Harder (A & B)

Softer

Defined this 2nd card only if A=B=0

The "valve" Lagrangian structure is modeled using Mooney-Rivlin rubber material model. This requires 3 parameters:

PR=Poisson ratio and

A & **B** = coefficients in the strain energy density function.

Purging Flow Interaction with Orifice

FLUID-STRUCTURE-INTERACTION:

Coupling the Lagrangian structure. (PID 4)

Coupling in normal direction and under compression only

```
*CONTROL_ALE
$       DCT       NADV       METH       AFAC       BFAC       CFAC       DFAC       EFAC
         2          1          2    -1.0000 0.0000000   .0000000 0.0000000
$     START        END      AAFAC      VFACT     VLIMIT        EBC

*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE     MASTER     SSTYP      MSTYP      NQUAD      CTYPE      DIREC      MCOUP
          4          2          1          0          4          5          2          0
$     START        END       PFAC       FRIC    FRACMIN     NORMAL
$        Cq       Hmin       Hmax      ILEAK

*SET_PART_LIST
         2
         1          2          3
```

The 3 fluids considered here (PSID 2)

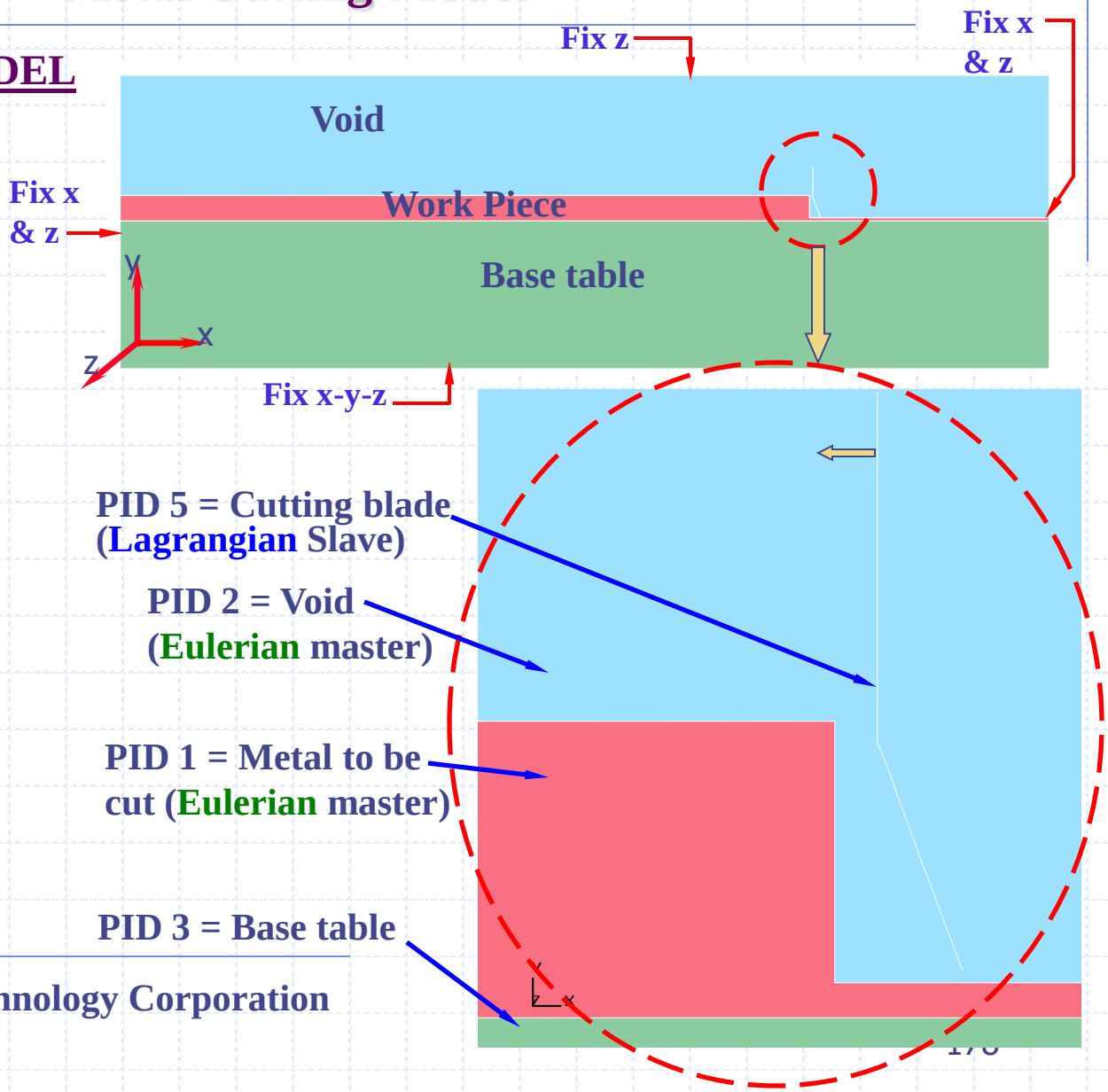
Coupling to Lagrangian SOLID

METAL CUTTING
(Machining)

Metal Cutting Model

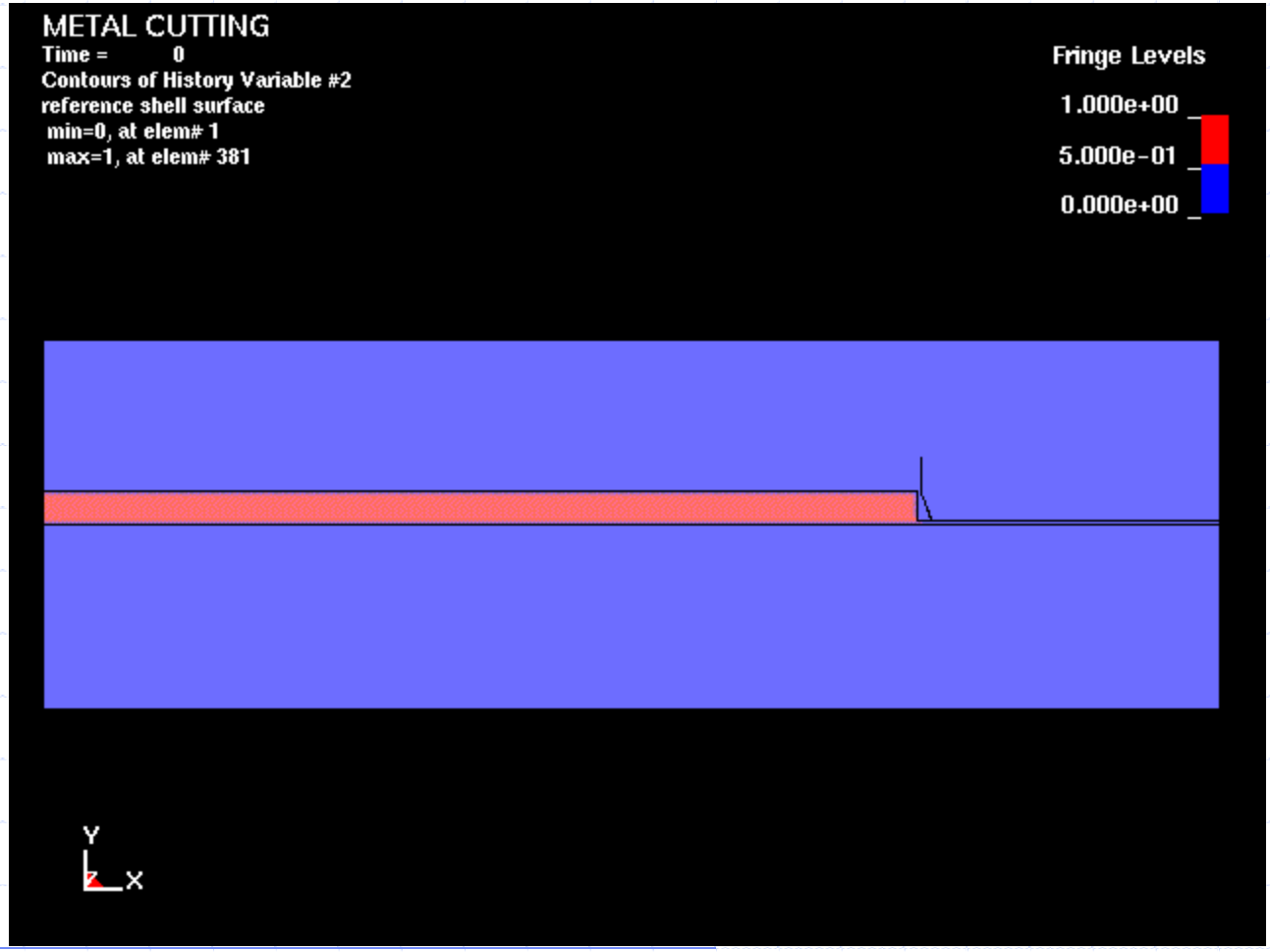
THE PHYSICAL MODEL

- The blade moves left cutting into the material at a prescribed (constant) rate.
- The work piece is "clamped" to the base table.
- Only PID 1 & 2 interact with the blade, PID 3 does not (thus no need for coupling to PID 3).



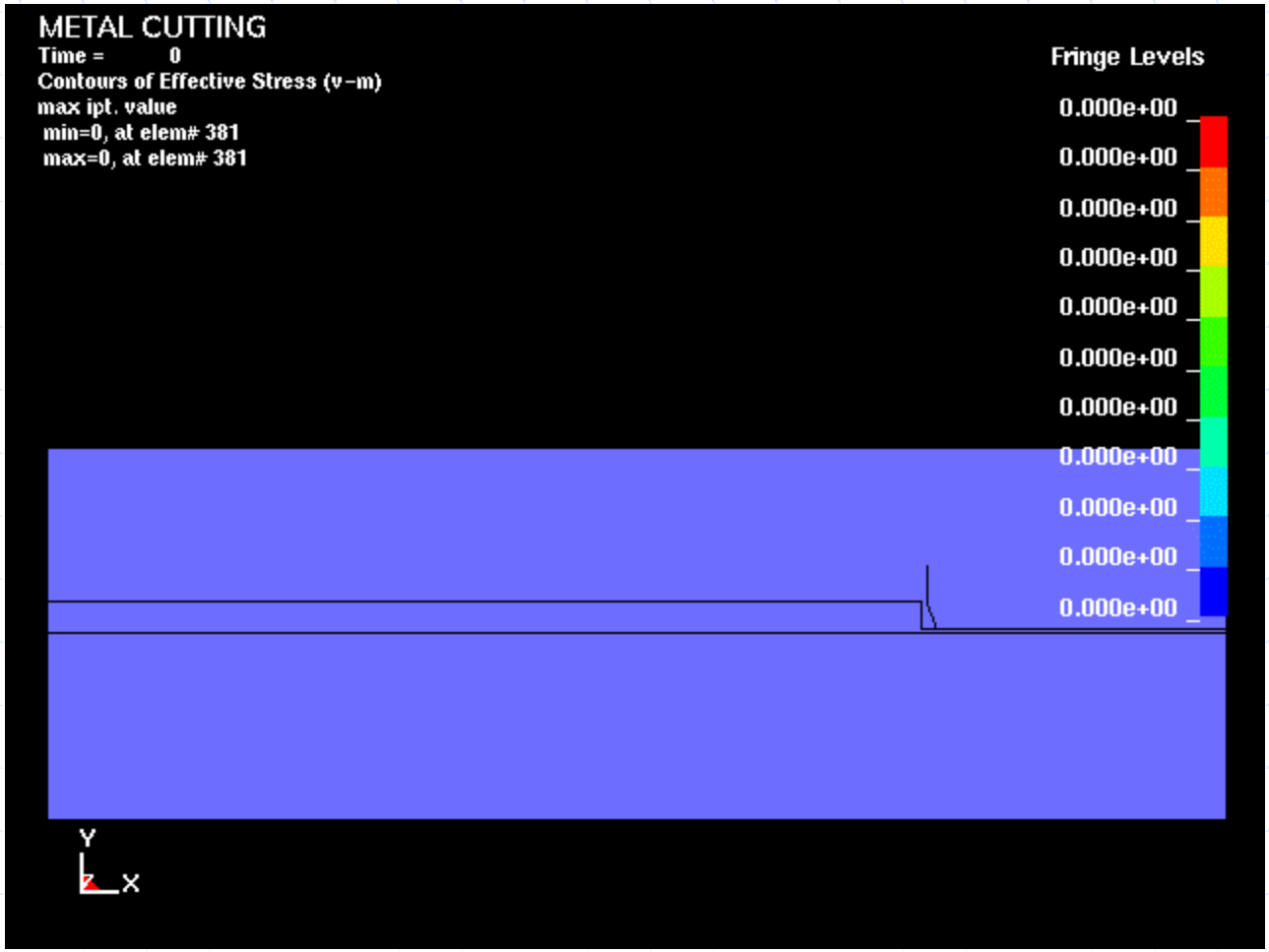
Metal Cutting Model

RESULTS: Material contour avi (double-click on the picture below)



Metal Cutting Model

RESULTS: Effective Stress avi (double-click on the picture below)



Metal Cutting Model

PART CONSTRUCTION – ALE parts for material being cut and void space:

PID 1

ELFORM=12

1 material + void

```
*PART
fluid part = material to be cut
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0         0         0
*SECTION_SOLID_ALE
         1        12
*MAT_PIECEWISE_LINEAR_PLASTICITY
$      MID        RO         E        PR      SIGY      ETAN      FAIL      TDEL
         1 7.8000-03 2.0700+11 0.3000000 4.0000+08 0.0000000 0.0000000
$        C         P      LCSS      LCSR        VP
$     EPS1      EPS2      EPS3      EPS4      EPS5      EPS6      EPS7      EPS8
       0.0       1.0      10.0
$      YS1       YS2       YS3       YS4       YS5       YS6       YS7       YS8
     5.0E8     8.0E8     8.0E8
$---------------------------------------------------------------------------------
*PART
void part of the 3D elms, providing the space for the cut material to flow
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         1         0         0         0         0         0
*SECTION_SOLID_ALE
         2        12
*INITIAL_VOID_PART
         2
```

Element formulation=12= 1 material + void

PID 2

This turns PID 2 into a void part

Metal Cutting Model

PART CONSTRUCTION – Rigid cutting blade & holding table:

PID 3 (table)

PID 5 (moving cutting blade)

```
*PART
The base table holding the work piece
$      PID      SECID        MID      EOSID       HGID       GRAV    ADPOPT       TMID
         3          3          1          0          0          0          0          0
*SECTION_SOLID
         3          0
$-------------------------------------------------------------------------------
$ TM=translational mass *PART_INERTIA applies only to part with *MAT_RIGID
*PART_INERTIA
Rigid cutting blade moving at prescribed velocity
$      PID      SECID        MID      EOSID       HGID       GRAV    ADPOPT       TMID
         5          5          3          0          0          0          0          0
$       XC         YC         ZC         TM       IRCS
         0.         0.         0.      1.e-7          0
$      IXX        IXY        IXZ        IYY        IYZ        IZZ   <-- Moment of Inertia
         1.         0.         0.         1.         0.         1.
$      VTX        VTY        VTZ        VRX        VRY        VRZ   <-- Inertial Velocity
         0.         0.         0.         0.         0.         0.
*SECTION_SHELL
         5          0
1.e-2,1.e-2,1.e-2,1.e-2   <-- Shell thickness
*MAT_RIGID
$      MID        RO         E         PR          N     COUPLE          M
         3 1.0000-02 2.0000+11 0.0000000          0          0          0
$      CMO      CON1      CON2
         0          0          0
$   LCO|A1         A2         A3         V1         V2         V3
         0          0          0          0          0          0
```

Metal Cutting Model

BOUNDARY CONDITION – Prescribed motion of the cutting blade:

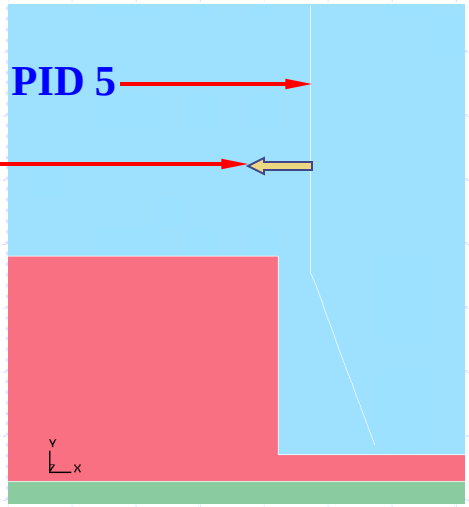
```
$ PID=5; DOF=1=x; VAD=0=vel : give the rigid part 5=blade a vel in x-dir
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$NID|NSID|PID     DOF       VAD      LCID        SF       VID     DEATH     BIRTH
         5         1         0         1       1.0       0.0       0.0       0.0
*DEFINE_CURVE
         1
   0.00000, -30000.0
   0.00001, -30000.0
   0.00007, -30000.0
   0.00071,   0.0
   0.000085, 30000.0
   1.00000,   30000.0
```

PID 5

VAD=0=Velocity

DOF=1=x

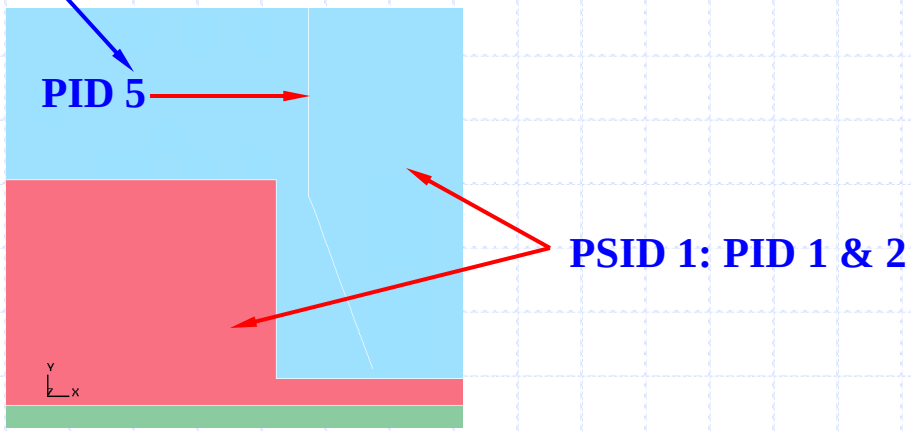
Velocity Curve
(moving in negative x direction)



Metal Cutting Model

FLUID-STRUCTURE-INTERACTION:

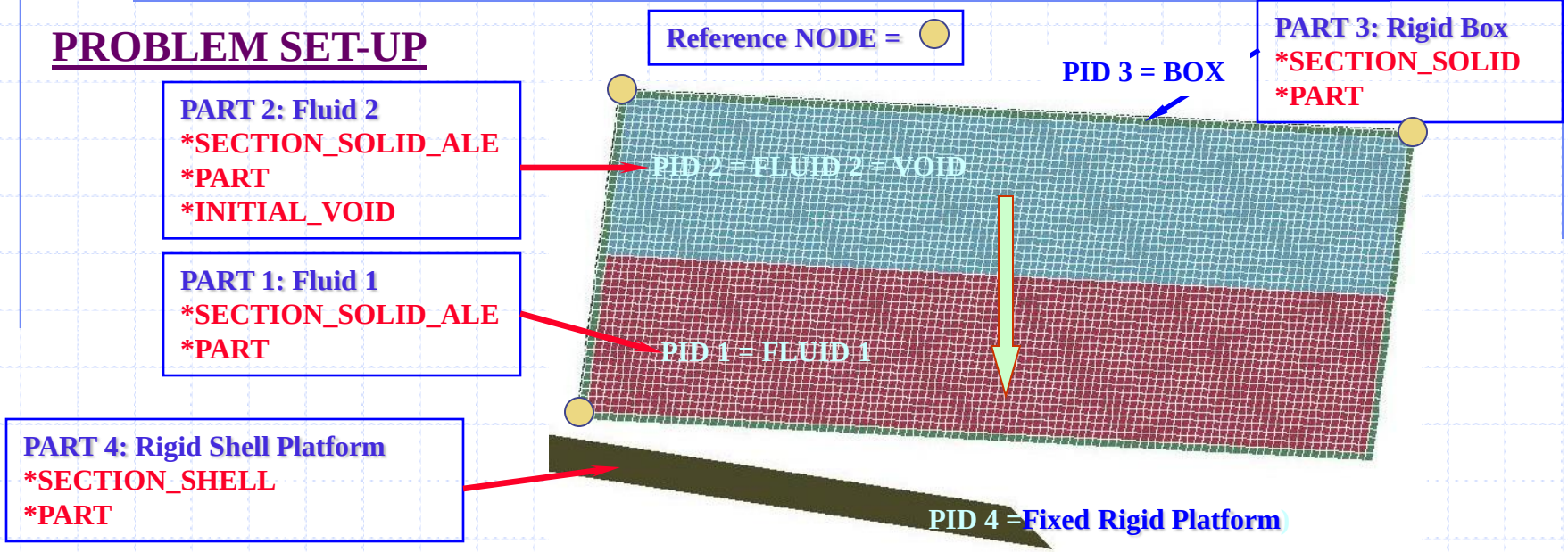
```
*CONTROL_ALE
$       DCT      NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
          2         1         2        -1.
$     START       END     AAFAC     VFACT    VLIMIT       EBC
          0         0         0     0.001         0         0
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
          5         1         1         0         3         4         2         1
$     START       END      PFAC      FRIC   FRACMIN    NORMAL
          0         0       0.0       0.1
$        Cq      Hmin      Hmax     ILEAK
          0         0         0         0
*SET_PART_LIST
          1
          1         2
```



ALE System Following A Reference 3-Node System
Dropping of a Fluid Box

Dropping Of A Fluid Box

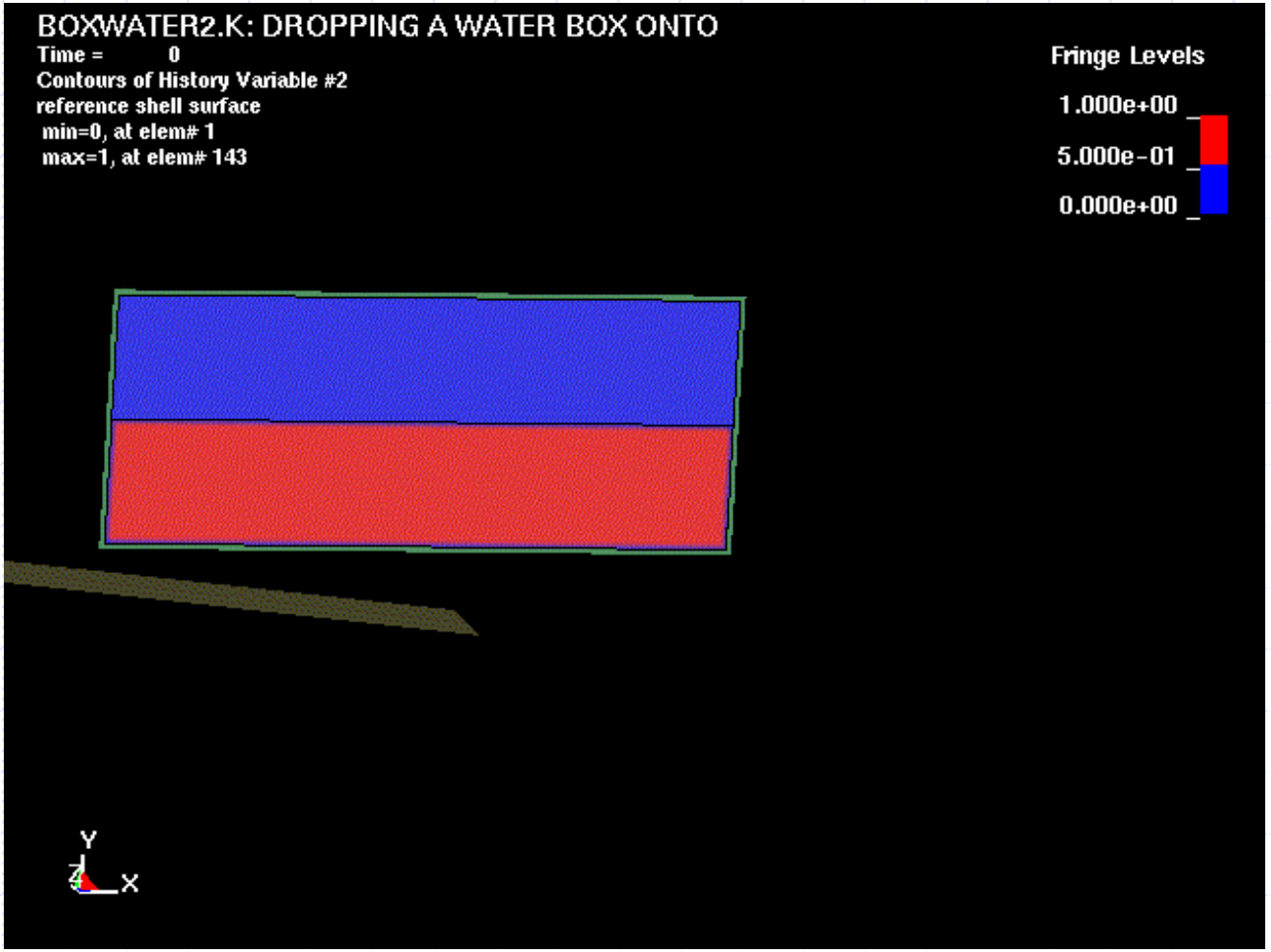
PROBLEM SET-UP



- A rigid box containing 2 fluids of different densities (one of the fluids is void) is dropped and hit the corner of a platform.
- The fluid system is assumed to follow the motion of the rigid tank. The motion of the rigid tank is defined by the 3 nodes attached to the body of the tank.
- The consequent motion results in the sloshing of the fluid inside the tank.

Dropping Of A Fluid Box

RESULTS (double-click on the figure below)

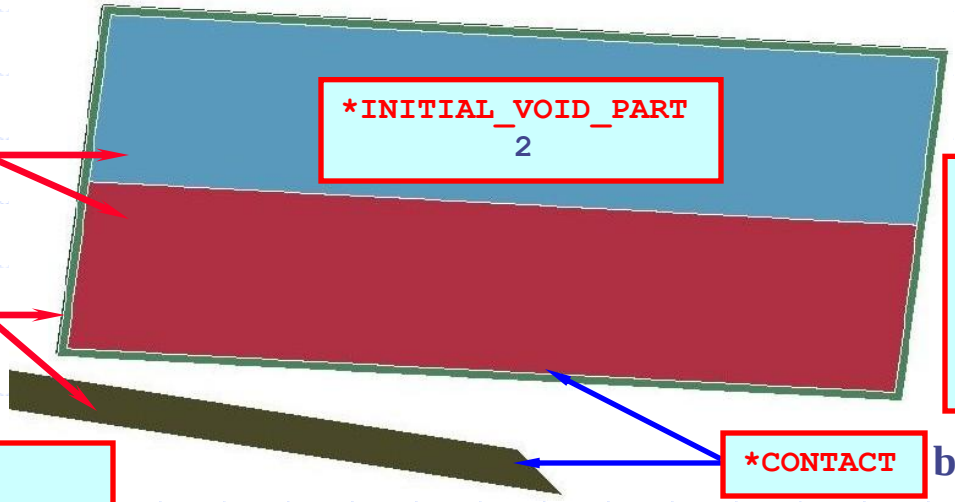


Dropping Of A Fluid Box

MODEL ORGANIZATION, INITIAL & BOUNDARY CONDITIONS

*SET_PART_LIST
PSID=1:parts 1&2

*SET_PART_LIST
PSID=2:parts 3&4



```
*INITIAL_VOID_PART
         2
```

```
*LOAD_BODY_Y
         1       1.00
*DEFINE_CURVE
1
0.0,981.0
1.0,981.0
```

*CONTACT between Lagrangian parts

```
*SET_PART_LIST
         1
         1         2
*SET_PART_LIST
         2
         3         4
```

```
*INITIAL_VELOCITY
$     NSID    NSIDEX     BOXID
         0
$       VX        VY        VZ       VXR       VYR       VZR
      0.0     -20.0       0.0
```

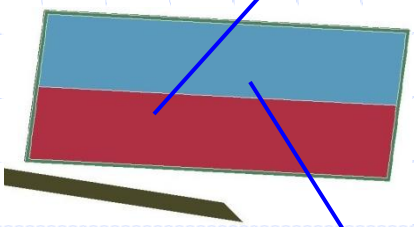
All nodes in the system

- PSID 1 = Eulerian parts and PSID 2 = Lagrangian parts.
- When NSID=0 in the *INITIAL_VELOCITY card, LS-DYNA imposes a velocity on all nodes. However, the *NODE constraints will override this velocity condition.
- *LOAD_BODY_Y imposes gravitational field on the system.

Dropping Of A Fluid Box

MODEL CONSTRUCTION

Defining the fluid and void parts. The fluid EOS is defined with a constant bulk modulus, $K=C_1$ below.

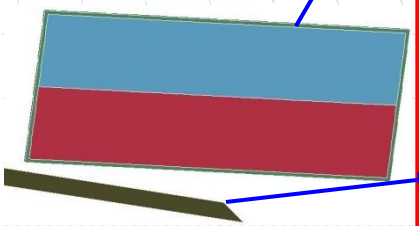


```
*PART
water in the box
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         1         0         0         0         0
*SECTION_SOLID_ALE
$    SECID    ELFORM       AET
         1        12
$     AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
$---------------------------------------------------------------------------
*MAT_NULL
$      MID       RHO        PC        MU     TEROD     CEROD        YM        PR
         1 1000.0000 -1.000+10 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
*EOS_LINEAR_POLYNOMIAL
$    EOSID        C0        C1        C2        C3        C4        C5        C6
         1 0.0000000 1.50000+9 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
$       E0        V0
 0.0000000 1.0000000
$===========================================================================
*PART
void portion in the box
         2         1         1         1         0         0         0         0
*INITIAL_VOID_PART
         2
```

Dropping Of A Fluid Box

MODEL CONSTRUCTION (cont.)

Defining the moving rigid box and fixed "very heavy" platform.



```
*PART
rigid box containing water
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         3         3         3         0         0         0         0         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         3         0
*MAT_RIGID
$      MID       RHO         E   POISSON         N    COUPLE         M     ALIAS
         3 2000.0000 1.00000+8 0.0000000 0.0000000 0.0000000 0.0000000
 0.0000000 0.0000000 0.0000000
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
$=============================================================================
*PART
rigid super-heavy platform
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         4         4         4
*SECTION_SHELL
$      SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP
         4         0
$       T1        T2        T3        T4      NLOC
     0.011     0.011     0.011     0.011
*MAT_ELASTIC
$      MID       RHO         E        PR        DA        DB         K
         4 1000000.0 1.0000+14
```

Artificially very heavy platform

Dropping Of A Fluid Box - *ALE_ Commands

In the *ALE_REFERENCE_SYSTEM_GROUP card, PRTYP=5 makes PSID 1 automatically follow the motion of a reference system, which is defined by 3 user-defined nodes, specified by the *ALE_REFERENCE_SYSTEM_NODE card. PRID=1 specifies the NSID of the *ALE_REFERENCE_SYSTEM_NODE card to be used.

```
*ALE_REFERENCE_SYSTEM_GROUP
$       SID     STYPE     PRTYP      PRID    BCTRAN      BCEXP      BCROT     ICOORD
PSID 1    1         0         5         1
$        XC        YC        ZC    EXPLIM

*SET_PART_LIST
         1
         1         2
*ALE_REFERENCE_SYSTEM_NODE
$      NSID
         1
$        N1        N2        N3        N4        N5        N6        N7        N8
         5         6         7
```

NSID of the 3 reference nodes

node-ID's of the 3 reference nodes

This forces the meshes of PID 1 & 2 to move with 3 nodes on the rigid container. *The fluid meshes of PID 1 & 2 have their boundary nodes merged with the box. So there is no need to define the *CONSTRAINED_LARANGE_IN_SOLID card here.*

Dropping of a Barrel Containing Fluid

3D: Dropping Of A Barrel containing Fluid

PHYSICAL SET-UP

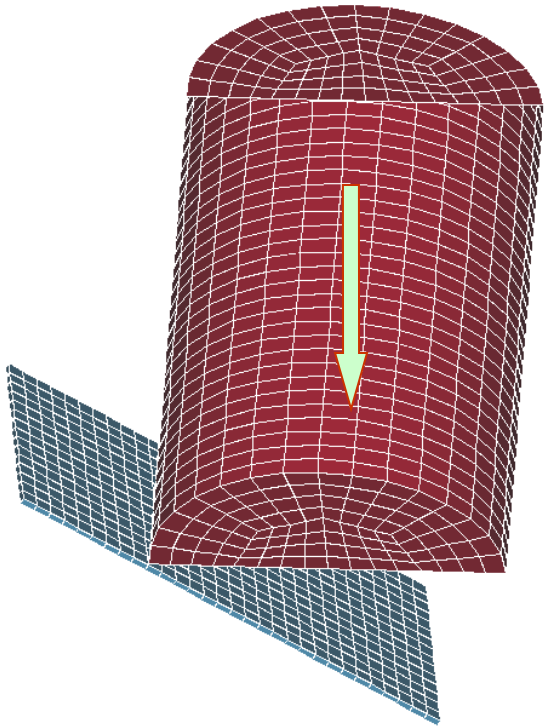
PID 1 = Shell structure = fluid container.
PID 3 = Rigid solid platform.
PID 10 = The fluid inside the container.
PID 11 = The void outside the container.

This model simulates a dropping of a container filled with water onto a rigid platform.

The fluids set-up can be done by

- define PID 10 & 11 with PID 11 as void; or by
- define PID 10 then use
  *INITIAL_VOLUME_FRACTION card to fill up the appropriate elements inside the container.

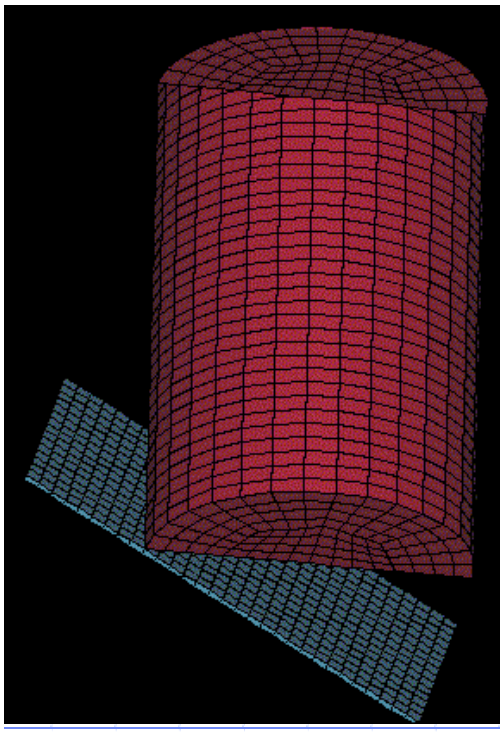
The fluid, PID 10, is given an initial downward velocity. FSI "pulls" PID 1 down.



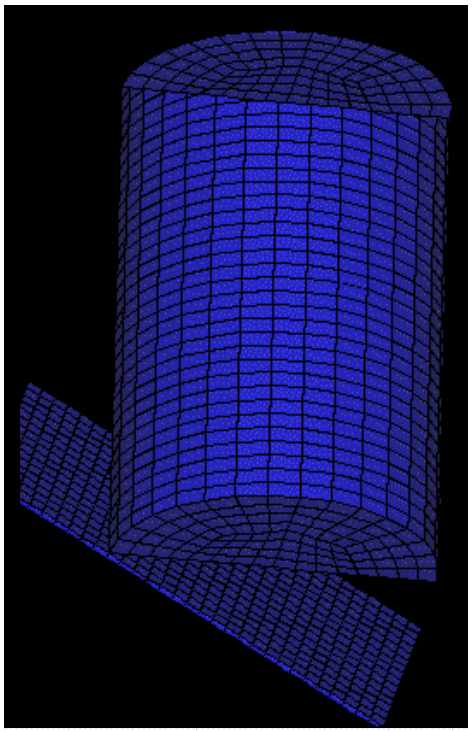
3D: Dropping Of A Barrel containing Fluid

RESULTS(double-click on the figures below)

Barrel deformation

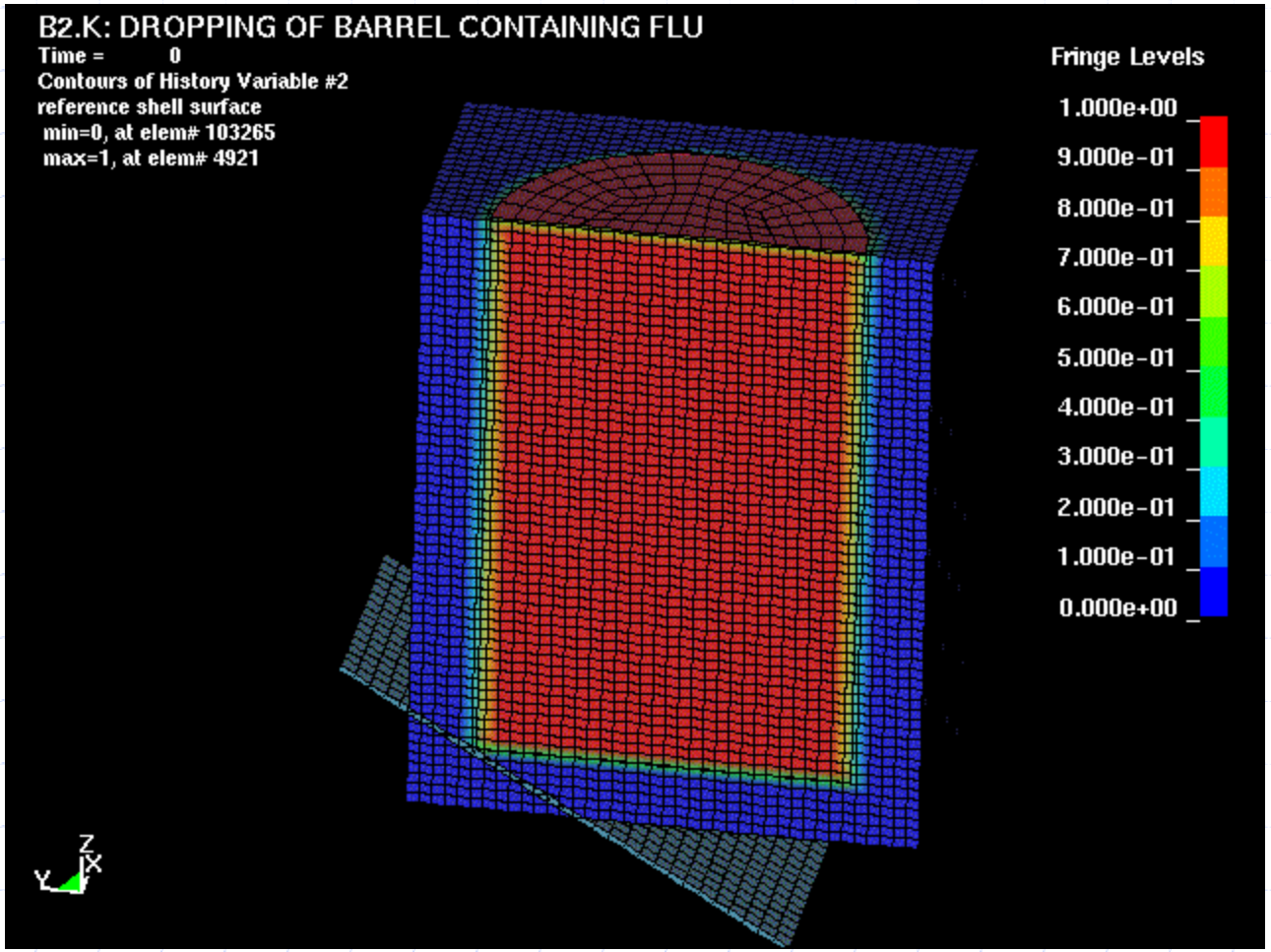


Barrel deformation & stress



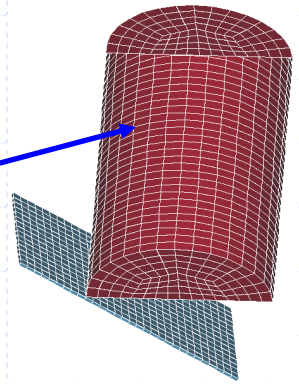
3D: Dropping Of A Barrel containing Fluid

RESULTS(double-click on the figures below) *Barrel deformation with fluid contour.*



3D: Dropping Of A Barrel containing Fluid

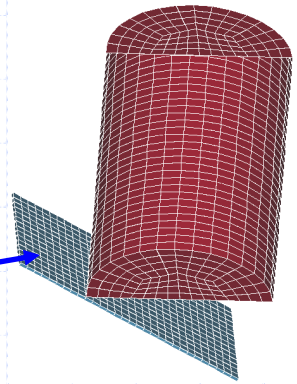
PART DEFINITION: Fluid container.



```
*PART
Deformable shell structure = Barrel or fluid container
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         1         0         0         0
*SECTION_SHELL
         1         7 0.0000000 0.0000000 0.0000000 0.0000000         0
 1.0000000 1.0000000 1.0000000 1.0000000 0.0000000
$ estimations:  ETAN=E/1000;    BETA=0=KINEMATIC;  BETA=1=ISOTROPIC HARDENING
*MAT_PLASTIC_KINEMATIC
$      MID        RO         E        PR      SIGY      ETAN      BETA
         1 7.8400E-6 2.0680E+8       0.3     1.0e6     2.0e5       0.0
$      SRC       SRP        FS        VP
       0.0       0.0       0.0
*HOURGLASS
         1         4 0.0000000         0 0.0000000 0.0000000
```

3D: Dropping Of A Barrel containing Fluid

PART DEFINITION: Rigid platform.

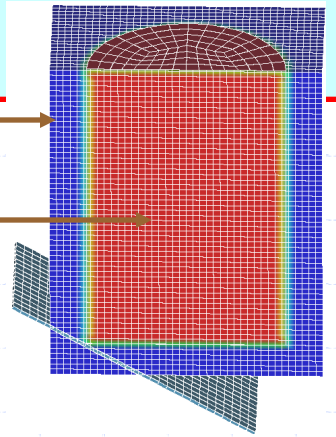


```
*PART
rigid plaform (1000x as heavy, 20x as hard comparing to shell/barrel)
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         3         3         3         0         0         0         0         0
*SECTION_SOLID
         3         0
*MAT_RIGID
$      MID        RO         E        PR         N    COUPLE         M
         3 7.8400E-6 2.0680E+8       0.3       0.0 0.0000000 0.0000000
$      CMO      CON1      CON2
       1.0       7.0       7.0
$       A1        A2        A3        V1        V2        V3
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
```

3D: Dropping Of A Barrel containing Fluid

PART DEFINITION: Water inside container and void outside.

```
*PART
water = fluid inside barrel
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
        10         2         2         2         0         0         0         0
*SECTION_SOLID
         2        12
*MAT_NULL
$      MID       RHO        PC        MU     TEROD     CEROD        YM        PR
         2    1.0E-6   -1.0E+1    8.7E-7 0.0000000 0.0000000 0.0000000
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMMA         A        E0
         2   1.65E+6  1.920000  .0000000  .0000000  0.100000  .0000000       0.0
$       V0
  .0000000
$---------------------------------------------------------------------------
*PART
void mesh outside barrel
        11         2         2         2         0         0         0         0
*INITIAL_VOID_PART
        11
```



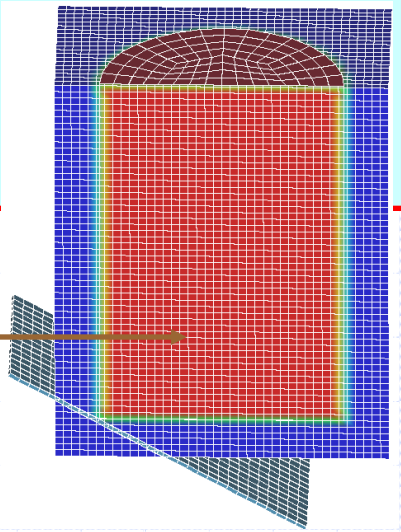
3D: Dropping Of A Barrel containing Fluid

INITIAL & CONTACT CONDITIONS:

```
$ Approximately Vz~40km/hr=11200mm/s ;  Vy~0.0km/hr=    0mm/s
*INITIAL_VELOCITY_GENERATION
$       ID       STYP      OMEGA         VX         VY         VZ
        10          2          0        0.0        0.0   -11200.0
$       XC         YC         ZC         NX         NY         NZ      PHASE
         0          0          0          0          0          0          0
$=========================================================================
$ MSID=0=single surface; SSTYP=2=PSID:PID 1=shell barrel, PID 3=rigid platform
*CONTACT_AUTOMATIC_SINGLE_SURFACE
$      SSID       MSID      SSTYP      MSTYP     SBOXID     MBOXID        SPR        MPR
         1          0          2          0          0          0          0          0
$        FS         FD         DC         VC        VDC     PENCHK         BT         DT
 0.0000000  0.0000000  0.0000000  0.0000000  0.0000000          0  0.0000000  0.0000000
$       SFS        SFM        SST        MST       SFST       SFMT        FSF        VSF
 00.000000  00.000000  0.0000000  0.0000000  0.0000000  0.0000000
$-------------------------------------------------------------
*SET_PART_LIST
         1
         1          3
```

MSID=0=single-surface contact

*INITIAL_VELOCITY_GENERATION → Gives PID 10=fluid, its initial velocity



3D: Dropping Of A Barrel containing Fluid

ALE SET-UP:

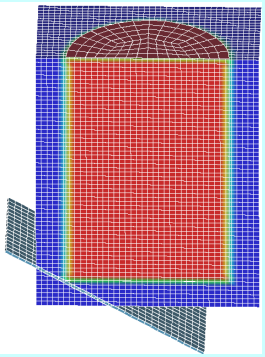
```
*CONTROL_ALE
$      DCT      NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
         2         1         2 -1.000000 0.0000000 0.0000000 0.0000000 0.0000000
$    START       END     AAFAC     VFACT    VLIMIT       EBC
 0.0000000 0.0000000 0.0000000     2.e-6
$---------------------------------------------------------------------------
$ PID=10=fluid inside barrel==>PRTYP=5=mesh follow 3 nodes using PRID=1
$ BCTRAN=1=fixed x; BCEXP=7=no mesh expansion; BCROT=5=no rot about y&z
*ALE_REFERENCE_SYSTEM_GROUP
$      SID     STYPE     PRTYP     PRID   BCTRAN    BCEXP    BCROT   ICOORD
        10         1         5         1        1        7        5
$       XC        YC        ZC   EXPLIM

*ALE_REFERENCE_SYSTEM_NODE
         1
    100122    100202    100617
$---------------------------------------------------------------------------
$ coupling PID 1 to PID 10
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
         1        10         1         1         4         4         2
$    START       END      PFAC      FRIC    FRCMIN      NORM
         0         0         0         0         0         0
$       CQ      HMIN      HMAX     ILEAK
         0         0         0         0
```

PID 10

NSID of the 3 reference nodes

node-ID's of the 3 reference nodes

←Interaction between PID 1 (shell) & PID 10 (fluid)



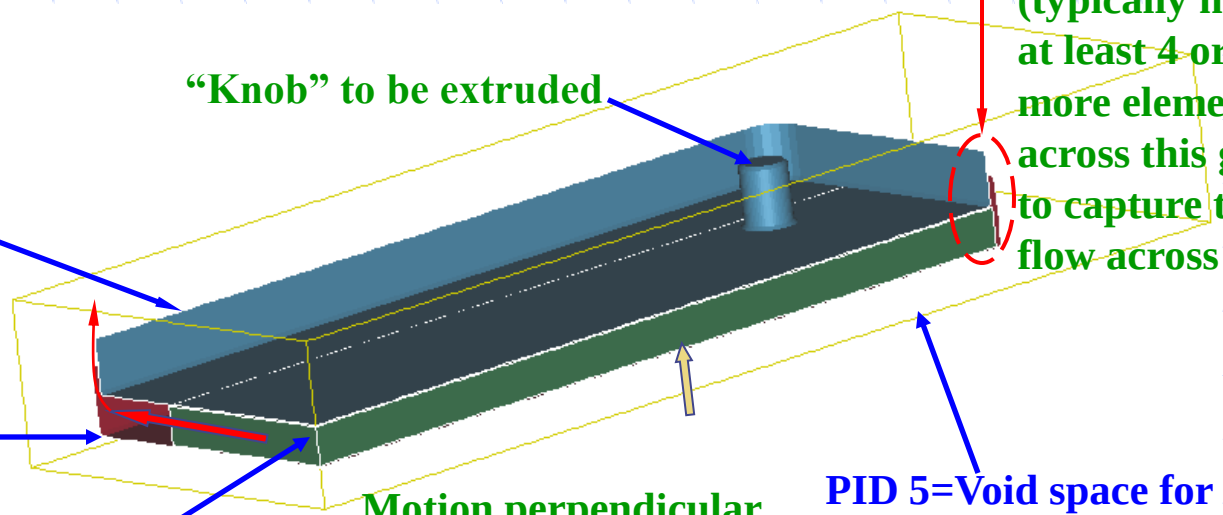
Extrusion of a Plastic Panel For a Cell Phone (Courtesy of Scientific Materials INC.)

Simple Extrusion of a Plastic Pannel

PHYSICAL SET-UP

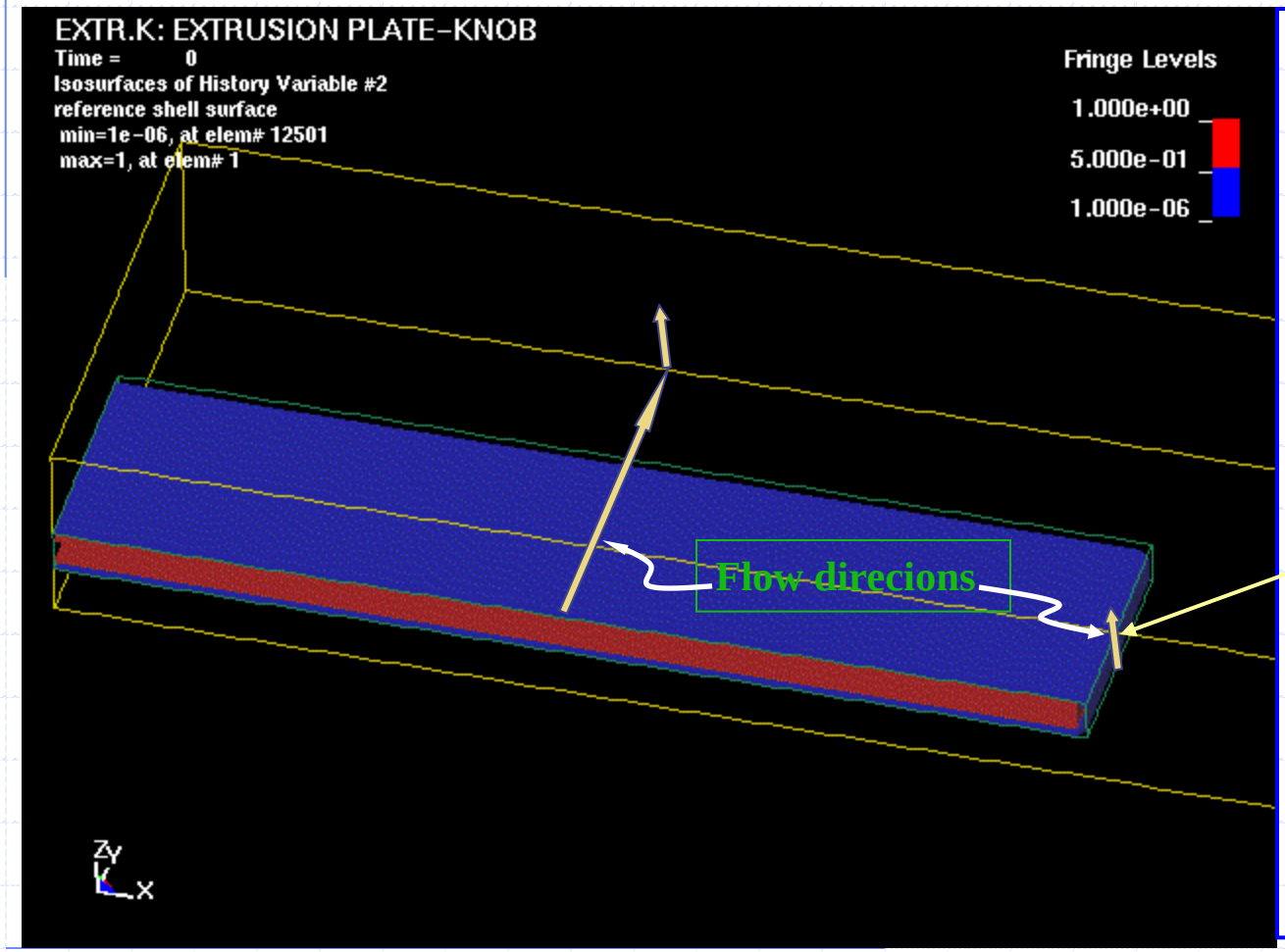
The lower shell motion squeezes the work piece material into the "knob" cavity of the upper shell.

The calculation can be expensive due to the small mesh required to resolve the small vertical gaps between the 2 shell structures.



Simple Extrusion of a Plastic Pannel

RESULT: (double-click on the picture below)



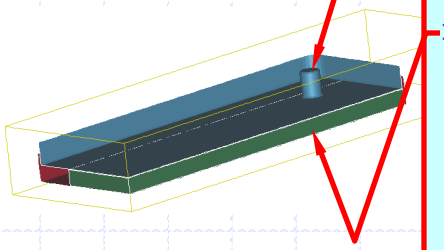
As the bottom platform moves up ward the material is squeezed into the gaps.

Note that on the small end gap the material seems discontinuous. This may be a result of the void mesh being too coarse there.

More elements will be needed to resolve the fluid interface accurately.

Simple Extrusion of a Plastic Pannel

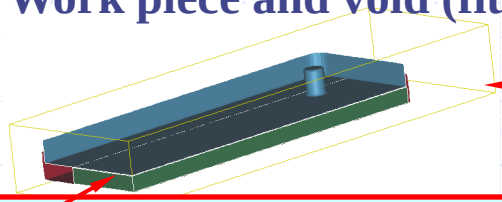
PART DEFINITIONS: Top & bottom tool pieces



```
*PART
top die fixed
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         3         3         3         0         0         0         0         0
*SECTION_SHELL
         3         0 0.0000000 0.0000000 0.0000000 0.0000000         0
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
*MAT_RIGID
         3 7850.0000 2.1000+10 0.3000000 0.0000000 0.0000000 0.0000000
 0.0000000 0.0000000 0.0000000
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
$---------------------------------------------------------------------------
*PART
bottom punch moving up
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         2         2         2         0         0         0         0         0
*SECTION_SHELL
$      SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP
         2         0 0.0000000 0.0000000 0.0000000 0.0000000         0
$       T1        T2        T3        T4      NLOC
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
*MAT_RIGID
$      MID        RO         E        PR         N    COUPLE         M
         2 7850.0000 2.1000+10 0.3000000 0.0000000 0.0000000 0.0000000
$      CMO      CON1      CON2
 0.0000000 0.0000000 0.0000000
$       A1        A2        A3        V1        V2        V3
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
```

Simple Extrusion of a Plastic Pannel

PART DEFINITIONS: Work piece and void (fluid/master parts)



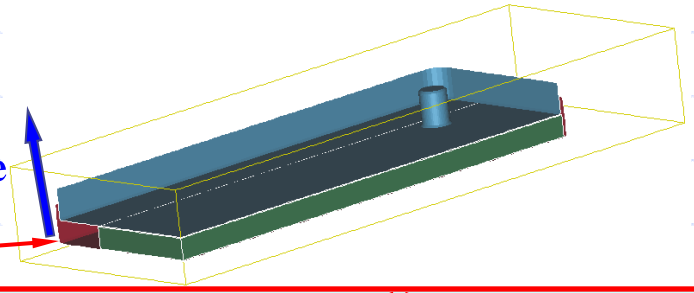
PID 5=Void space for material to flow into

```
*PART
workpiece
$      PID      SECID        MID      EOSID       HGID       GRAV     ADPOPT       TMID
         4          4          4          0          0          0          0          0
*SECTION_SOLID_ALE
$    SECID     ELFORM        AET
         4         12
$     AFAC       BFAC       CFAC       DFAC      START        END      AAFAC
  .0000000   .0000000   .0000000   .0000000   .0000000   .0000000
*MAT_PLASTIC_KINEMATIC
$      MID        RO         E         PR       SIGY       ETAN       BETA
         4 10000.000 3.50000+8 0.3000000 1.00000+5 1.00000+5 1.0000000
$      SRC        SRP        FS        VP
 0.0000000 0.0000000 0.0000000
$---------------------------------------------------------------------------
*PART
void
$      PID      SECID        MID      EOSID       HGID       GRAV     ADPOPT       TMID
         5          5          4          0          0          0          0          0
*INITIAL_VOID_PART
         5
*SECTION_SOLID_ALE
         5         12
  .0000000   .0000000   .0000000   .0000000   .0000000   .0000000
```

Simple Extrusion of a Plastic Pannel

IC's & BC's:

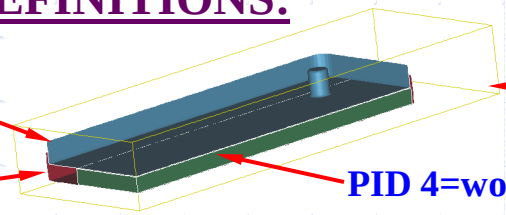
PID 2=bottom tool piece (moves up)



```
$ PID 2 moves: DOF=4=moves along a vector ID=#1 ; VAD=2=displacement ; using LCID=1
*BOUNDARY_PRESCRIBED_MOTION_RIGID
$NID|NSID|PID     DOF       VAD      LCID        SF       VID     DEATH     BIRTH
         2         4         2         1 1.00000-0         1 2.0000000 0.0000000
$ Curve defining displacement VS. time.
*DEFINE_CURVE
$     LCID      SIDR       SFA       SFO      OFFA      OFFO    DATTYP
         1         0 0.0000000 0.0000000 0.0000000 0.0000000
      0.00000000E+00      0.00000000E+00
      0.01000000E-00      0.00290000E+00
$ Vector defining the direction along which the motion is to take place.
*DEFINE_VECTOR
$      VID        XT        YT        ZT        XH        YH        ZH
         1 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000
```

Simple Extrusion of a Plastic Pannel

ALE CONSTRAINT DEFINITIONS:




```
*CONTROL_ALE
$      DCT       NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
         2          1         2   -1.0000 0.0000000  .0000000 0.0000000
$    START       END     AAFAC     VFACT    VLIMIT       EBC
  .0000000  .0000000  .0000000     1.e-5
$ constraint fluids (work-piece & void =PSID4=PID 4 & 5) with bottom tool piece
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
         2         4         1         0         3         4         1         1
$     START       END      PFAC      FRIC   FRICMIN      NORM
         0         0                                      1
$ constraint fluids (work-piece & void =PSID4=PID 4 & 5) with top tool piece
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE    MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
         3         4         1         0         3         4         1         1
$     START       END      PFAC      FRIC   FRICMIN      NORM
         0         0                                      0
*SET_PART_LIST
         4
         4         5
```

PID 2=bottom piece

PID 3=top piece

Slave=PID

Master=PSID

Couple to Lagrangian shell

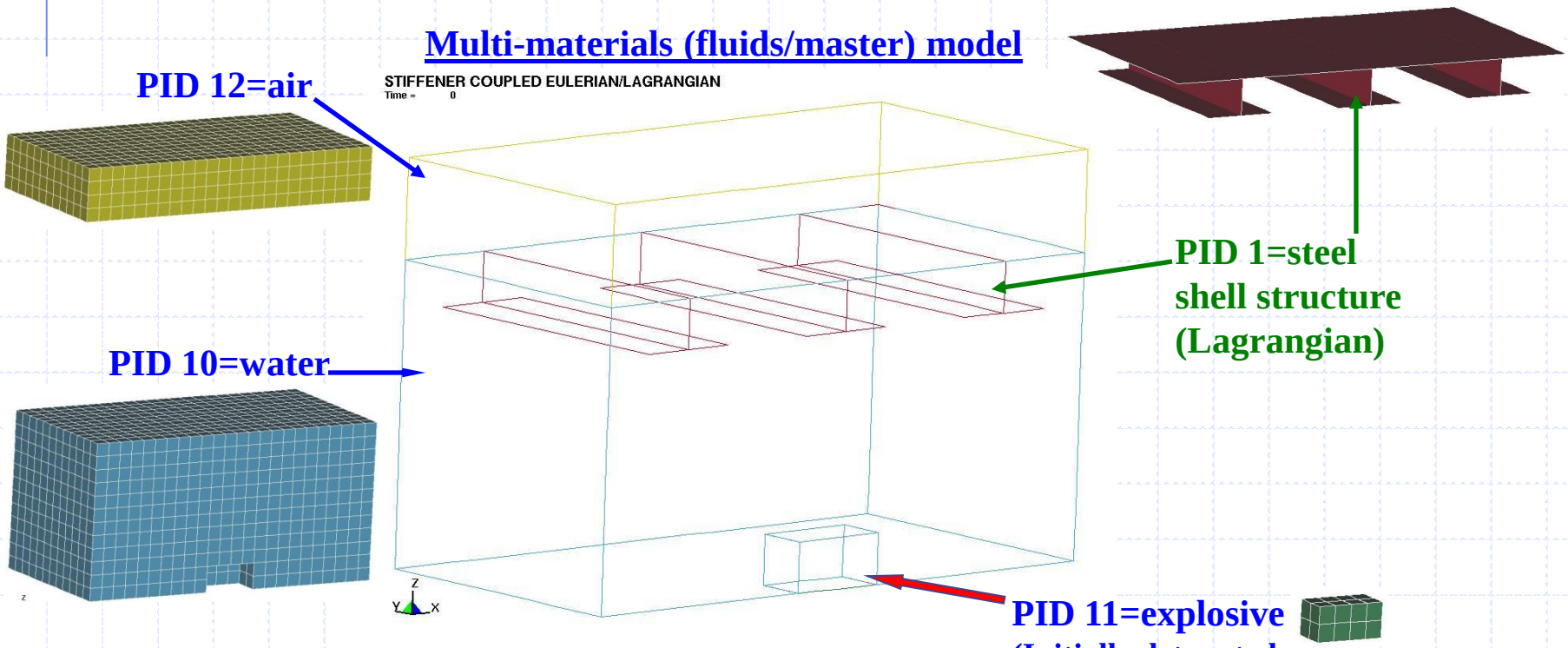
Under Water Detonation Effects on a Shell structure

Under Water Explosion Effects on a Shell Structure

PHYSICAL SET-UP

- A non-symmetrical model having an explosive exploding under water and sending pressure waves to interact with a steel shell structure.
- The **Lagrangian shell structure** is not constrained in the vertical (z) direction.
- All "fluid meshes" have merged nodes at their boundary interfaces.

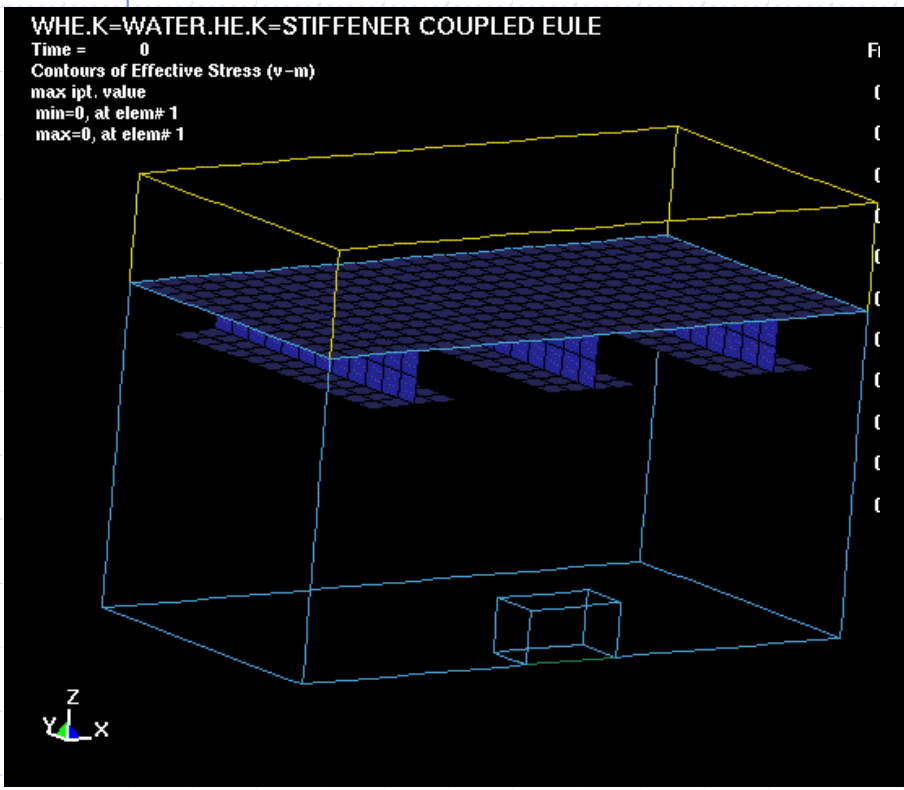
Multi-materials (fluids/master) model



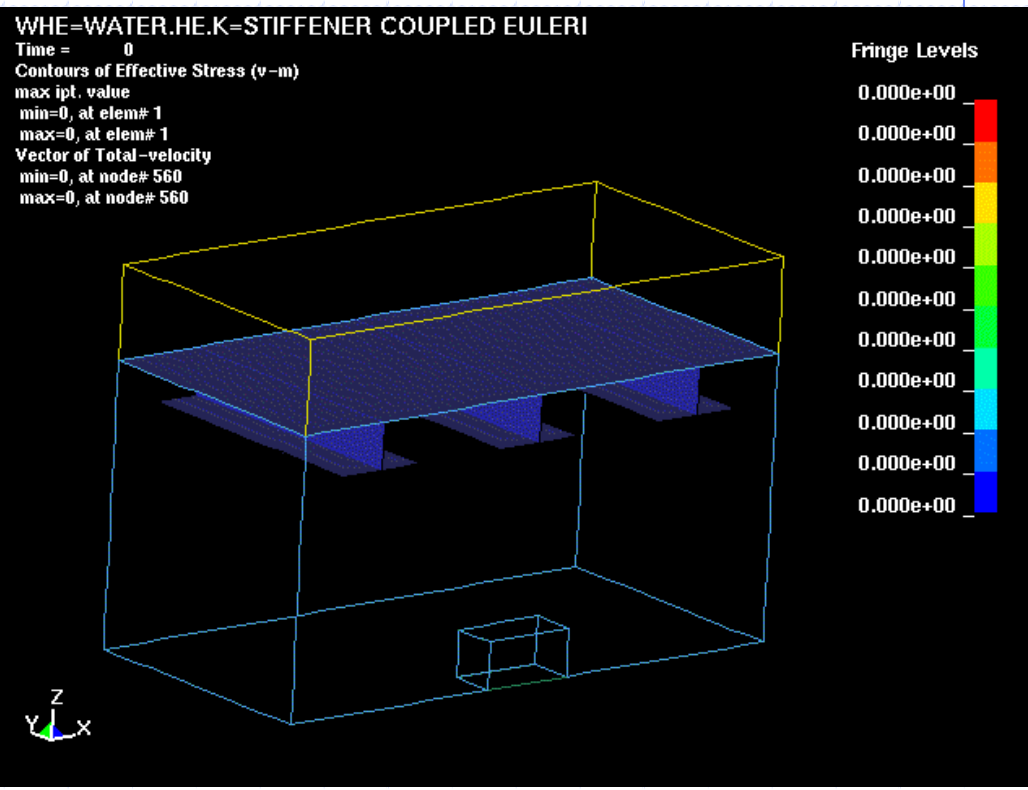
Under Water Explosion Effects on a Shell Structure

RESULTS(double-click on the figures below)

Von Mises stress on shell structure

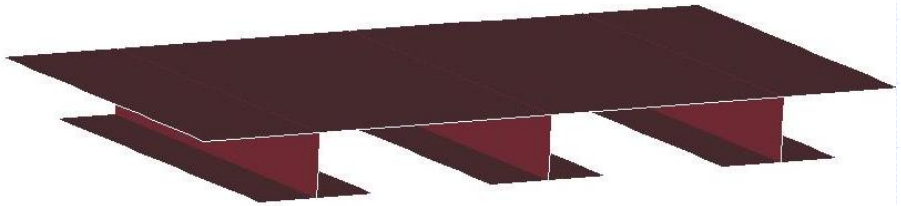


Von Mises stress on shell structure & vel vector



Under Water Explosion Effects on a Shell Structure

PART DEFINITIONS: Shell structure = PID 1

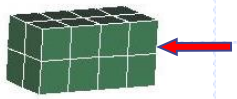


Thickness →

```
*PART
Slave metal structure : PID 1: mat # 3 = Kinematic/Isotropic Elastic-Plastic
$       PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
          1         1         1         0         1         0
*SECTION_SHELL
$       SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP
          1         2 0.000E+00      3.00 0.000E+00 0.000E+00         0
$        T1        T2        T3        T4      NLOC
     0.200     0.200     0.200     0.200 0.000E+00
$---------------------------------------------------------------------------
*MAT_PLASTIC_KINEMATIC
$       MID        RO         E        PR      SIGY      ETAN      BETA
          1      7.83      2.07     0.300 8.000E-03 0.000E+00 0.000E+00
$       SRC       SRP        FS        VP
 0.000E+00 0.000E+00 0.000E+00
*HOURGLASS
$      HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
          1         0 0.000E+00         0 0.000E+00 0.000E+00
```

Under Water Explosion Effects on a Shell Structure

PART DEFINITIONS: High explosive = PID 11



PID 11=explosive (Initially detonated at time 0)

JWL EOS:

$$p = A\left[1 - \frac{\omega}{R_1 V}\right]e^{-R_1 V} + B\left[1 - \frac{\omega}{R_2 V}\right]e^{-R_2 V} + \frac{\omega \cdot E}{V}$$

```
*PART
High Explosive: PID 11 : mat # 8 = High Explosive Burn
$       PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         11        11        11        11        11         0
*SECTION_SOLID_ALE
$     SECID    ELFORM       AET                                               Multi-material
         11        11         0
$      AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
          0
$---------------------------------------------------------------------------------
*MAT_HIGH_EXPLOSIVE_BURN                                    Chapman-Joguet
$       MID        RO         D       PCJ      BETA         K         G      SIGY
         11      1.63     0.784     0.260 0.000E+00
*EOS_JWL
$     EOSID         A         B        R1        R2      OMEG        E0        V0
         11      3.71 3.230E-02      4.15     0.950     0.300 4.300E-02      1.00
*HOURGLASS
$      HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
         11         0 0.000E+00         0 0.000E+00 0.000E+00
$---------------------------------------------------------------------------------
*INITIAL_DETONATION
$       PID         X         Y         Z        LT
         11 0.000E+00 0.000E+00 0.000E+00 0.000E+00
```

Det-Vel

Pressure

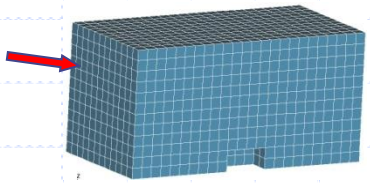
Detonate PID 11

Ignition time

Detonation point

Under Water Explosion Effects on a Shell Structure

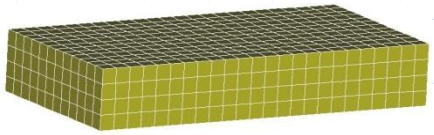
PART DEFINITIONS: Water=PID 10



```
*PART
Water: PID 10
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
        10        10        10        10        10         0
*SECTION_SOLID_ALE
$    SECID    ELFORM       AET
        10        11         0
$     AFAC      BFAC      CFAC      DFAC     START       END     AAFAC
         0
$---------------------------------------------------------------------------
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
        10      1.00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMMA         A        E0
        10     0.148      1.75 0.000E+00 0.000E+00     0.280 0.000E+00 0.000E+00
$       V0
      1.00
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
        10         0 0.000E+00         0 0.000E+00 0.000E+00
```

Under Water Explosion Effects on a Shell Structure

PART DEFINITIONS: Air=PID 12



Perfect gas is assumed for air.

```
*PART
Air: PID 12 = mat # 9
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
        12        12        12        12        12         0
*SECTION_SOLID
$    SECID    ELFORM       AET
        12        11         0
$---------------------------------------------------------------------------
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
        12 1.280E-03 0.000E+00 0.000E+00 0.000E+00 0.000E+00
*EOS_LINEAR_POLYNOMIAL
$    EOSID        C0        C1        C2        C3        C4        C5        C6
        12 0.000E+00 0.000E+00 0.000E+00 0.000E+00     0.400     0.400 0.000E+00
$       E0        V0
 0.000E+00 0.000E+00
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
        12         0 0.000E+00         0 0.000E+00 0.000E+00
```

Under Water Explosion Effects on a Shell Structure

ALE CONSTRAINT DEFINITIONS:

van Leer + Half-Index-Shift

```
$ DCT=2=EUL; METH=4=DONOR+HIS
*CONTROL_ALE
$       DCT      NADV      METH      AFAC      BFAC      CFAC      DFAC      EFAC
          2         1         2        -1       0.0       0.0       0.0       0.0
$     START       END     AAFAC     VFACT    VLIMIT       EBC
        0.0       0.0       0.0       0.0       0.0       0.0
$---------------------------------------------------------------------------
*ALE_MULTI-MATERIAL_GROUP
10,1
11,1
12,1
$---------------------------------------------------------------------------
$SSTYP=0=PSID; MSTYP=0=PSID, CTYPE=4=PENALTY; DIREC=2=COMPRESSION; MCOUP=1=HI RO
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE     MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
          1          2         0         0         4         4         2         0
$     START       END      PFAC      FRIC    FRCMIN      NORM
        0.0       0.0       0.1       0.0       0.0         1
$        CQ      HMIN      HMAX     ILEAK
        0.0       0.0       0.0         1
$---------------------------------------------------------------------------
*SET_PART_LIST
1
1
*SET_PART_LIST
2
10,12
```

Provide interface tracking for multi-materials

Turn on leakage control

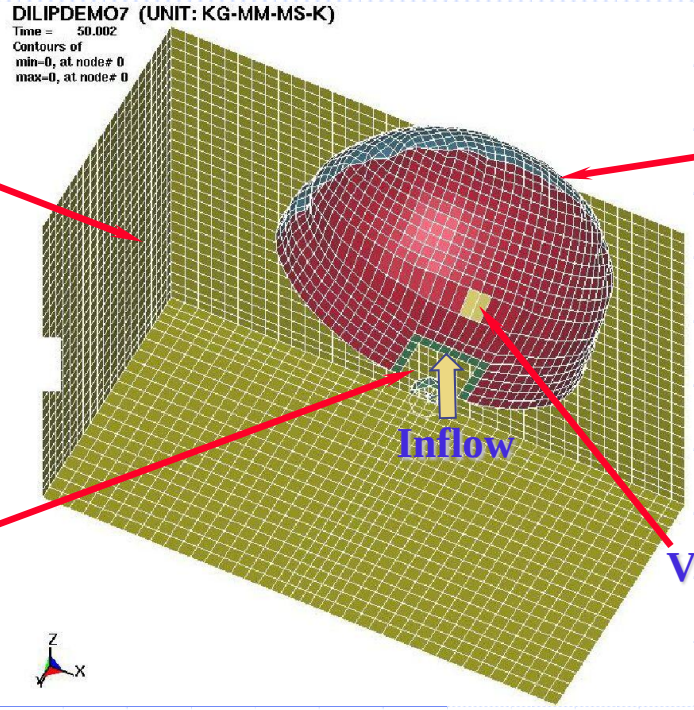
Flipping the directions of the shell normal vectors

Using Fluid-Structure-Interaction Approach to Model Airbag Inflation Process

Using ALE Coupling to Model Airbag Inflation Process

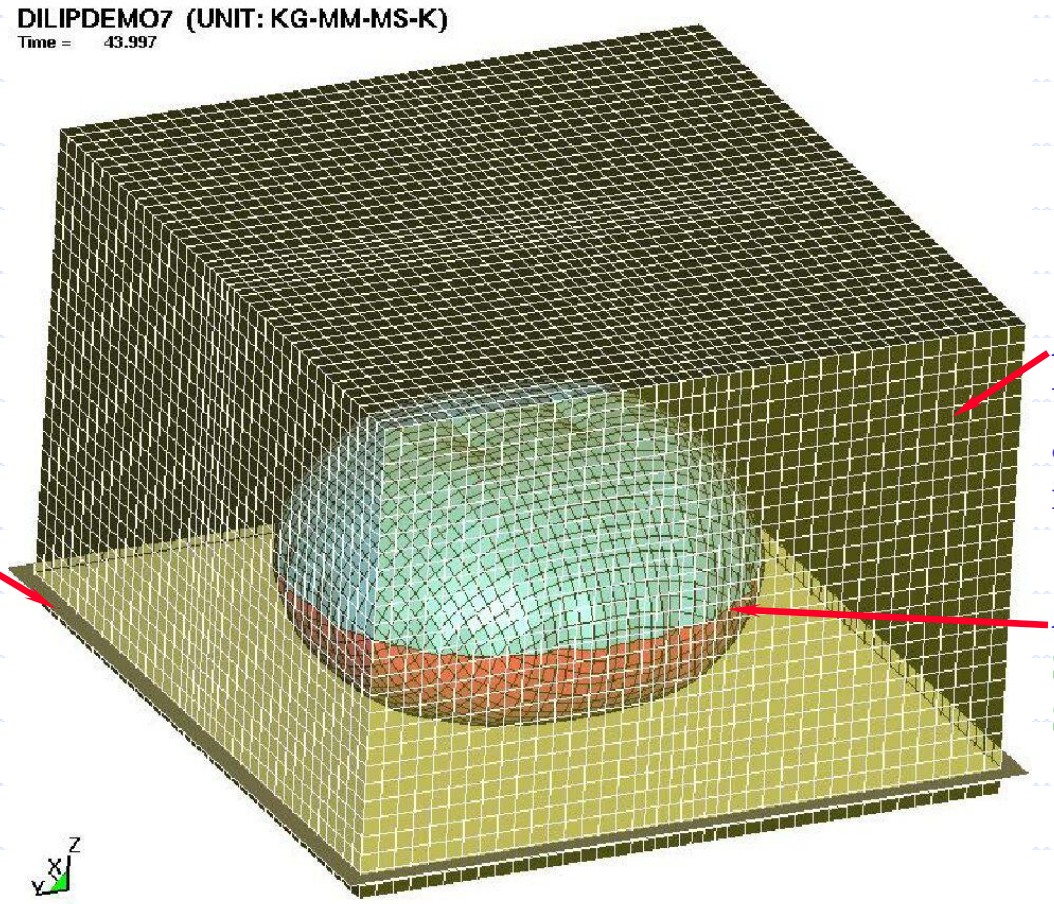
PHYSICAL SET-UP

- One **Eulerian solid element** mesh (H50000=air) define the fluid initial mesh.
- It defines a surrounding air mesh providing room for the bag to expand into.
- One **Eulerian part** (H50003=no initial mesh) = inflator gas source.
- The **Lagrangian shell structures** make up the airbag.
- A rigid Lagrangian platform to limit the motion of the bag.



Using ALE Coupling to Model Airbag Inflation Process

[Fluids=Air-Gas meshes] + [Structures = Airbag parts]:



Fixed backing platforms defined by *RIGIDWALL_PLANAR

Air mesh covering the space which the airbag will expand into. (H50000)

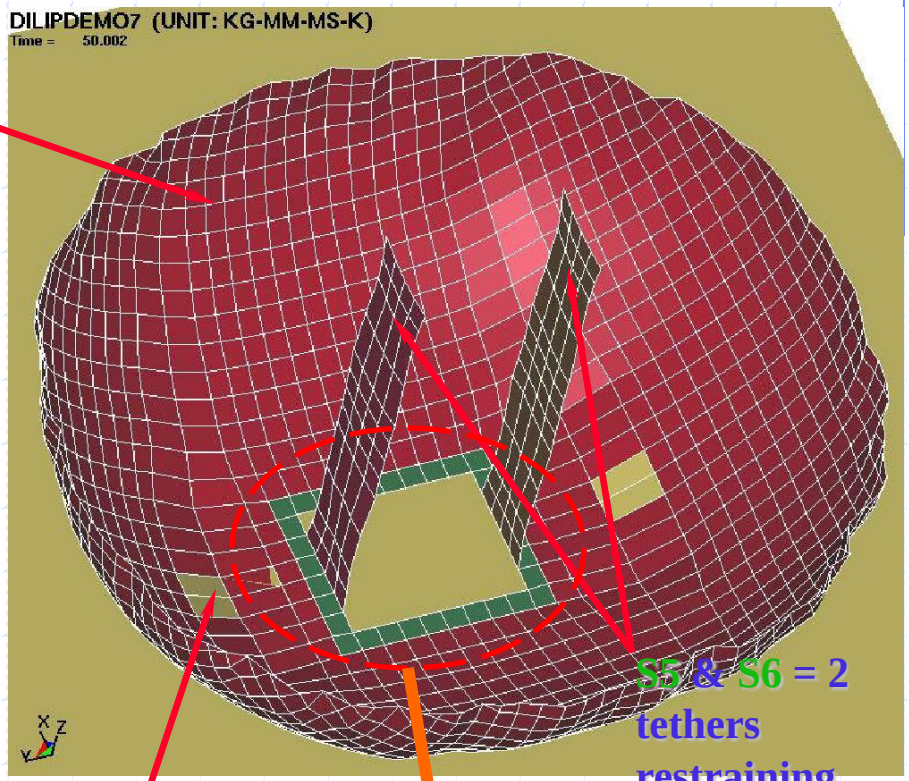
Airbag (PIDs: S1, S2, S5, S6, S9)

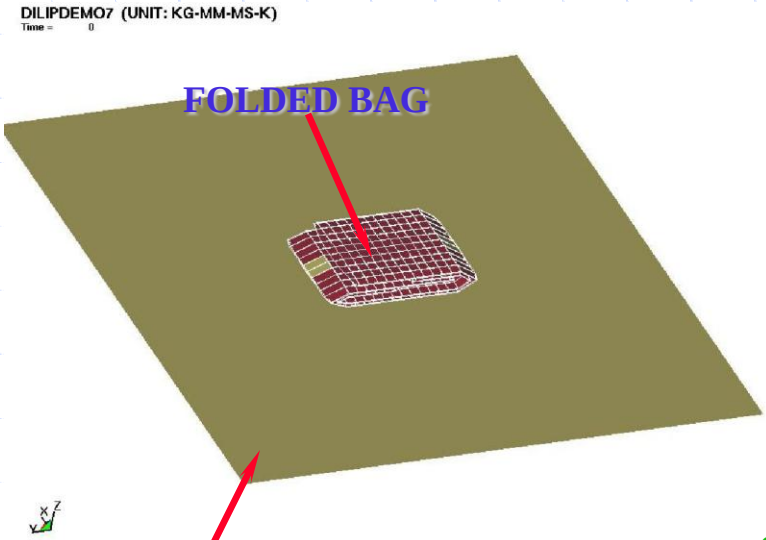
Blanking out the outside of the bag for internal view … ➔

Using ALE Coupling to Model Airbag Inflation Process

GEOMETRY:

Inflated airbag:
S1 is the main lower outer bag definition
(S2=top portion)





Backing platform
*RIGIDWALL_PLANAR

S9 = 2 vent patches

S5 & S6 = 2 tethers restraining the top portion of the airbag

Zoom-in in a later slide …

Using ALE Coupling to Model Airbag Inflation Process

AIRBAG STRUCTURE SET-UP

The Lagrangian shell parts making up the airbag structure consist of PID's: 1, 2, 5, 6, 9. For each of these parts, the definition consists of 3 cards typically looking like the following:

Shear correction factor

# of integration points

```
*PART
material type # 34  (fabric)
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0         0         0
*SECTION_SHELL
$      SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP
         1         5 0.0000000 4.0000000 0.0000000 0.0000000         1
$       T1        T2        T3        T4      NLOC
 4.00000-4 4.00000-4 4.00000-4 4.00000-4 0.0000000
$       B1        B2        B3        B4        B5        B6        B7        B8
      0.0       0.0       0.0       0.0
*MAT_FABRIC
$      MID        RO        EA        EB        EC      PRBA      PRCA      PRCB
         1 8.76000-7   .3000000  .2000000  .3000000  .2000000  .2000000  .2000000
$      GAB       GBC       GCA       CSE        EL       PRL    LRATIO      DAMP
  .0400000  .0400000  .0400000 1.0000000  .0600000  .3500000  .1000000  .2000000
$     AOPT       FLC       FAC       ELA      LNRC      FORM
 3.0000000
$                                     A1        A2        A3
  .0000000  .0000000  .0000000 1.0000000  .0000000  .0000000
$       V1        V2        V3        D1        D2        D3      BETA
 1.0000000  .0000000  .0000000  .0000000  .0000000  .0000000
```

Young modulus

Poisson ratio

Material angles at integration points

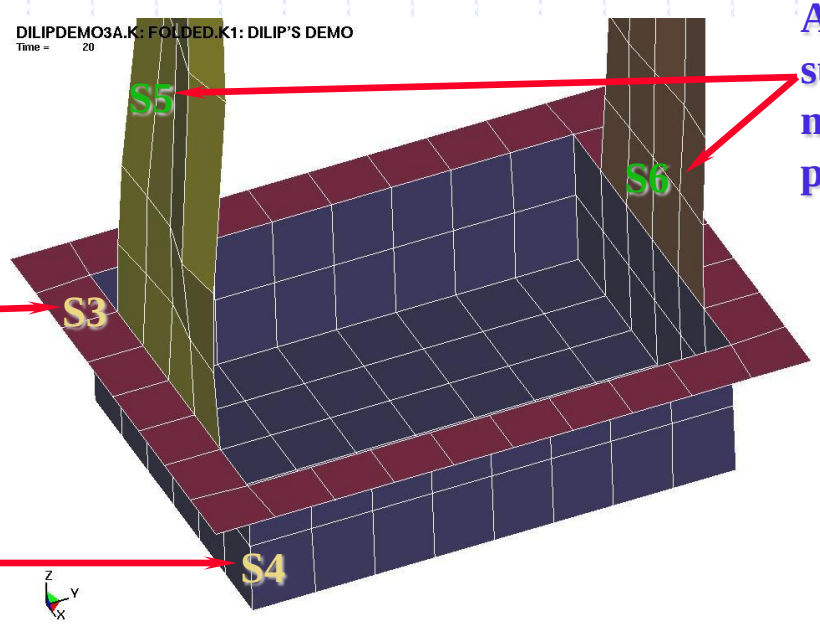
Shear modulus

Material Axes option

Vector for AOPT=3

Using ALE Coupling to Model Airbag Inflation Process

INTERNAL GEOMETRY



Airbag internal support structures made up of 2 shell parts.

Rigid inflator box rim (fixed)

Rigid inflator box containing priming gas block (fixed)

Contact among the Lagrangian shell structures:

***CONTACT_AUTOMATIC_SINGLE_SURFACE,**

Slave=*PSID 1*: PID's S1-S6, S9 ; Master=None (self-contact).

Fluid-Structure-Coupling (air+gas=master; all Lagrangian parts= slave):

***CONSTRAINED_LAGRANGE_IN_SOLID**

Slave=*PSID 1001*:{ PID's S1-S6, S9} ; Master= *PSID 1004*:{ PID's H50000 & H50003}

Using ALE Coupling to Model Airbag Inflation Process

INFLATOR BOX SET-UP

The inflator box is made up of 2 rigid material parts (S3 & S4) at the bottom of the airbag.

Shear correction factor

# of integration points

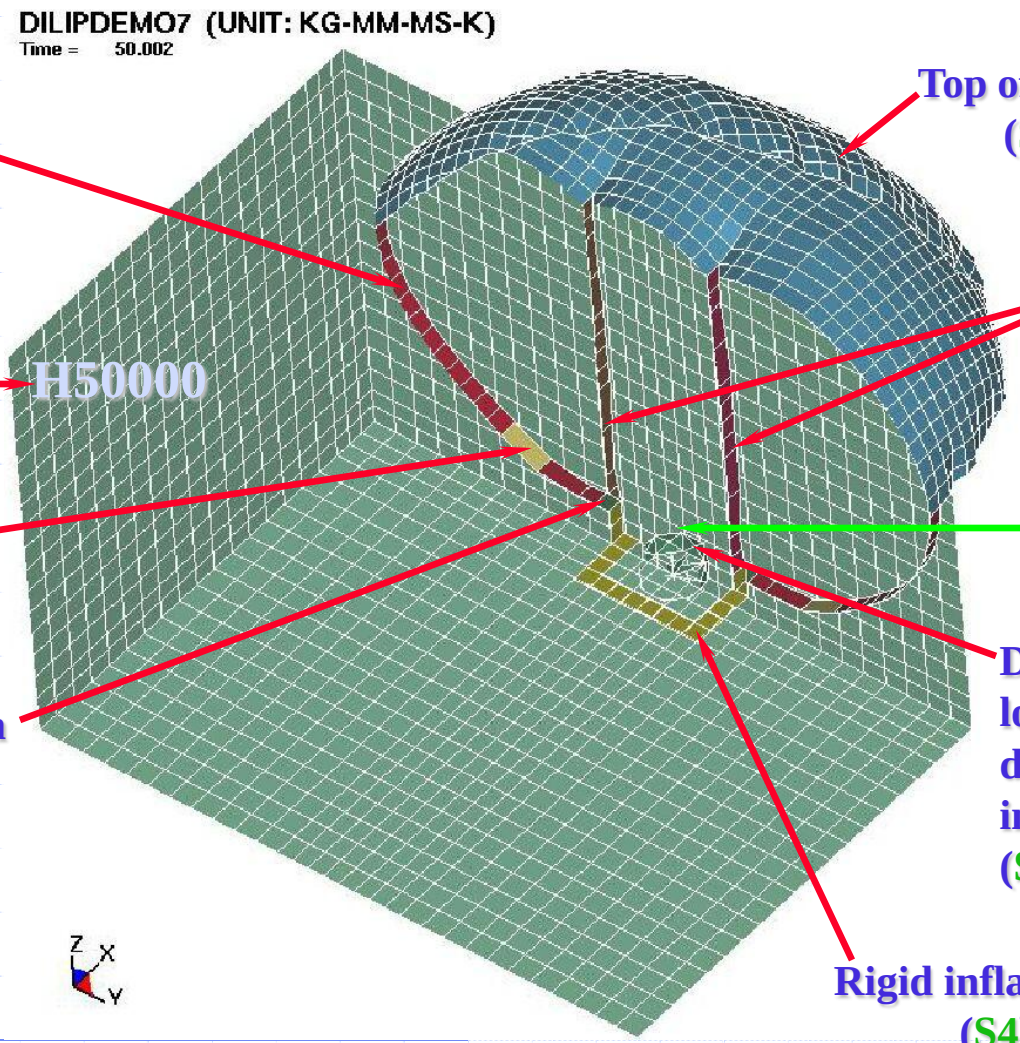
Shell thickness @ its 4 nodes

Center-of-mass constraint = This fixes the inflator box in the global coordinates.

```
$=========================================================================
*PART
rigid inflator box rim    : thick = 0.4 mm
         3         3         3         0         0         0         0         0
*SECTION_SHELL
$       SID    ELFORM      SHRF       NIP     PROPT   QR/IRID     ICOMP
         3         5 0.0000000 4.0000000 0.0000000 0.0000000         1
$        T1        T2        T3        T4      NLOC
 0.4000000 0.4000000 0.4000000 0.4000000 0.0000000
$        B1        B2        B3        B4        B5        B6        B7        B8
       0.0       0.0       0.0       0.0
*MAT_RIGID
$       MID        RO         E        PR         N    COUPLE         M
         3 7.8500-06 2.0000000 0.3000000 0.0000000 0.0000000 0.0000000
$       CMO      CON1      CON2
 1.0000000 7.0000000 7.0000000
$        A1        A2        A3        V1        V2        V3
 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
$=========================================================================
*PART
rigid inflator box side walls and bottom: thick = 0.4 mm
$       PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         4         3         3         0         0         0         0         0
$=========================================================================
```

Using ALE Coupling to Model Airbag Inflation Process

BOTTOM VIEW:



Bottom of airbag (S1)

Top of airbag (S2)

Tethers (S5 & S6)

Air block (H50000)

Vent patches (S9)

Airbag Opening

Rigid inflator rim (S3)

Dummy orifice locator ring, dynamically inactive. (S100000)

Rigid inflator box (S4)

FLUID PARTS SET-UP

Air definition using new EOS card defaulting to 1 atm pressure initially. This is the only fluid part that initially has a mesh defined.

```
*PART
surrounding air block = ale block
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
     50000     50000     50000     50000     50000         0         0
*SECTION_SOLID
         50000        11
*MAT_NULL
$      MID       RHO        PC        MU      TEROD     CEROD        YM        PR
     50000 1.2906E-9  -1.0E-05       0.0       0.0       0.0
*EOS_IDEAL_GAS
$    EOSID        Cv        Cp        C1        C2        T0        V0
     50000     719.0    1006.0                    273.15       1.0
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
     50000         1  1.00e-04
$=========================================================================
*PART
point sources material property definitions
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
     50003     50003     50003     50003     50003         0         0
*MAT_NULL
$      MID       RHO        PC        MU      TEROD     CEROD        YM        PR
     50003 5.192E-10  -1.0E-05       0.0       0.0       0.0
*EOS_IDEAL_GAS
$    EOSID        Cv        Cp        C1        C2        T0        V0
     50003    899.00   1196.00                    800.00000       1.0
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
     50003         1  1.00e-04
```

Inflator gas (initially requires no mesh definition – since it is supplied from the "point source reservoir") material property definition is defined here.

Inflator gas "nominal" state at P=1atm is obtained from an inverse process that assume isentropic path: Solid propellant ➔ burnt ➔ expand to final gaseous state in the tank ➔ expand this to 1atm to get …

FLUID PARTS SET-UP → INFLATOR GAS PROPERTIES

HOW DO WE DEFINE:

(1) "WHAT THERMODYNAMIC STATE" THE INPUT GAS IS AT? → T & RELVOL

(2) "WHAT SPEED" THE GAS IS FLOWING? → SPEED CURVE

(3) "WHERE" THE ORIFICES ARE LOCATED? → ARBITRARY NODES

(4) "WHICH DIRECTION" THE INFLATOR GAS IS INJECTED INTO THE SYSTEM?

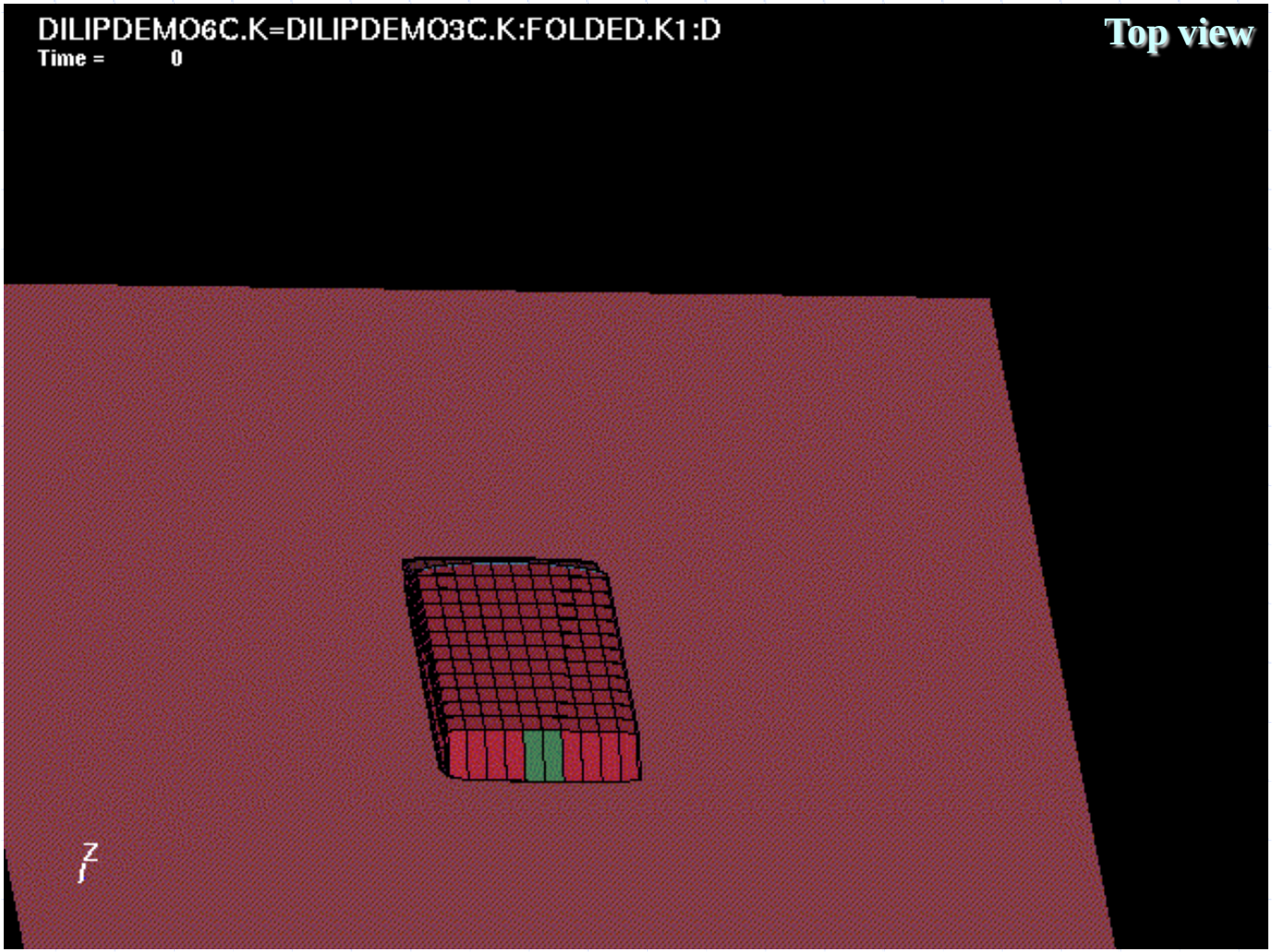
(5) "WHAT INFLATOR ORIFICE AREA(S)" TO BE USED FOR EACH SOURCE?

(1) (2) (3) (4) (5)

```
*SECTION_POINT_SOURCE
$     SECID     LCIDT  LCIDVOLR   LCIDVEL      <= 3 curves in tempvolrvel.k file
      50003      2011      2111      2001
$   NODEID    VECTID      AREA
    100019         1     8.750
    100020         2     8.750
    100021         3     8.750
    100022         4     8.750
    100023         5     8.750
    100024         6     8.750
    100017         7     8.750
    100018         8     8.750
*DEFINE_VECTOR
$       VID        XT        YT        ZT        XH        YH        ZH
          1       0.0       0.0 -24.50000  21.21320  21.21320 -24.50000
          2       0.0       0.0 -24.50000  30.00000-1.000e-06 -24.50000
          3       0.0       0.0 -24.50000  21.21320 -21.21320 -24.50000
          4       0.0       0.0 -24.50000-1.000e-06 -30.00000 -24.50000
          5       0.0       0.0 -24.50000 -21.21320 -21.21320 -24.50000
          6       0.0       0.0 -24.50000 -30.000001.0000e-06 -24.50000
          7       0.0       0.0 -24.50000 -21.21320  21.21320 -24.50000
          8       0.0       0.0 -24.500001.0000e-06  30.00000 -24.50000
```

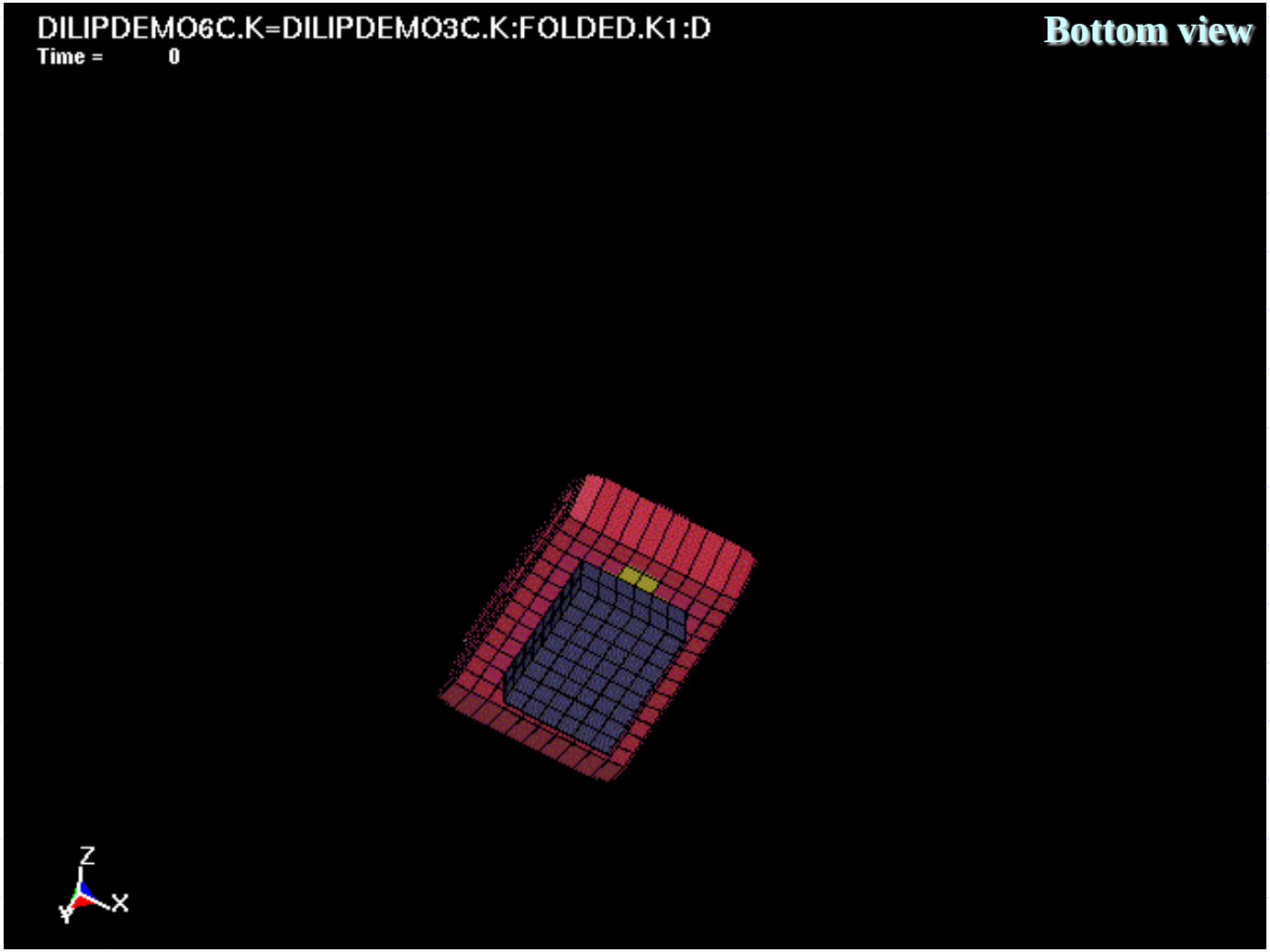
Airbag Deployment AVI

(double-click on picture below)



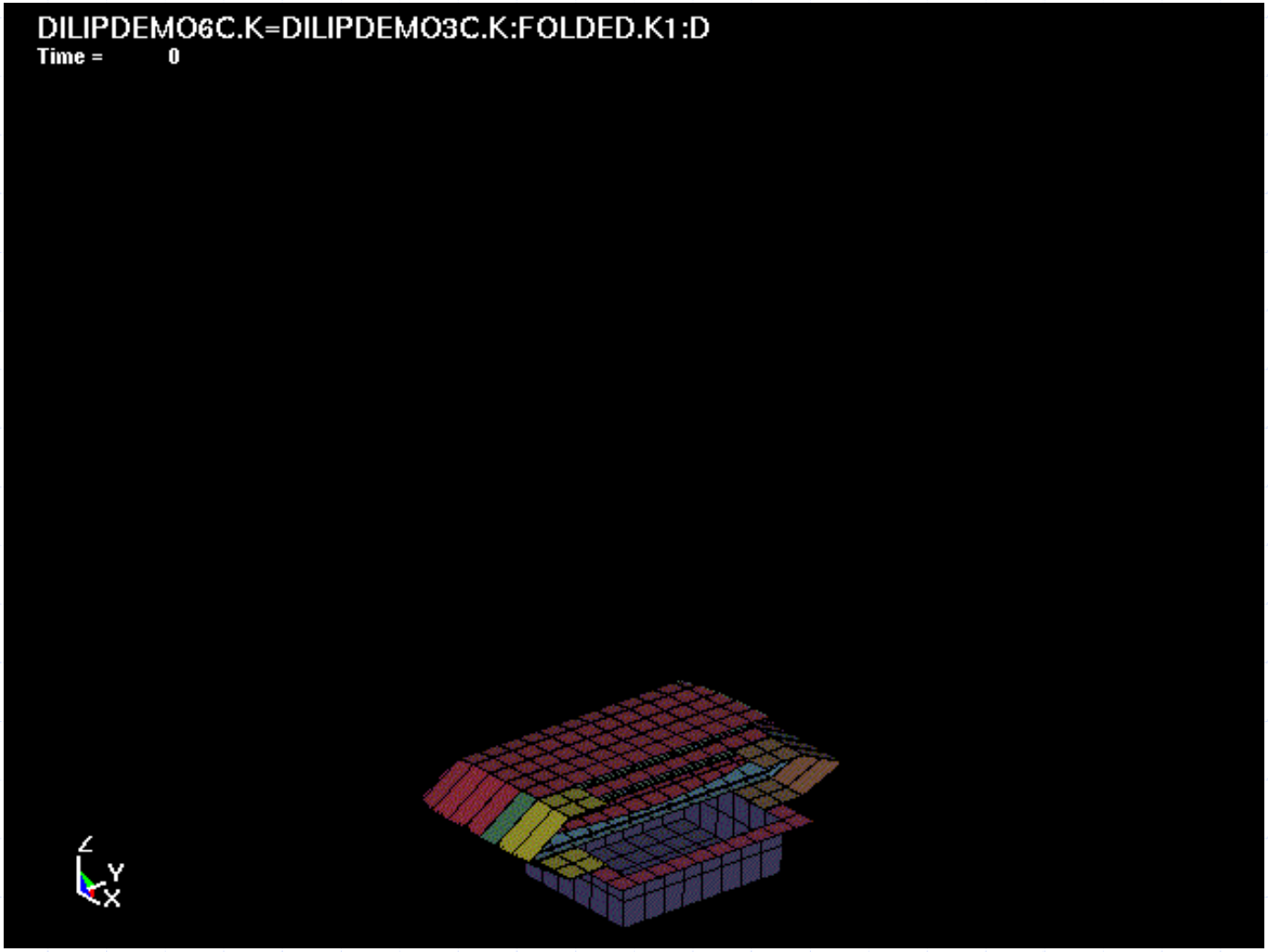
Airbag Deployment AVI

(double-click on picture below)



Airbag Deployment AVI

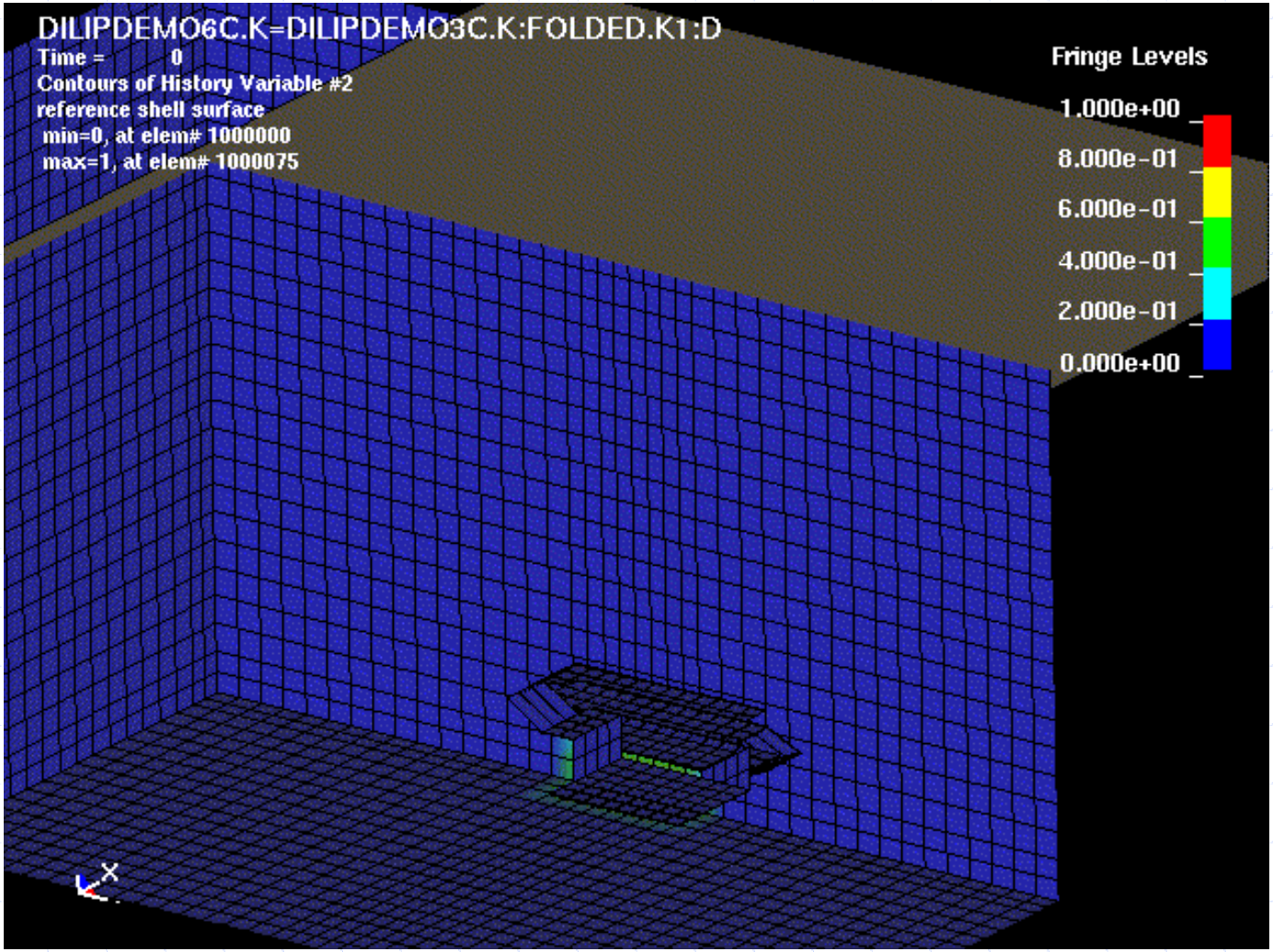
(double-click on picture below)



An animation of a cross sectional cut of the airbag – the fluids (gas and air) are not displayed.

Airbag Deployment Cross-Section AVI

(double-click on picture below)



An animation of a cross sectional cut of the airbag – the fluids (gas and air) are displayed also. The red fluid flows in from the inflator.

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

Ultimately, we want to attach an airbag to an inflator box, which itself is fixed to a moving steering wheel. For this example, we will first deal with the simple condition of a fixed inflator box. The moving of reference systems will be added in later example. We will be concerned with the following constraints:

LAGRANGIAN

- Anchoring the inflator box using *MAT_RIGID cards.
- Providing 2 *RIGID_WALL (masters) for airbag (slave=NSID 90) motion constraints.
- Providing single-surface contact for all Lagrangian components via *CONTACT_AUTOMATIC_SINGLE_SURFACE card.

Using ALE Coupling to Model Airbag Inflation Process

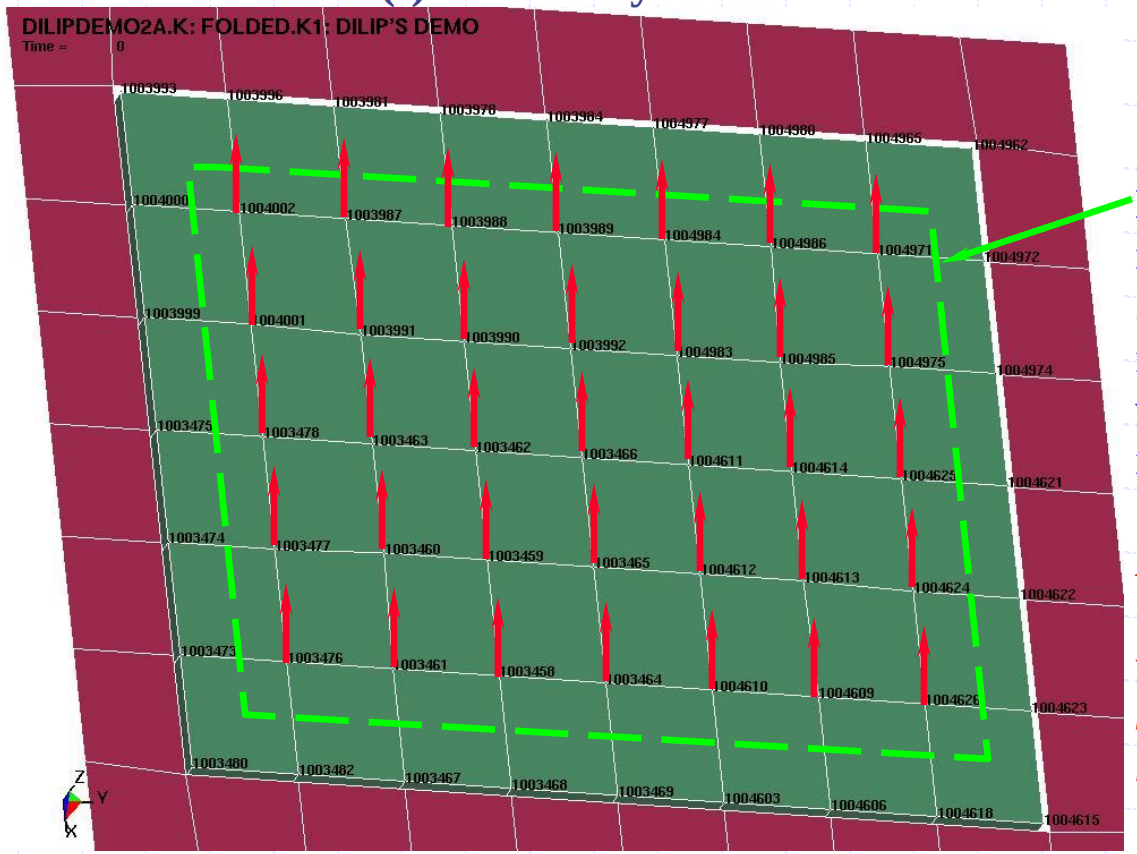
APPLYING BOUNDARY & CONSTRAINT CONDITIONS: *EULRIAN/ALE*

- Providing P=1atm to all 6 faces, [4 sides + top + bottom], of the air-gas meshes (SGSID 2) *LOAD_SEGMENT_SET card.
- Providing internal-energy-per-ref-vol, [$e_{ipv0}(t)$], and relative-volume, [$v_r(t)=vol/vol0$], using *BOUNDARY_AMBIENT_EOS card. Note that this applies to PID H7 which is defined with a *SECTION_SOLID_ALE with AET=4=ambient solid element type.
- Providing velocity to the nodes on the top face of the priming gas block. This supplies the flow into the bag ➔ inflating pressure. This nodal velocity is set by *BOUNDARY_PRESCRIBED_MOTION_SET card, giving all nodes in NSID 203 the v(t) defined by LCID 103.
- Providing coupling between the ALE "fluids" (master = PID's H7 & H8) and the Lagrangian airbag (slave = PID's S1-S6) via *CONSTRAINED_LAGRANGE_IN_SOLID card.

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

1. A ***BOUNDARY_PRESCRIBED_MOTION_SET** card giving all nodes in **NSID 203** the v(t) defined by **LCID 103.**



Nodes on the top face of the inflator gas block (NSID 203) facing the airbag mouth are given nodal velocity to provide the flow inflating the airbag.

NOTE:
The edge nodes have zero velocity. See notes on boundary inflow setting in later section.

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

- A *BOUNDARY_PRESCRIBED_MOTION_SET card giving all nodes in NSID 203 (all nodes on surface of priming gas block) a velocity, v(t), defined by LCID 103. This is the inlet velocity into the airbag (see figure in previous slide).

```
$=========================================================================
$ [BC #1]
$ Provides velocities for the nodes at the inflator mouth = airbag inflow.
$ DOF    = 1=x;      2=y;      3=z;
$ VAD    = vel|acc|displacement flag: = 0= vel  (rigbod   &   nodes)
*BOUNDARY_PRESCRIBED_MOTION_SET
$NID|NSID|PID    DOF       VAD      LCID        SF       VID     DEATH     BIRTH
       203         3         0       103       1.0         0       0.0       0.0
*DEFINE_CURVE
103
0.000,120.0
200.0,120.0
$
$ NSID 203 = 5X7 = 35 nodes the whole top surface without the edge nodes
$ A = 66X99 = 6534 mm^2 = half the original area used.
$
*SET_NODE_LIST
       203       0.0       0.0       0.0       0.0
   1004002   1003987   1003988   1003989   1004984   1004986   1004971   1004001
   1003991   1003990   1003992   1004983   1004985   1004975   1003478   1003463
   1003462   1003466   1004611   1004614   1004625   1003477   1003460   1003459
   1003465   1004612   1004613   1004624   1003476   1003461   1003458   1003464
   1004610   1004609   1004626
$=========================================================================
```

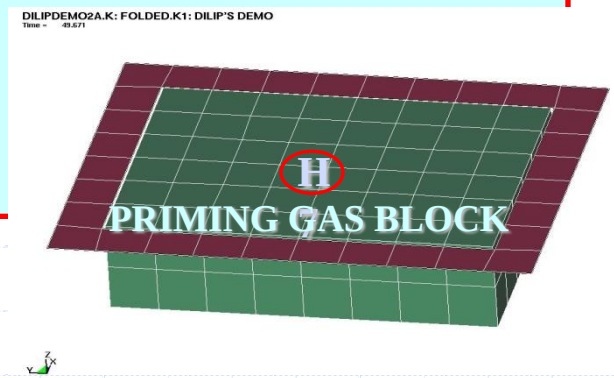
Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

- Using the *BOUNDARY_AMBIENT_EOS card to provide the thermodynamic condition of a fluid via 2 variables: internal-energy-per-ref-vol, [$e_{ipv0}(t)$], and relative-volume, [$v_r(t)=vol/vol0$].
- This applies to PID H7 which is defined with a *SECTION_SOLID_ALE with AET=4=ambient solid element type.

```
$=========================================================================
$ [BC #2]
$ LCID 101 = e_ipv0(t) = internal_E_per_ref_vol; LCID 102 = relative_vol = v_r(t)=vol/vol0=rho0/rho
*BOUNDARY_AMBIENT_EOS
$       PID       LC1       LC2
          7       101       102
*DEFINE_CURVE
101
0.00000,25.00E-4
200.000,25.00E-4
*DEFINE_CURVE
102
0.000,0.7
200.0,0.7
```

$$P(e_{ipv0}, v_r) = (\gamma - 1) * e_{ipv0} / v_r$$



Defining thermodynamic condition of the in-flow gas.

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

e_{ipv0} is defined as the **internal energy** per unit **reference volume** (*zero stress state*)

$$e_{ipv0} = \frac{E_i}{V_0} = \frac{[MC_vT]}{[V_0]} \sim \frac{[kg]\left[\frac{J}{kg * K^o}\right][K^o]}{[m^3]}$$

e_{ipv0} can be obtained by

$$e_{ipv0} = \frac{[e_{internal\_per\_mass}]}{[V_0/M]} = \frac{[C_vT]}{[1/\rho_0]} = \rho_0 C_v T$$

Make sure the unit system for C_v is correct for the calculation of e_{ipv0}

$$e_{ipv0} = \rho_0 C_v T \sim \left[\frac{kg}{m^3}\right]\left[\frac{J}{kg * K^o}\right][K^o] = \frac{J}{m^3}$$

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

The relative volume, v_r, is defined as the current volume over reference volume. As conservation of mass is enforced, it can be expressed in terms of the specific volume (volume per mass) ratio, or equivalently, density ratio.

$$v_r = \frac{V}{V_0} = \frac{V/M}{V_0/M} = \frac{\upsilon}{\upsilon_0} = \frac{\rho_0}{\rho}$$

The final result is energy per unit current volume ~ N*m/m^3 ~ N/m^2 ~ P

$$\frac{e_{ipv0}}{v_r} = \frac{\rho_0 C_v T}{\rho_0/\rho} = \rho C_v T = \frac{energy}{unit\_current\_volume}$$

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

- A *LOAD_SEGMENT_SET card provides a uniformed pressure load of 1 atm to the 6 faces surrounding the fluid meshes.
- The fluid mesh 6 faces are defined in SGSID 2.
- The pressure load curve, P(t), is defined by LCID 43.
- Since the air is defined with 1 atm internal pressure in the *EOS_ card, this BC counteracts with the internal pressure and prevents the air from automatically flushing out of the mesh (this can make dt very small and stop the run).

```
$=========================================================================
$ [BC #3]
$ This sgsid 2 contains : 4 side-faces + top-face + bot-face of air-gas blocks.
$ Provides uniform P = 1atm ~ 1.0e-4 on segments surrounding air block.
*LOAD_SEGMENT_SET
$     SGSID       LCID         SF  ARRIVALt
         2         43       1.00
*DEFINE_CURVE
$      LCID       SIDR        SFA        SFO       OFFA       OFFO    DATTYP
        43          0 0.0000000 0.0000000 0.0000000 0.0000000
$           X=abcissa          Y=ordinate
      0.00000000E+00      1.00000000E-04
      200.0000000000      1.00000000E-04
$=========================================================================
```

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

- A *CONSTRAINED_RIGID_BODIES card attaches the inflator box rim (S3) to the inflator box (S4).
- The *RIGIDWALL_PLANAR cards define the top and bottom backing platforms. NSID 90 contains all Lagrangian nodes that may come into contact with these walls.

```
$=========================================================================
$ [BC #4]
*CONSTRAINED_RIGID_BODIES
         4         3
$=========================================================================
$ [BC #5]
$ NSID 90 contains the slave nodes.  Defining top & bottom rigid backing walls.
*RIGIDWALL_PLANAR
$     NSID    NSIDEX     BOXID
        90         0         0
$       XT        YT        ZT        XH        YH        ZH      FRIC      WVEL
 0.0000000 0.0000000  391.0000 0.0000000 0.0000000 291.00000 0.1000000
*RIGIDWALL_PLANAR
$     NSID    NSIDEX     BOXID
        90         0         0
$       XT        YT        ZT        XH        YH        ZH      FRIC      WVEL
 0.0000000 0.0000000 -1.500000 0.0000000 0.0000000 990.00000 0.0000000
$=========================================================================
```

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

- A *CONTACT_AUTOMATIC_SINGLE_SURFACE card provides a simple contact control for PSID 1 (containing Lagrangian parts 1-6, 9 = [airbag + inflator parts).
- There are many ways to define this type of contact. This is just a simple example.
- The level of sophistication of the contact definition(s) will depend on the complexity of the airbag folding pattern.

```
$=========================================================================
$ [BC #6]
$ SSTYP= SLAVE set types:0=SGSID ; 1=SHSID ; 2=PSID ; 3=PID ; 4=NSID ; 6=PSID
*CONTACT_AUTOMATIC_SINGLE_SURFACE
$     SSID      MSID     SSTYP     MSTYP    SBOXID    MBOXID       SPR       MPR
         1         0         2         0         0         0         0         0
$       FS        FD        DC        VC       VDC    PENCHK        BT        DT
 0.5000000 0.0000000 0.0000000 0.0000000 0.0000000         0 0.0000000  0.000000
$      SFS       SFM       SST       MST      SFST      SFMT       FSF       VSF
 0.0000000 0.0000000 0.5000000 0.0000000 0.0000000 0.0000000
$     SOFT    SOFSCL    LCIDAB    MAXPAR      EDGE     DEPTH     BSORT    FRCFRQ
         2 0.0000000         0 0.0000000 0.0000000         3         5         0
*SET_PART_LIST
         1
         1         2         3         4         5         6         9
$=========================================================================
```

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

- A *CONSTRAINED_LAGRANGE_IN_SOLID card provides a fluid structure coupling between the slave PSID 1001 (Lagrangian shell airbag parts S1-S6) and the master PSID 1004 (Eulerian solid fluid parts H7 & H8).
- Since all the airbag parts are shell elements➔ coupling type=CTYPE=4.
- Coupling is activated under compression: DIREC=2.
- Turn on coupling effect when the fluid volume faction is > 03, FRCMIN=0.3.

```
$=========================================================================
$ [BC #7]
*CONSTRAINED_LAGRANGE_IN_SOLID
$     SLAVE     MASTER     SSTYP     MSTYP     NQUAD     CTYPE     DIREC     MCOUP
      1001       1004         0         0         6         4         2         1
$     START       END      PFAC      FRIC    FRCMIN      NORM
          0         0      0.10         0       0.3         0
$        CQ      HMIN      HMAX     ILEAK
          0         0         0         1
*SET_PART_LIST
      1001
         1         2         3         4         5         6
*SET_PART_LIST
      1004
         7         8
$=========================================================================
```

Using ALE Coupling to Model Airbag Inflation Process

APPLYING BOUNDARY & CONSTRAINT CONDITIONS:

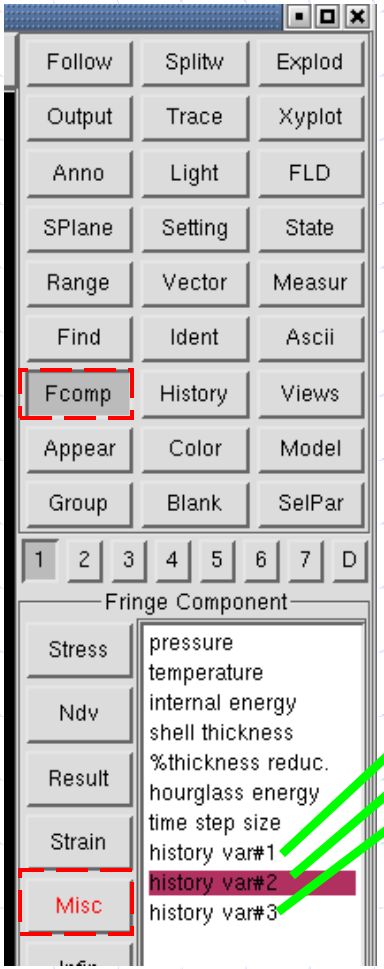
- A *CONTROL_ALE card defines the basic continuum treatment method used and advection controls.
- DCT = 2 = Eulerian method
- NADV = 1 = number of cycle(s) per advection step, typically 1.
- METH = 2 = Van Leer 2nd order with half-index-shift (HIS).
- An *ALE_MULTI-MATERIAL_GROUP card defines the ALE material group whose interfaces are to be tracked during the deformation process. The 1st line define the PID of the 1st ALE material group, etc. This is history variable # 2 in LS-POST: under FCOMP ➔ MISC ➔ HISVAR 2 ➔ APPLY.

```
$=========================================================================
*CONTROL_ALE
$       DCT       NADV       METH       AFAC       BFAC       CFAC       DFAC       EFAC
          2          1          2     -1.00      0.000      0.000      0.000
$     START        END      AAFAC      VFACT     VLIMIT        EBC
     0.000      0.000      0.000
*ALE_MULTI-MATERIAL_GROUP
$       SID     IDTYPE
          7          1
          8          1
$=========================================================================
```

Some Tips on ALE Post-Processing Using LS-POST Example: Airbag Inflation Process

Some Simple ALE Post-Processing Tips using *LS-POST*

History variable plotting: Density and volume fractions (vf).



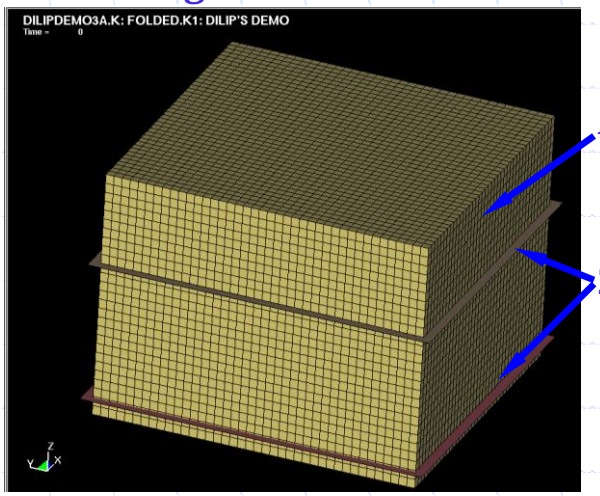
NOTE:
Plotting of Lagrangian parts readily shows the material deformation because the mesh follows the material. Since Eulerian or ALE materials "flow" in their meshes, we need to plot, instead, their volume fractions which describe the interfaces defining the material boundaries.

The resolution of the mesh defines the resolution of the interfaces.

History var # 1 = Density
History var # 2 = vf of the 1st ALE material
History var # 3 = vf of the 2nd ALE material
History var # 4 = vf of the 3rd ALE material
… etc.
(Additional history variables may depend on the material model used).

Some Simple ALE Post-Processing Tips using *LS-POST*

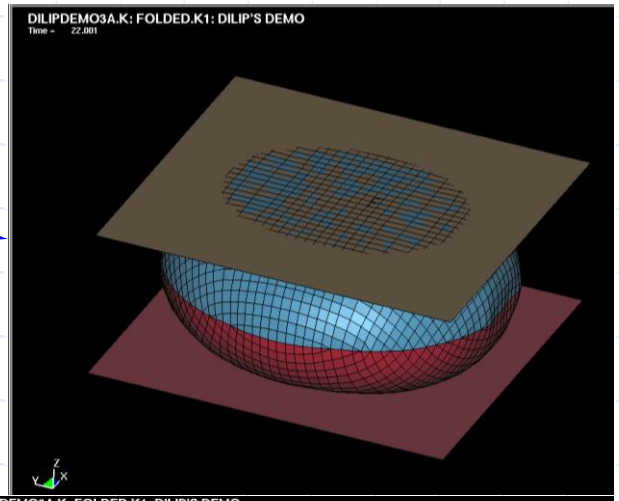
The airbag model is used for illustration. Assume the following set-up:



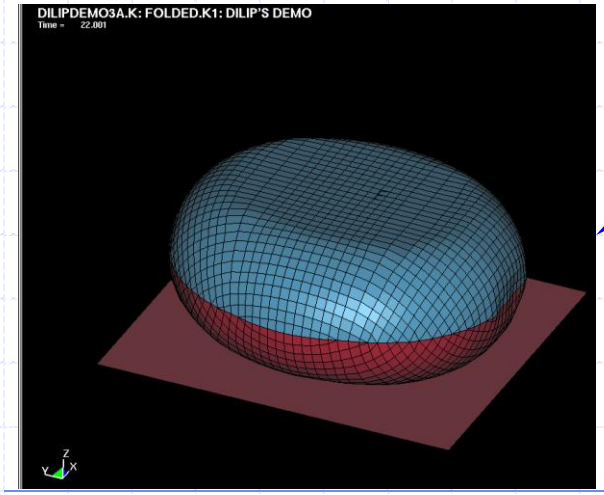
Air mesh

2 rigid walls

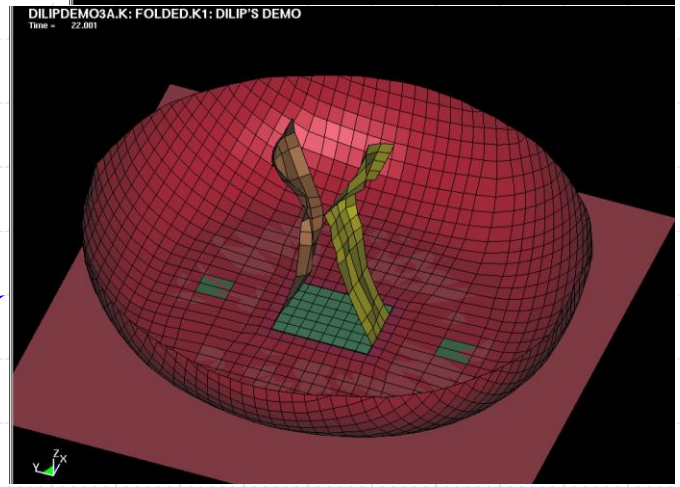
Blanking out the ALE mesh



Blanking out the top rigid wall

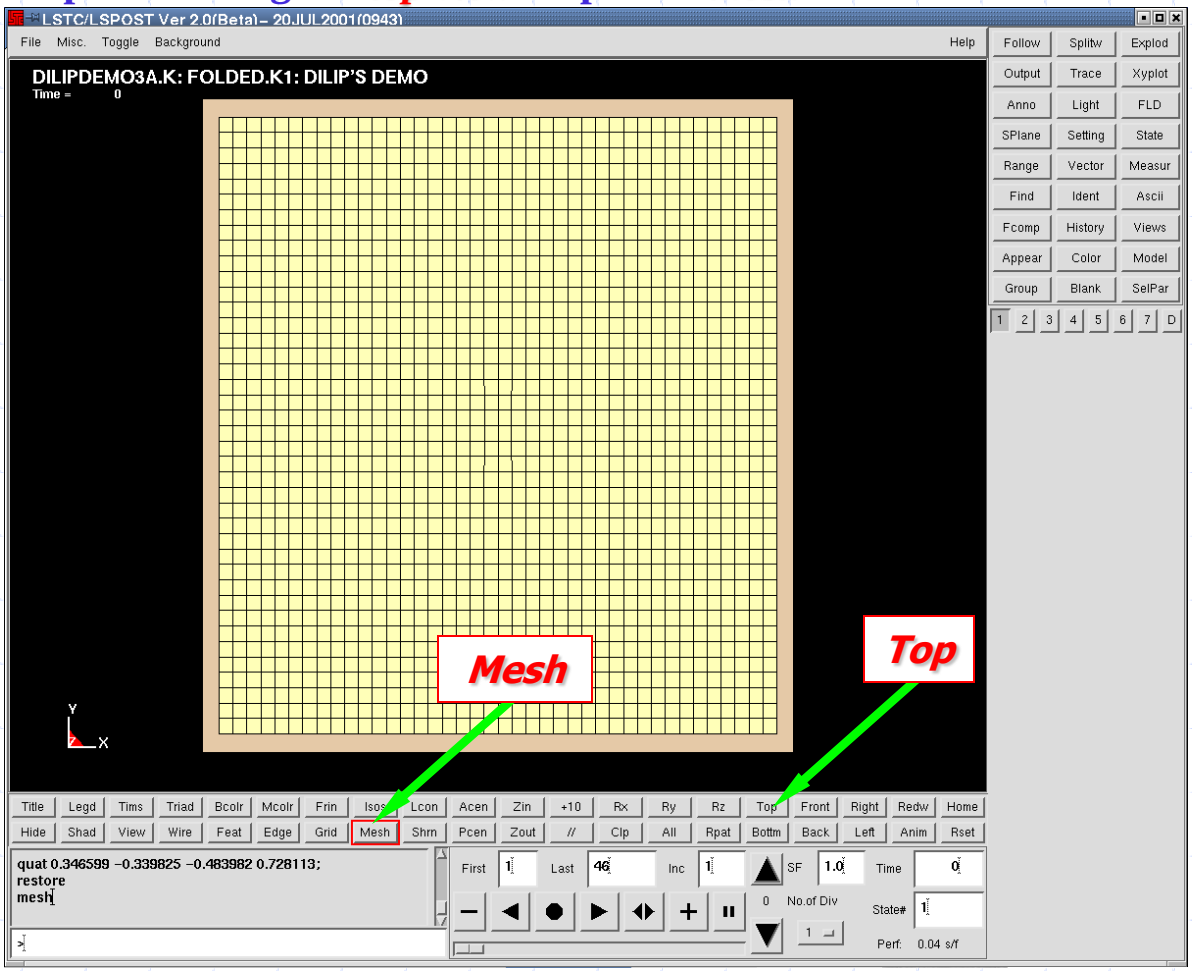


Blanking out the top airbag shell

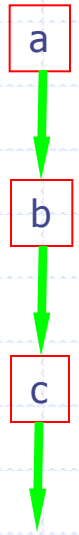


Some Simple ALE Post-Processing Tips using *LS-POST*

[1] Load the d3plot file ➔ go to *top view* ➔ put on *mesh*.

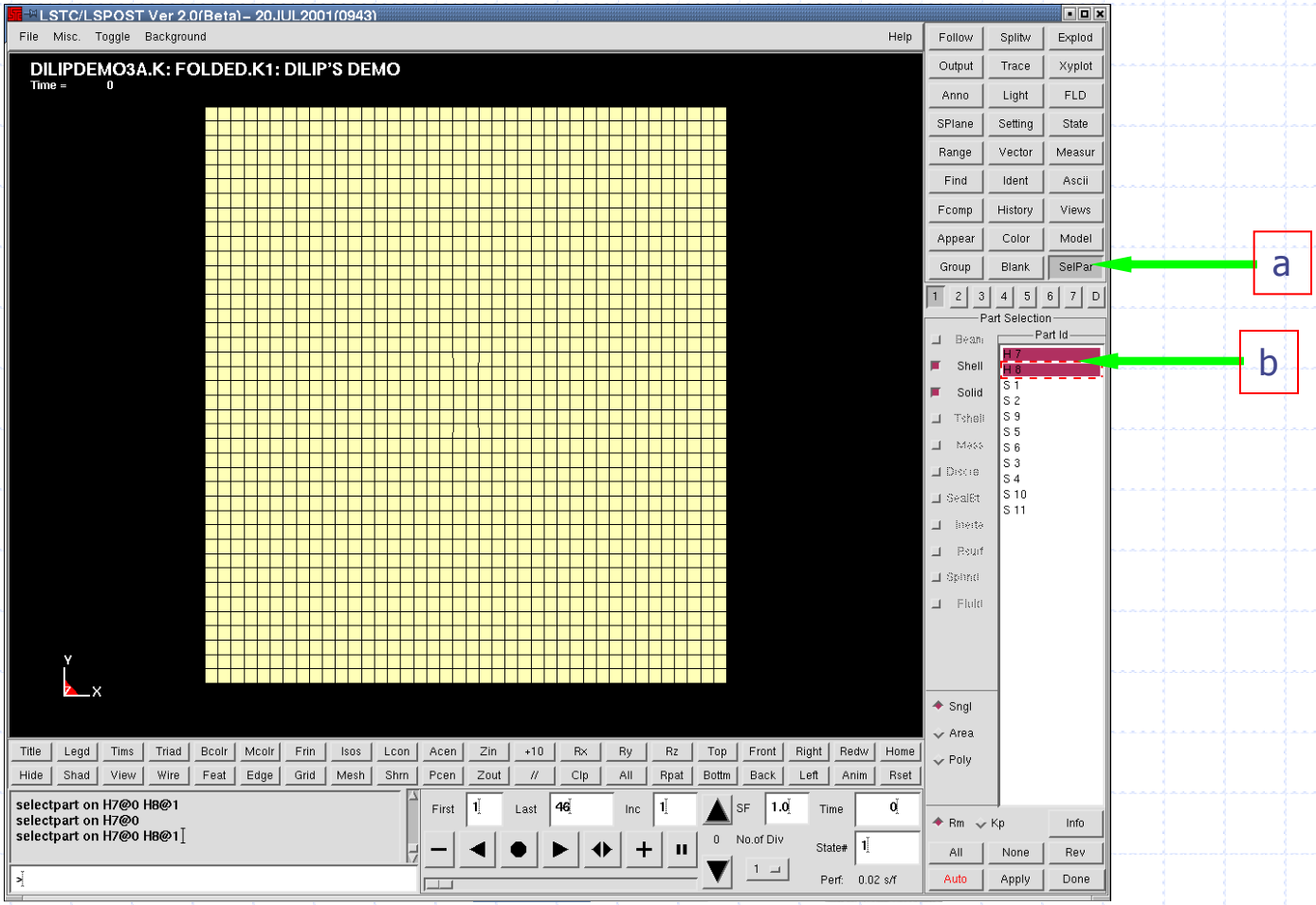


Subsequently actions follow the sequence



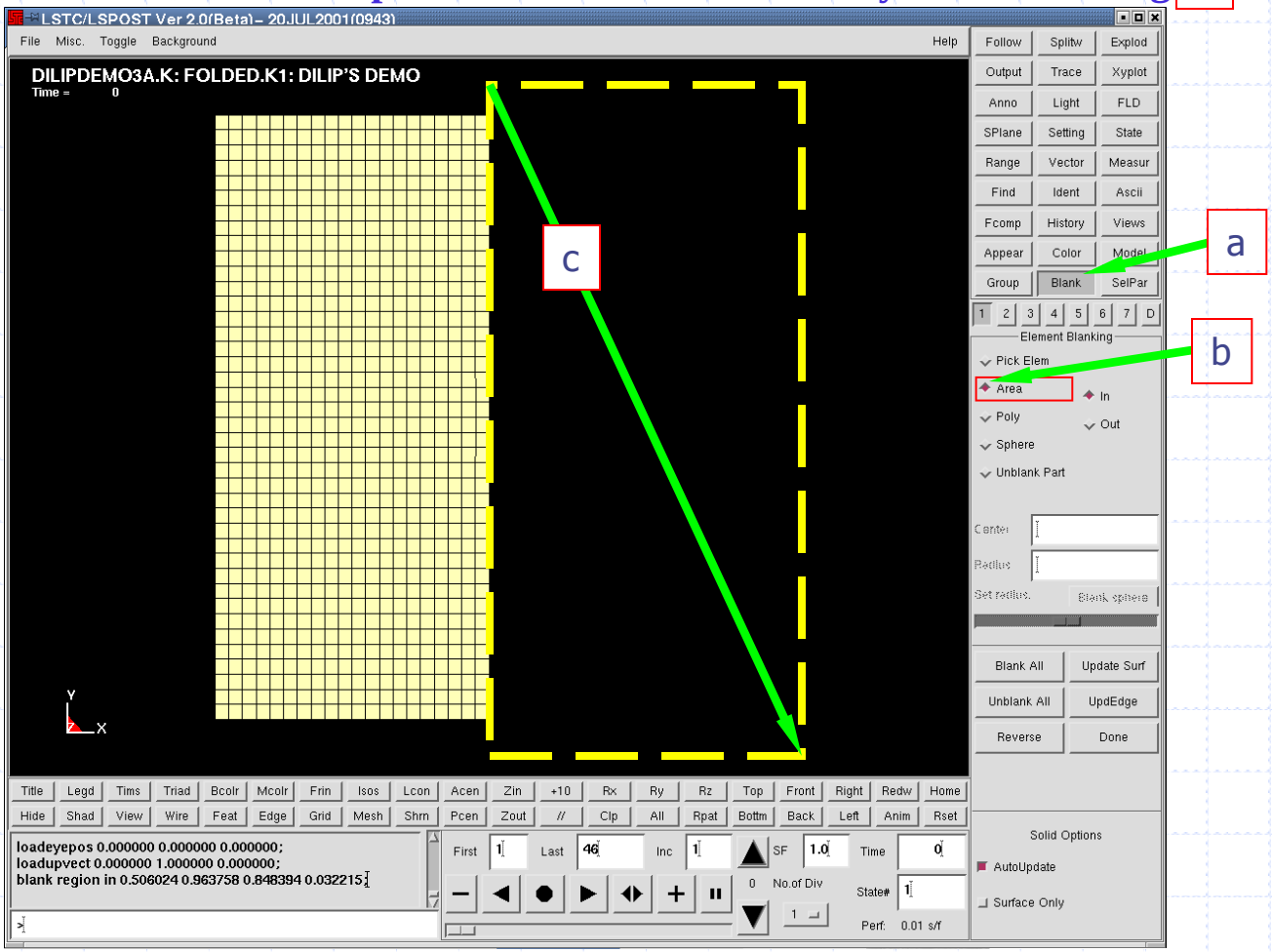
Some Simple ALE Post-Processing Tips using *LS-POST*

[2] From *top view* ➔ *Select* only the ALE(or fluid) meshes (PIDs: H7 and H8 in this case).



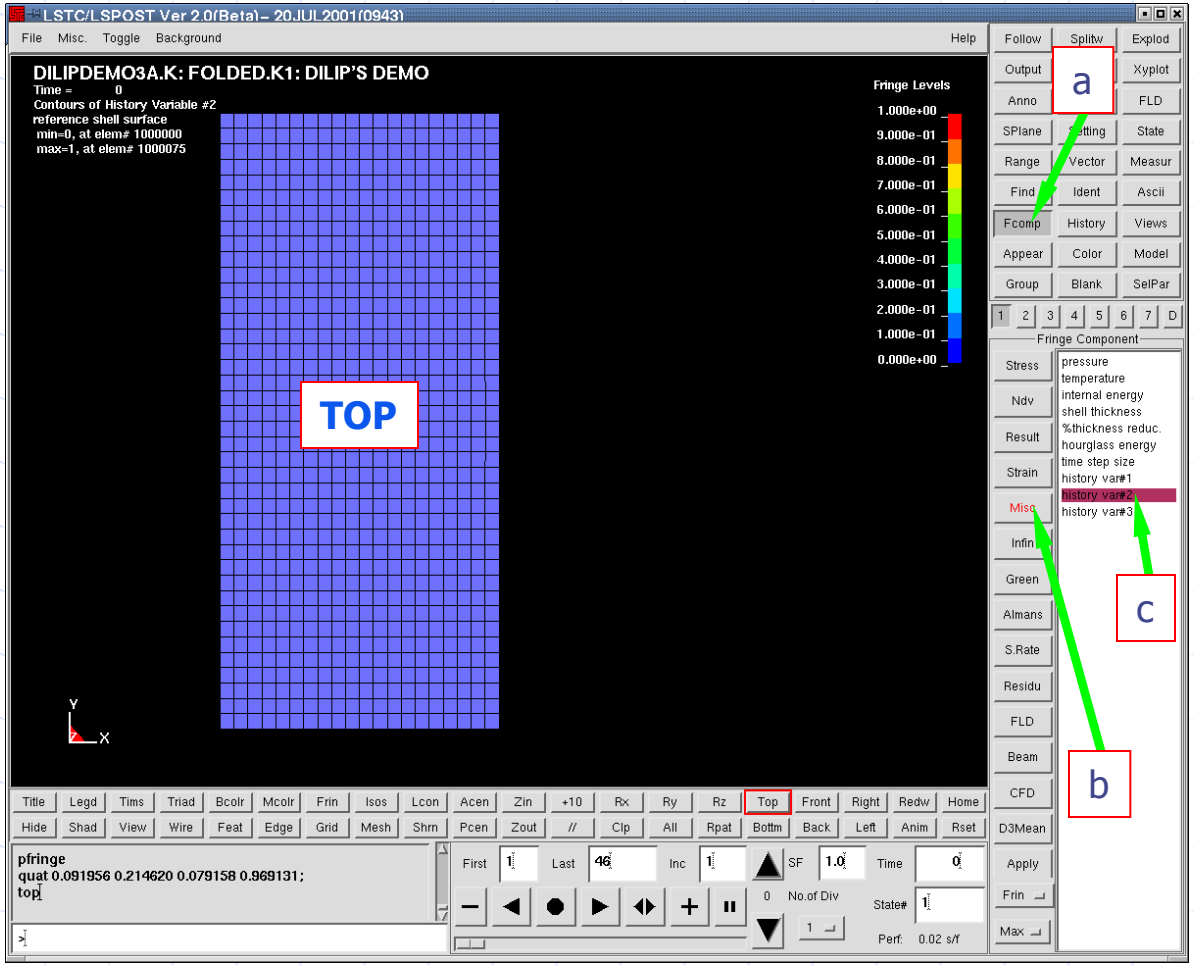
Some Simple ALE Post-Processing Tips using *LS-POST*

[3] From *top view* ➔ *blank* out a portion of the ALE meshes by click-and-drag c.



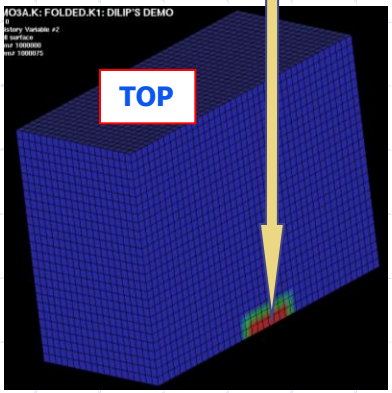
Some Simple ALE Post-Processing Tips using *LS-POST*

[4] *fringe-plot* volume fraction (vf) of ALE material.



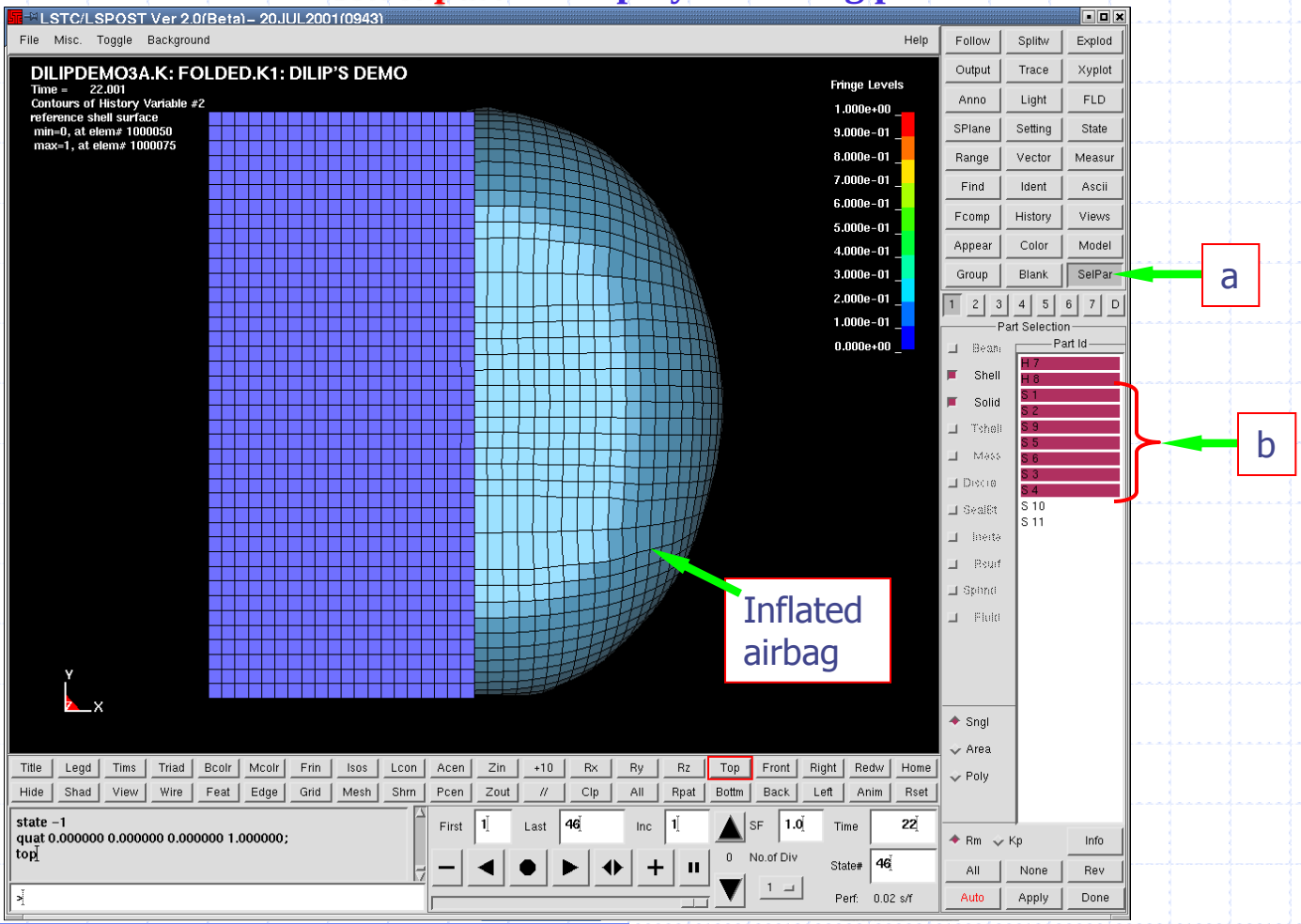
Selecting *fcomp*→*misc*→*history-var#* to select the volume fraction of the ALE material to be monitored.

Obtaining at-and-angle view (red portion is the primer gas)



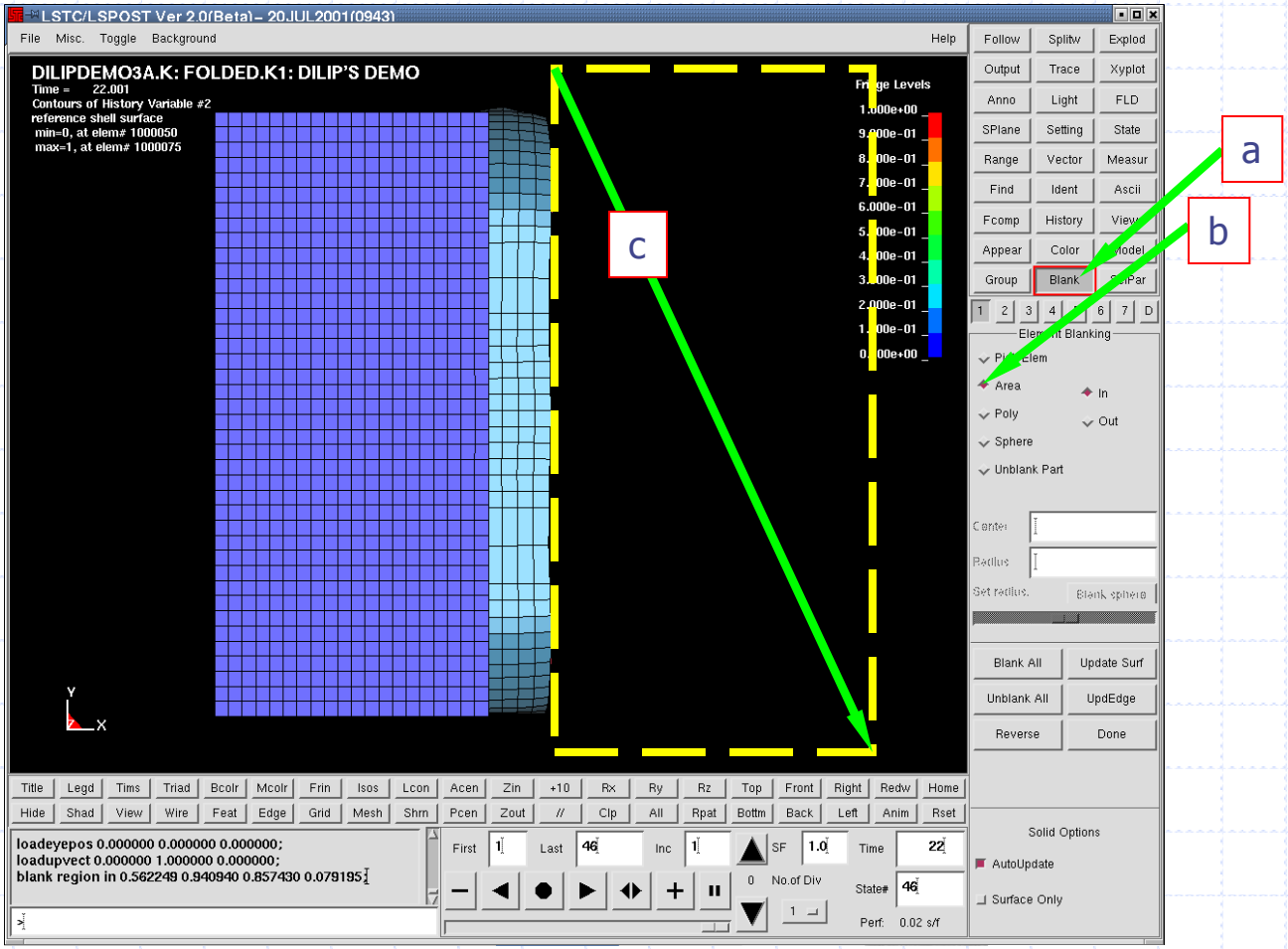
Some Simple ALE Post-Processing Tips using *LS-POST*

[5] Go to inflated condition ➔ *select-part* ➔ display all airbag parts



Some Simple ALE Post-Processing Tips using *LS-POST*

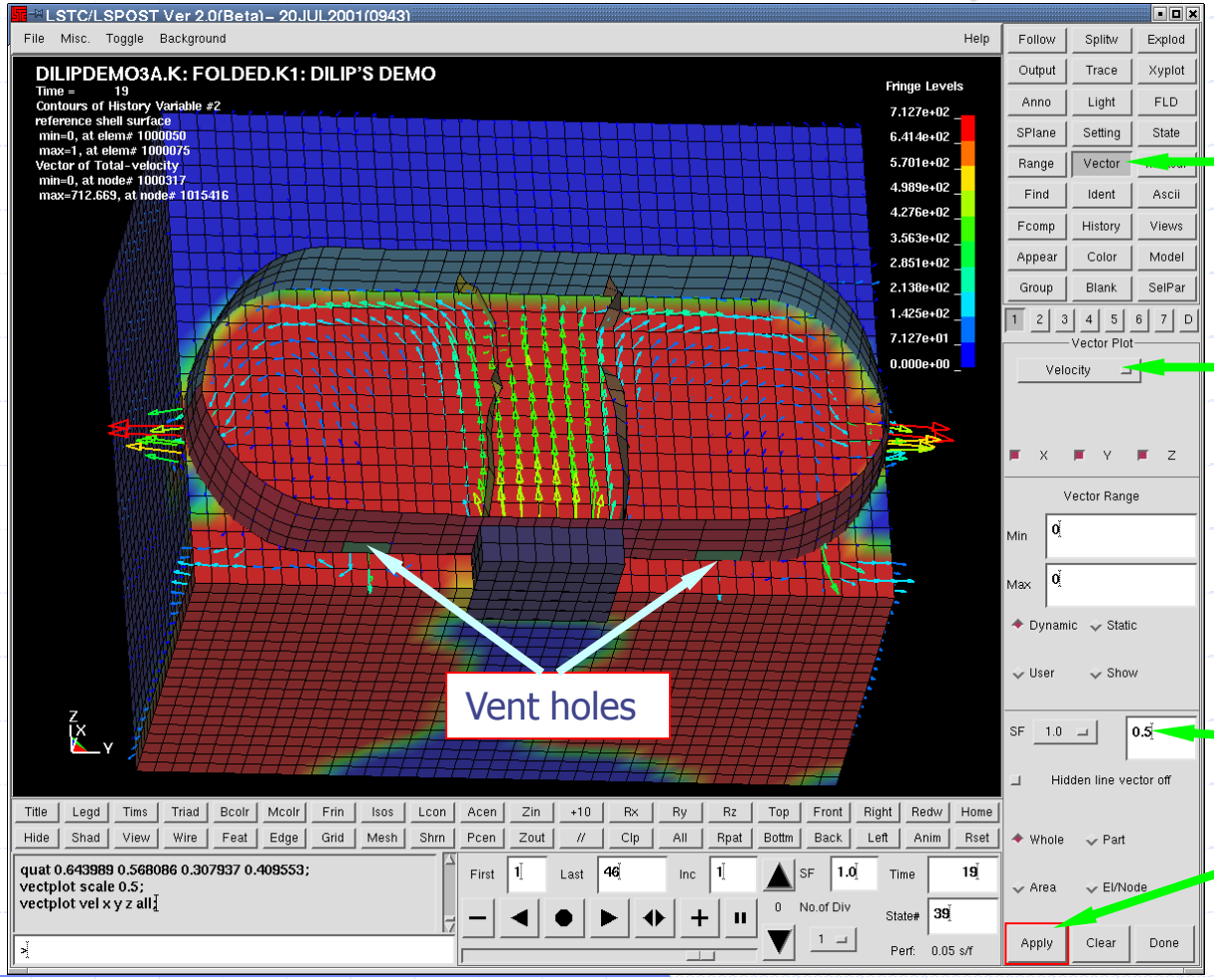
[6] *blank* out portion of the airbag to see through into the fluid flow ➔ still top view.



Some Simple ALE Post-Processing Tips using *LS-POST*

[7] Use control-left-mouse to rotate the figure ➔ *vector* ➔ *velocity* ➔ *SF=0.5* ➔ *apply*

- The *Red* region is the vf of the 1st ALE material.
- Vectors show velocity pattern in the fluid.



Vector plotting must be selected after selecting the "other" component plotting.

Some CAUTIONS on ALE
Inflow VELOCITY Boundary Condition Setting

ALE Inflow Boundary Condition Setting

Assume that we want to define a flow rate across a "face" region in the ALE/Eulerian mesh. One simple method is to assign nodal velocities to the nodes on this region. Due to the use of interpolating (shape) functions in FEM, one must consider the accumulative effect of this interpolation. We can start by looking at the effect of "1 nodal velocity on 1 element area". Then the accumulative effect can be obtained by superposition. First, a quick review of the general interpolation procedure over a quad surface is provided.

$$\dot{V} = \int_S \vec{v} \cdot \vec{n} * dA = \int_{s=-1}^{s=1} \int_{r=-1}^{r=1} v(r,s) * dr * ds = volume - flow - rate = \bar{v} * A$$

$$v(r,s) = N_1(r,s) * v_1 + N_2(r,s) * v_2 + N_3(r,s) * v_3 + N_4(r,s) * v_4 = \text{interpolated velocity}$$

$$N_i(r,s) = \frac{1}{4}(1 + r_i r)(1 + s_i s)$$

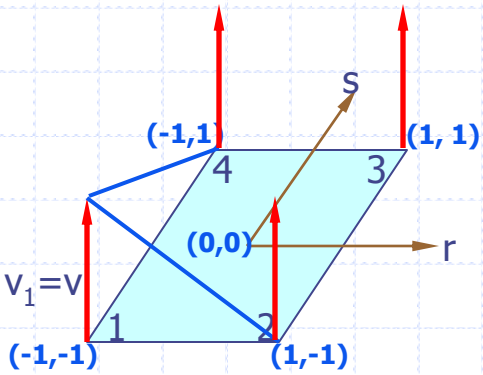
$$[r_1 = -1, s_1 = -1] \Rightarrow N_1(r,s) = \frac{1}{4}(1-r)(1-s)$$

$$[r_2 = +1, s_2 = -1] \Rightarrow N_2(r,s) = \frac{1}{4}(1+r)(1-s)$$

$$[r_3 = +1, s_3 = +1] \Rightarrow N_3(r,s) = \frac{1}{4}(1+r)(1+s)$$

$$[r_4 = -1, s_4 = +1] \Rightarrow N_4(r,s) = \frac{1}{4}(1-r)(1+s)$$

= shape functions (Cook-Malkus-Plesha Pages 167,173)



ALE Inflow Boundary Condition Setting

The volume flow rate over 1 quad surface area is

$$\dot{V} = \int_S \vec{v} \cdot \vec{n} * dA = \int_{s=-1}^{s=1} \int_{r=-1}^{r=1} \left[N_1(r,s) * v_1 + N_2(r,s) * v_2 + N_3(r,s) * v_3 + N_4(r,s) * v_4 \right] * dr * ds$$

The volume flow rates over 1 element surface due to each of the 4 nodal velocities are

$$\int_{s=-1}^{s=1} \int_{r=-1}^{r=1} N_1(r,s) * v_1 * dr * ds = \int_{s=-1}^{s=1} \int_{r=-1}^{r=1} \frac{1}{4}(1-r)(1-s) * v_1 * dr * ds$$

$$\int_{s=-1}^{s=1} \int_{r=-1}^{r=1} N_2(r,s) * v_2 * dr * ds = \int_{s=-1}^{s=1} \int_{r=-1}^{r=1} \frac{1}{4}(1+r)(1-s) * v_2 * dr * ds$$

$$\int_{s=-1}^{s=1} \int_{r=-1}^{r=1} N_3(r,s) * v_3 * dr * ds = \int_{s=-1}^{s=1} \int_{r=-1}^{r=1} \frac{1}{4}(1+r)(1+s) * v_3 * dr * ds$$

$$\int_{s=-1}^{s=1} \int_{r=-1}^{r=1} N_4(r,s) * v_4 * dr * ds = \int_{s=-1}^{s=1} \int_{r=-1}^{r=1} \frac{1}{4}(1-r)(1+s) * v_4 * dr * ds$$

ALE Inflow Boundary Condition Setting

Considering each sub-integral separately

$$\int_{r=-1}^{r=1}(1-r)*dr=\left(r-\frac{r^2}{2}\Big|_{-1}^{1}\right)=\left(1-\frac{1}{2}\right)-\left(-1-\frac{1}{2}\right)=2 \qquad \int_{s=-1}^{s=1}(1-s)*ds=\left(s-\frac{s^2}{2}\Big|_{-1}^{1}\right)=\left(1-\frac{1}{2}\right)-\left(-1-\frac{1}{2}\right)=2$$

$$\int_{r=-1}^{r=1}(1+r)*dr=\left(r+\frac{r^2}{2}\Big|_{-1}^{1}\right)=\left(1+\frac{1}{2}\right)-\left(-1+\frac{1}{2}\right)=2 \qquad \int_{s=-1}^{s=1}(1+s)*ds=\left(s+\frac{s^2}{2}\Big|_{-1}^{1}\right)=\left(1+\frac{1}{2}\right)-\left(-1+\frac{1}{2}\right)=2$$

The volume flow rate due to velocity at 1 node over the quad surface of 1 element is (just looking at the effect of node #1):

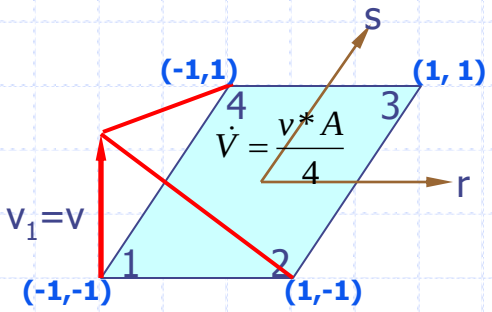
$$\int_{s=-1}^{s=1}\int_{r=-1}^{r=1}N_1(r,s)*v_1*dr*ds=\frac{v_1}{4}\int_{s=-1}^{s=1}\left[\int_{r=-1}^{r=1}(1-r)*dr\right]*(1-s)*ds=v_1=\bar{v}*A$$

$$\dot{V}_{due-to-1-node}=\bar{v}*A=v_1*1.0$$

$\bar{v}=\frac{v_1}{4}$ = equivalent average velocity over [area=4] of 1 element.

$$\dot{V}=\bar{v}*A=\frac{v*A}{4}$$

(area_ave_vel = nodal_vel/4)



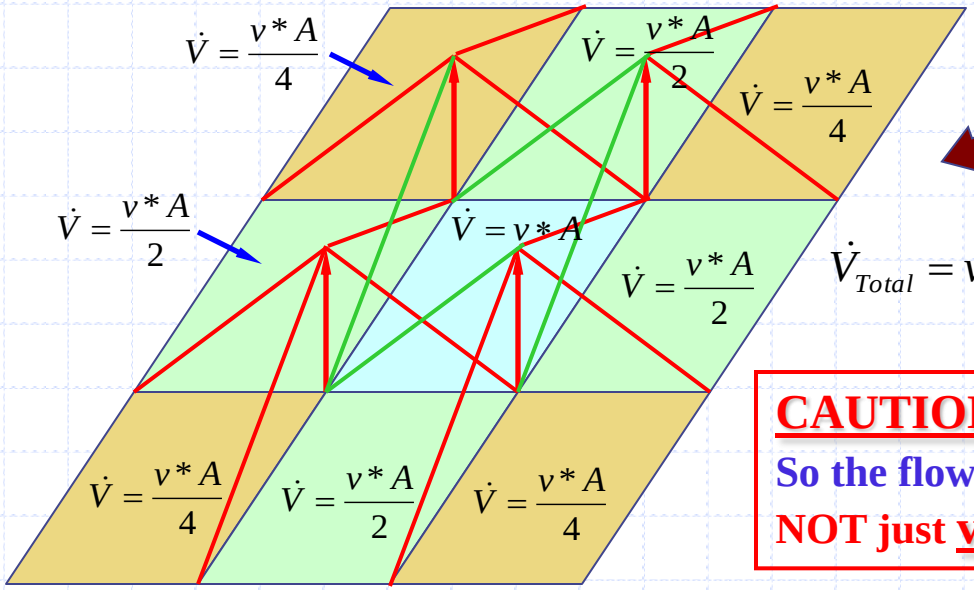
$v_2= v_3= v_4= 0$

ALE Inflow Boundary Condition Setting

So the total volume flow rate due to velocities at 4 nodes over the quad surface of 1 element is equal in magnitude to the sum of the 4 nodal velocities – for the center element only!

$$\dot{V} = \int_S \vec{v} \cdot \vec{n} * dA = v_1 + v_2 + v_3 + v_4 \quad \text{If the velocities are equal} \rightarrow \quad \dot{V} = 4 * v = \bar{v} * A \Rightarrow \bar{v} = v$$

Now consider the effect of 4 nodal velocities (assuming equal magnitude velocities and areas)



Artifact due to the linear interpolation nature of FEM

$$\dot{V}_{Total} = v*A + 4*\left(\frac{v*A}{2}\right) + 4*\left(\frac{v*A}{4}\right) = 4*v*A$$

CAUTION:

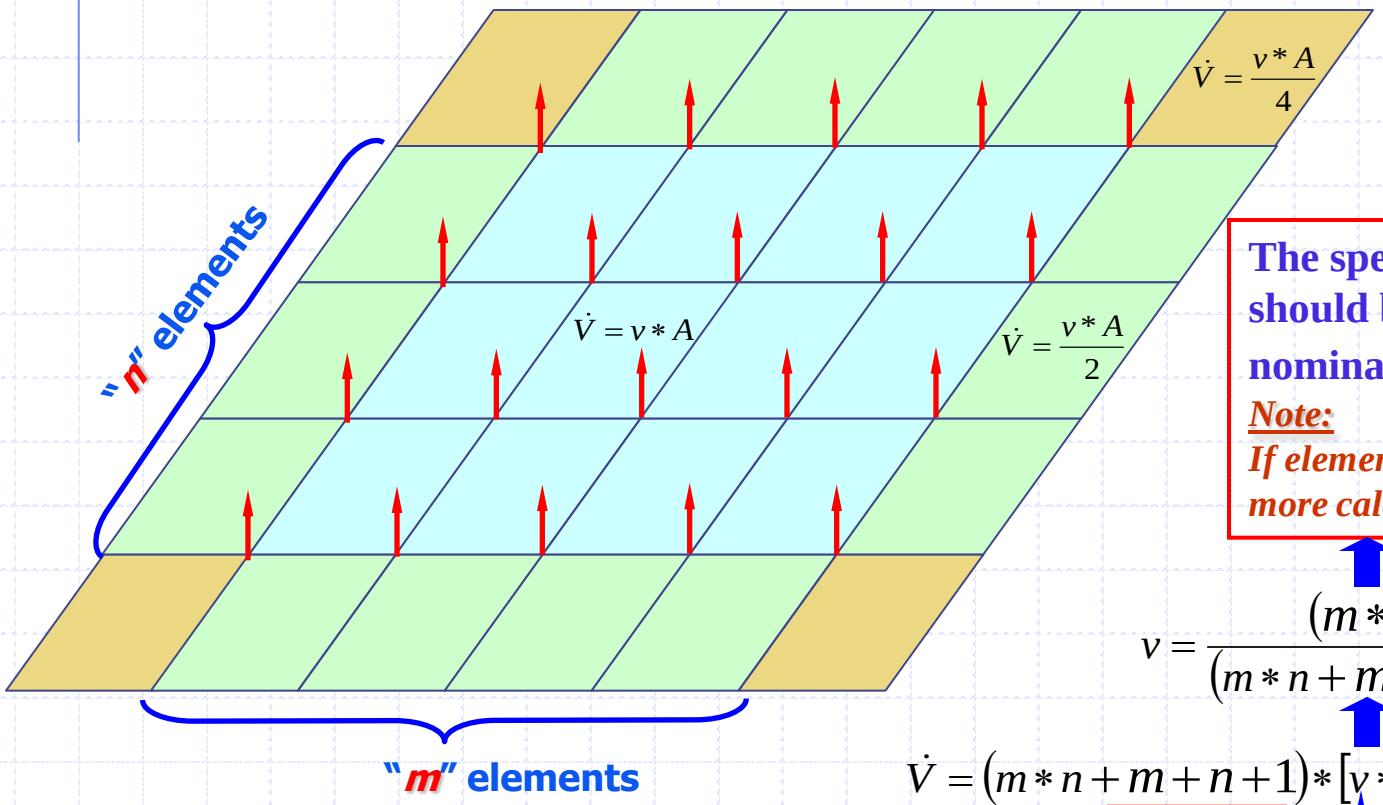
So the flow rate is actually 4 (v*A) value.

NOT just v*A!

ALE Inflow Boundary Condition Setting

In general, for "*m-by-n*" elements having nodal velocity defined at their nodes, we have

$$\dot{V} = m*n*[v*A] + 2*(m+n)*\left[\frac{v*A}{2}\right] + 4*\left[\frac{v*A}{4}\right]$$



The speed actually applied, v, should be less than the nominal, $\bar{v}$, value!

Note:

If element areas are not equal, more calculation is required!

$$v = \frac{(m*n)}{(m*n+m+n+1)} * \bar{v}$$

$$\dot{V} = (m*n+m+n+1)*[v*A] = m*n*[\bar{v}*A]$$

speed actually applied

Nominal value

One Approach to Volume-Filling For LS-DYNA ALE Analysis

Volume-Filling with LS-DYNA ALE Method

PHYSICAL DESCRIPTIONS

This Volume-Filling model is assumed to include 4 physical entities:

* A Lagrangian tank (S1).
* A surrounding air block (Eulerian=H3= the only fluid part or mesh initially defined).

*(Only the first 2 parts have their meshes defined initially.)_*

* A liquid fuel to be filled inside the tank (Eulerian=H4=no initial geometry defined).
* A fuel vapor to be filled inside the tank (Eulerian=H5 =no initial geometry defined).

(Parts H4 and H5 are the additional fluids that to be filled in certain specified spaces. No meshes are defined in the input deck initially for these 2 parts.)

* A rigid wall defined by *RIGIDWALL_PLANAR (no *PART definition required).

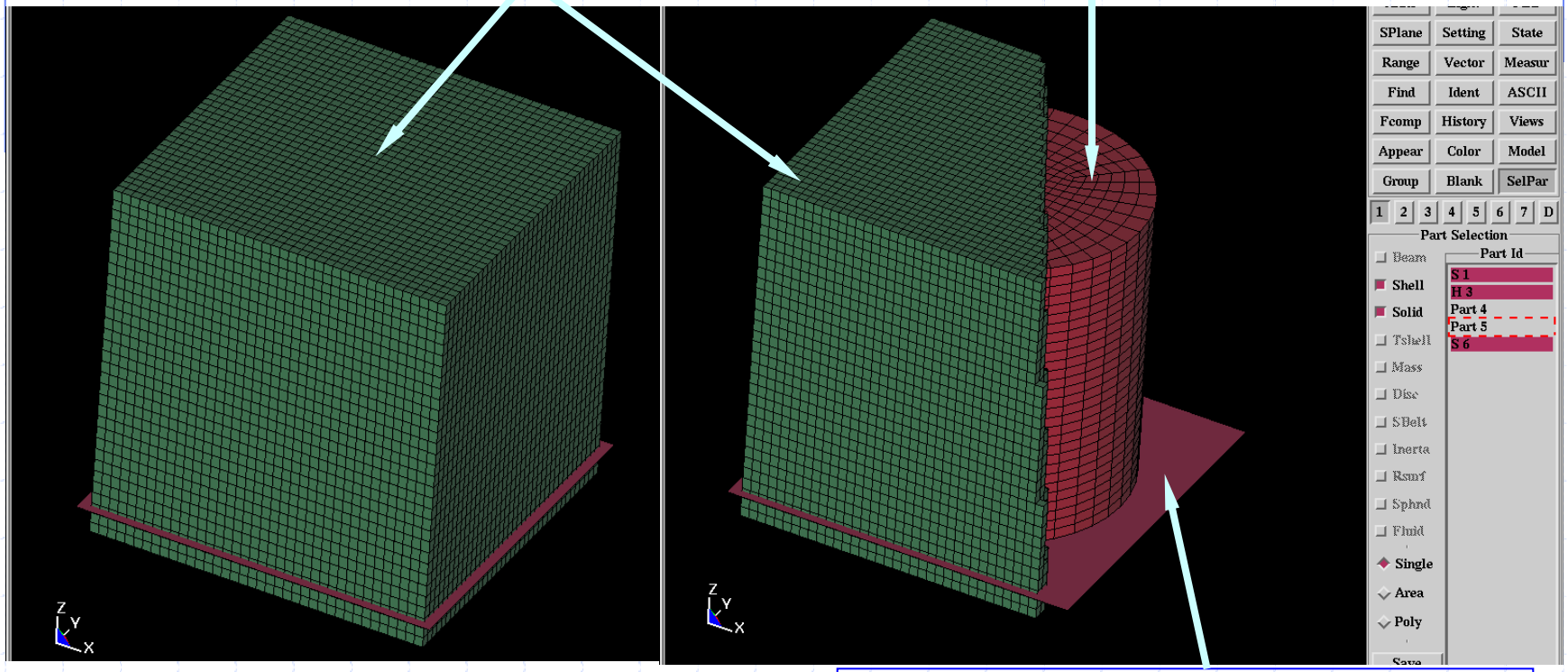
(This is a platform onto which the tank is dropped.)

Volume-Filling with LS-DYNA ALE Method

PHYSICAL DESCRIPTIONS

H3 = initial surrounding air mesh

S1 = tank (container) mesh



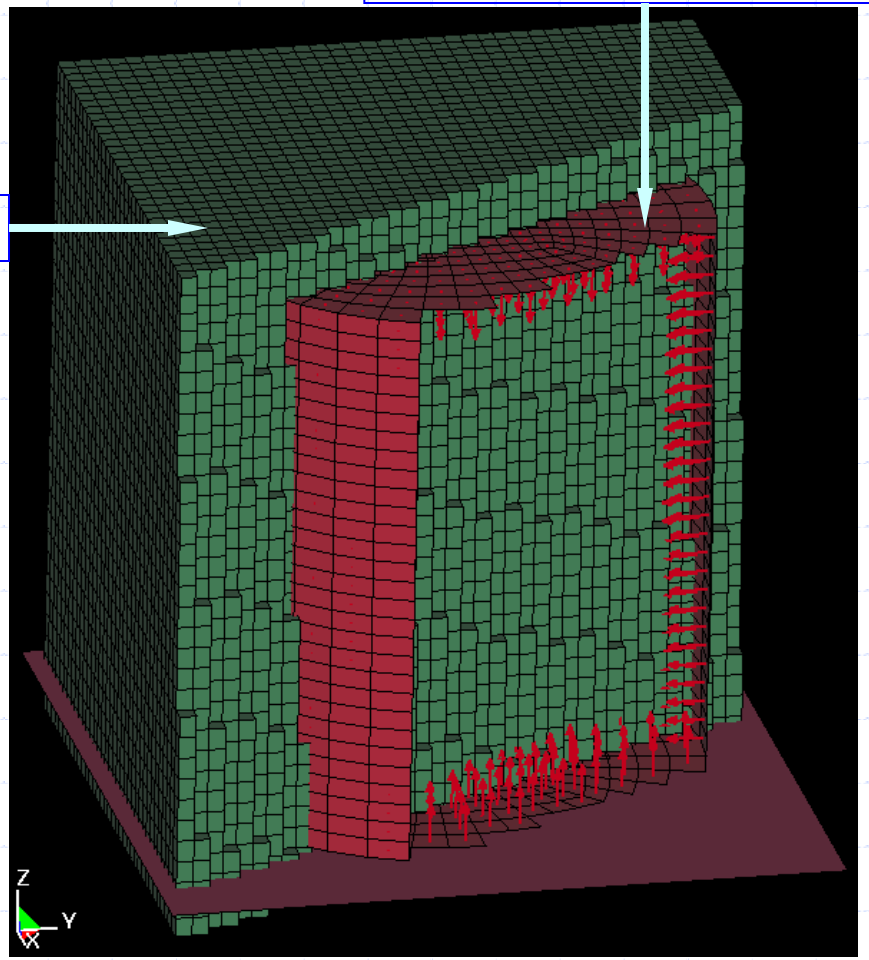
S6 = platform = *RIGIDWALL_PLANAR
(no *PART card required)

Volume-Filling with LS-DYNA ALE Method

PHYSICAL DESCRIPTIONS

S1 = tank (container) mesh

H3 = initial surrounding air mesh



The container is defined as a shell structure with all its segment normal vectors pointing INWARD uniformly.

The container should lie just outside of the fluid region to be contained.

Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: Parts with initial mesh definitions.

S1 = Lagrangian shell container

H3 = Eulerian surrounding air mesh

```
*PART
PID 1 = container = can = steel
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         1         1         1         0         0         0
*SECTION_SHELL
$    SECID    ELFORM       AET
         1         2         0
     0.200     0.200     0.200     0.200
*MAT_PLASTIC_KINEMATIC
$      MID        RO         E        PR      SIGY      ETAN      BETA
         1   7860.00 300.00E09     0.300 350.00E06 300.00E6       0.0
$      SRC       SRP        FS        VP
       0.0       0.0       0.0       0.0
$=========================================================================
*PART
background fluid (initially defined part or geometry or mesh) = air
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         3         3         3         3         3         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         3        11         0
*MAT_NULL
         3 1.2008462  -1.0E+02 1.8444E-5       0.0       0.0
*EOS_IDEAL_GAS
$    EOSID        Cv        Cp        C1        C2        T0        V0
         3     719.0    1006.0                    294.00       1.0
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
         3         1  1.00e-05
```

Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: Part without initial mesh definition.

H4 = Liquid fuel to be filled inside container

```
*PART
filling fluid =  liquid fuel to be filled inside gas tank
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         4         4         4         4         4         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         4        11         0
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
         4    998.21    -100.0 0.8684E-3       0.0       0.0       0.0       0.0
*EOS_GRUNEISEN
$    EOSID         C        S1        S2        S3     GAMAO         A        EO
         4  1.647E3     1.921    -0.096       0.0     0.350       0.0       0.0
$       VO
       1.0
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
         4         1  1.00e-04
```

Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: Part without initial mesh definition.

H5 = Fuel vapor to be filled inside container

Fixed rigid platform

```
*PART
The vapor gas of the liquid fuel to be filled inside the fuel tank (fuel vapor)
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
         5         5         5         5         5         0
*SECTION_SOLID
$    SECID    ELFORM       AET
         5        11         0
*MAT_NULL
$      MID        RO        PC        MU     TEROD     CEROD        YM        PR
         5 1.2008462  -1.0E+02 1.8444E-5       0.0       0.0
*EOS_IDEAL_GAS
$    EOSID        Cv        Cp        C1        C2        T0        V0
         5     719.0    1006.0                    294.00       1.0
*HOURGLASS
$     HGID       IHQ        QM       IBQ        Q1        Q2        QB        QW
         5         1  1.00e-05
$=========================================================================
*RIGIDWALL_PLANAR
$     NSID    NSIDEX     BOXID
         1
$       XT        YT        ZT        XH        YH        ZH      FRIC      WVEL
      0.0       0.0      -1.0       0.0       0.0       0.0       0.0       0.0
```

NSID1 = all nodes on the shell container (this is defined without a *PART card). After running ls970, LS-POST will show a part S6 for this rigid wall.

Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: *ALE_MULTI-MATERIAL_GROUP definition.

AMMG1 = ALE MULTI-MATERIAL GROUP number 1

```
$=========================================================================
$ S1 =          Lagrangian shell container
$ H3 = AMMG 1 = background air initially occupying the whole existing mesh.
$ H4 = AMMG 2 = filling fluid 1 = liquid fuel to be filled inside tank.
$ H5 = AMMG 3 = filling fluid 2 = fuel vapor  to be filled inside tank
$=========================================================================
*ALE_MULTI-MATERIAL_GROUP
         3         1
         4         1
         5         1
```

SET-ID's

The "1's" in this column indicates the SET-ID-TYPE's are "PART-ID's"

H3 = PID3 = AMMG1 = HISVAR2 = surrounding air.

H4 = PID4 = AMMG2 = HISVAR3 = liquid fuel.

H5 = PID5 = AMMG3 = HISVAR4 = fuel vapor.

Plotting in LSPREPOST: FCOMP ➔ MISC ➔HISTORY VARIABLES

Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: *INITIAL_VOLUME_FRACTION_GEOMETRY.

STEP 0 ➔ *initially fill PID 3 with AMMG1*

STEP 1 ➔

STEP 2 ➔

STEP 3 ➔

```
$=============================================================================
$  FILLOPT = Filling option:
$           0= fill inside  container geom = inside = side pointed to by normals
$           1= fill outside container geom = outside = opposite to "inside"
$  NORMDIR = 0=normals point INward of volume; 1=normals point OUTward of volume
$=============================================================================
*INITIAL_VOLUME_FRACTION_GEOMETRY
$ fill the whole pid 3 with AMMG 1=background air
$FPID/PSID   FIDTYPE INIAMMGID     <=== card 1: background fluid (norm point in)
         3         1          1
$ step 1 ------------------------------------------------------------------------
$ fill shell PID 1 (with inward normals) with AMMG 2=liquid fuel
$ CONTTYPE   FILLOPT FILAMMGID     <=== card 2: container #1 FILLOPT=0=fill head
         1         0          2
$    SETID   SETTYPE   NORMDIR     <=== card 3: details on container #1
         1         1          0
$ step 2 ------------------------------------------------------------------------
$ fill all elms above a "plane" @ z=400 with AMMG 3=gas both in and outside tank
$ CONTTYPE   FILLOPT FILAMMGID     <=== card 4: container #2: CONTTYPE=3=plane
         3         0          3
$   X0, Y0,   Z0,    NX, NY, NZ    <=== card 5: details on container #2=plane
   0.0,0.0,400.0,   0.0,0.0,1.0
$ step 3 ------------------------------------------------------------------------
$ fill all elms outside of tank with air again to get rid of the gas outside.
$ CONTTYPE   FILLOPT FILAMMGID     <=== card 6: container #1 FILLOPT=1=fill tail
         1         1          1
$    SETID   SETTYPE   NORMDIR     <=== card 7: details on container #1
         1         1          0
```

See next 2 pages for parameter definitions ➔

Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: *INITIAL_VOLUME_FRACTION_GEOMETRY.

```
$---------------------------------------------------------------------------
*INITIAL_VOLUME_FRACTION_GEOMETRY = filling container with AMMGID fluid(s)
$ CARD 1 ------------------------------------------------
$ FPID|PSID = BACKGROUND FLUID MESH PID:  FLUID MESH PID or PSID (geometry)
$   FIDTYPE = FLUID MESH ID type: (0=PSID    ,   1=PID)
$ INIAMMGID = INITIAL AMMG ID filling this mesh (from *ALE_MULTI-MATERIAL_GROUP)
$
$
$
$
$ CARD 2 ------------------------------------------------
$ CONTTYPE = Container Geometry Type
$          = 1 = Container is defined by a "PID" or "PSID"
$          = 2 = Container is defined by a "SGSID"
$          = 3 = Container is defined by a "PLANE"
$          = 4 = Container is defined by a "CYLINDER"
$          = 5 = Container is defined by a "BOX"
$          = 6 = Container is defined by a "SPHERE"
$  FILLOPT = Filling option:
$           0= fill inside  container geom = inside = side pointed to by normals
$           1= fill outside container geom = outside = opposite to "inside"
$ FILAMMGID= Filling Fluid ID = ALE MM group ID defined by a
$            *ALE_MULTI-MATERIAL_GROUP card
```

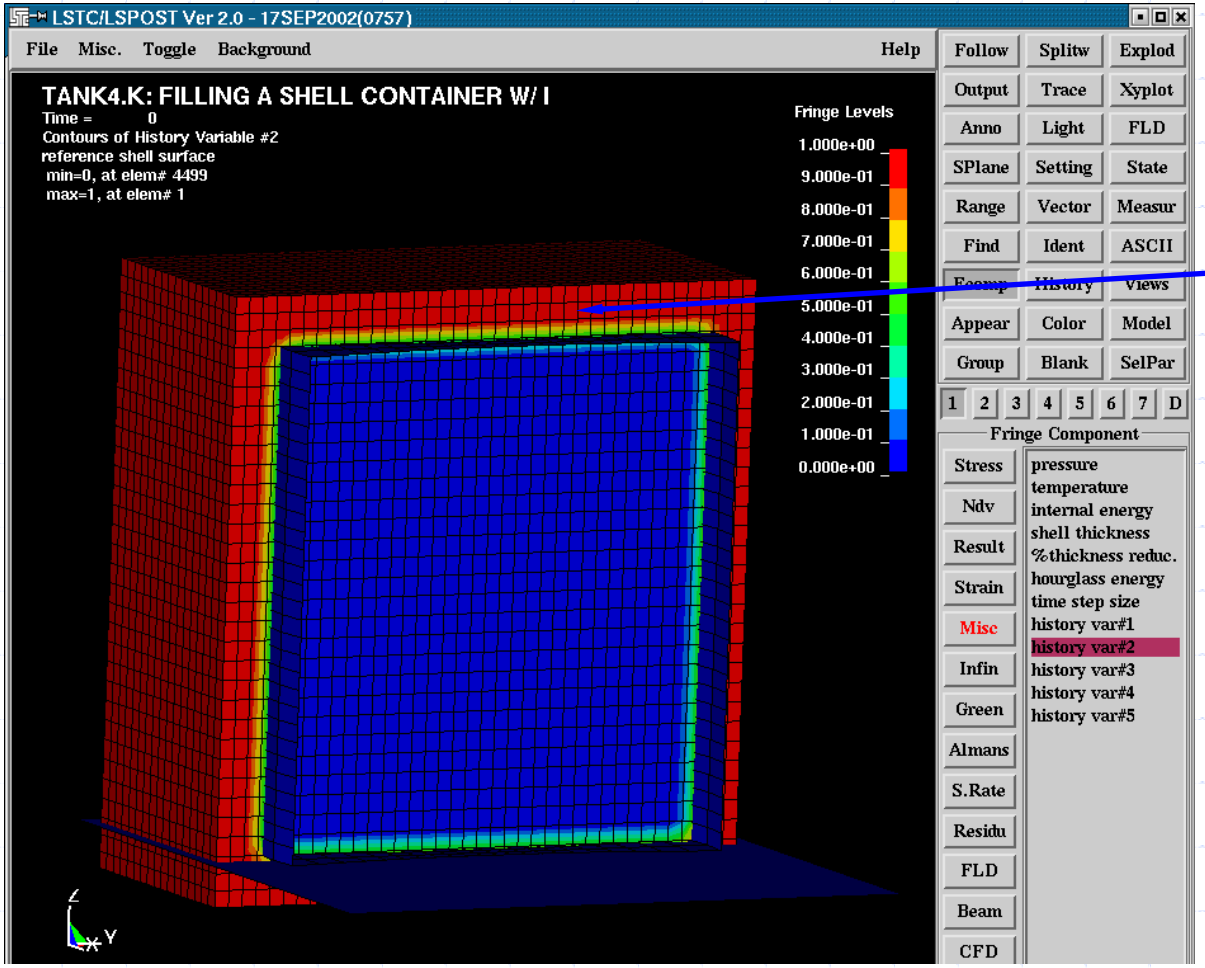
Volume-Filling with LS-DYNA ALE Method

LS-DYNA KEYWORDS: *INITIAL_VOLUME_FRACTION_GEOMETRY.

```
$ CARD 3 ------------------------------------------- CONTAINER GEOMETRY DEF
$ if in card 2: CONTTYP = 1 = Container is defined by "PID" or "PSID"
$    SETID = ID of container
$  SETTYPE = 0=PSID  or  1=PID
$  NORMDIR = 0=normals point INward of volume; 1=normals point OUTward of volume
$  ---------------------------------------------------------------------------
$ if in card 2: CONTTYP = 2 = Container is defined by "SGSID"
$    SGSID = SEGMENT SET ID of container
$  ---------------------------------------------------------------------------
$ if in card 2: CONTTYP = 3 = Container is defined by a "PLANE"
$ X0,Y0,Z0 = 1 coord ref. point on this plane
$ NX,NY,NZ = 3 direction cosines defining the plane normal.
$---------------------------------------------------------------------------
$ if in card 2: CONTTYP = 4 = Container is defined by a "CYLINDER"
$ X1,Y1,Z1 = 1st end point on center line of cylinder
$ X2,Y2,Z2 = 2nd end point on center line of cylinder
$       R1 = cylinder radius at point 1
$       R2 = cylinder radius at point 2
$  ---------------------------------------------------------------------------
$ if in card 2: CONTTYP = 5 = Container is defined by a "BOX"
$ XMIN,YMIN,ZMIN = 1st end point on diagonal line of "BOX"
$ XMAX,YMAX,ZMAX = 2nd end point on diagonal line of "BOX"
$  ---------------------------------------------------------------------------
$ if in card 2: CONTTYP = 6 = Container is defined by a "SPHERE"
$ XC,YC,ZC = center of the "SPHERE"
$ RADIUS   = radius of the "SPHERE"
$  ---------------------------------------------------------------------------
$ If there are more container definitions ==> they come in pair(s) as following:
$ CARD 2 = CARD 4 = define what type of container & how to fill it & what fluid
$ CARD 3 = CARD 5 = define container geometry definitions.
```

Volume-Filling with LS-DYNA ALE Method

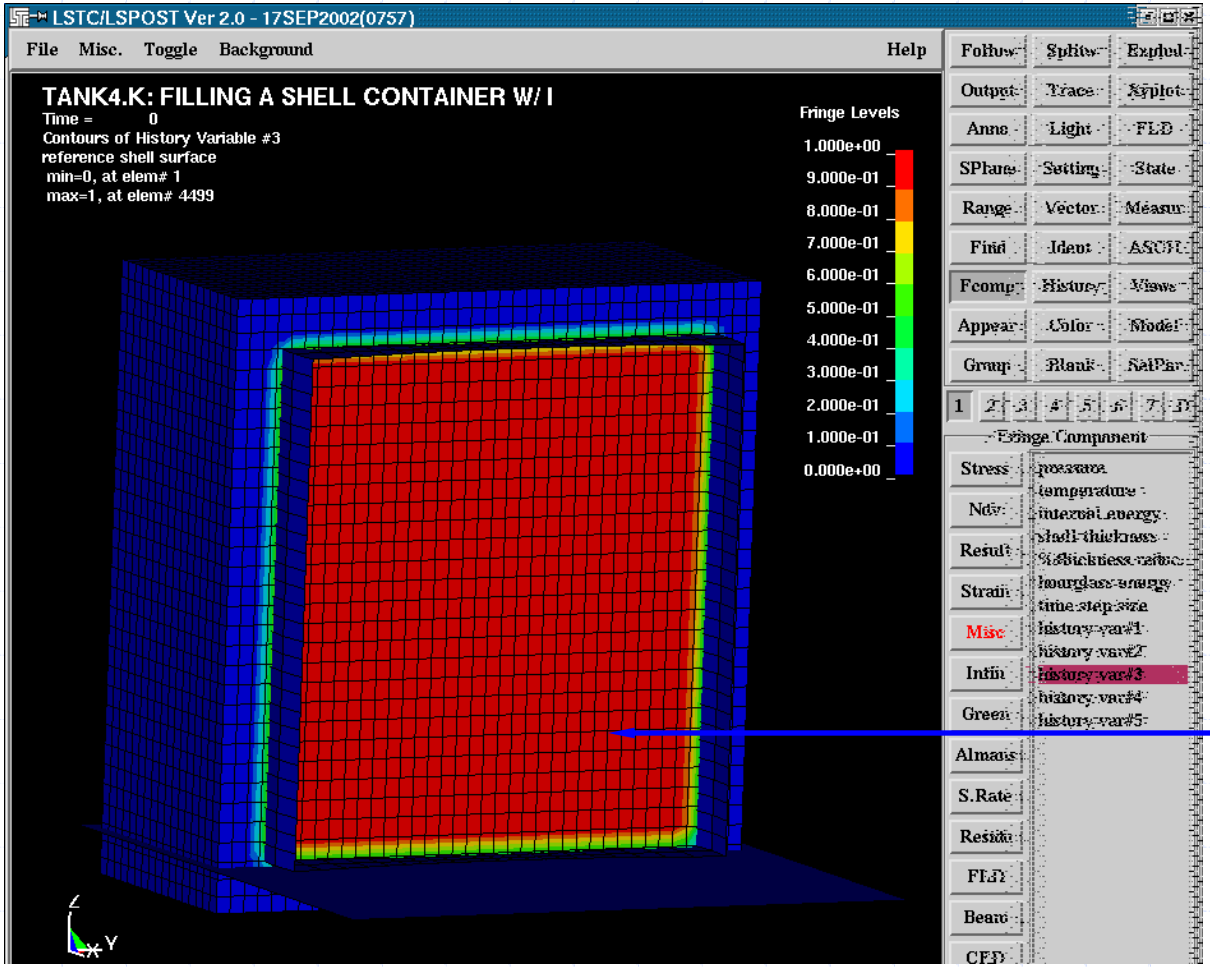
Step 1: including cards 2 & 3 – history variable 2 = AMMG1 = air volfrac contour



This is AMMG1

Volume-Filling with LS-DYNA ALE Method

Step 1: including cards 2 & 3 – history variable 3 = AMMG2 = liquid volfrac contour



During the 1st step:

Only 1 filling "action" has been carried out ➔

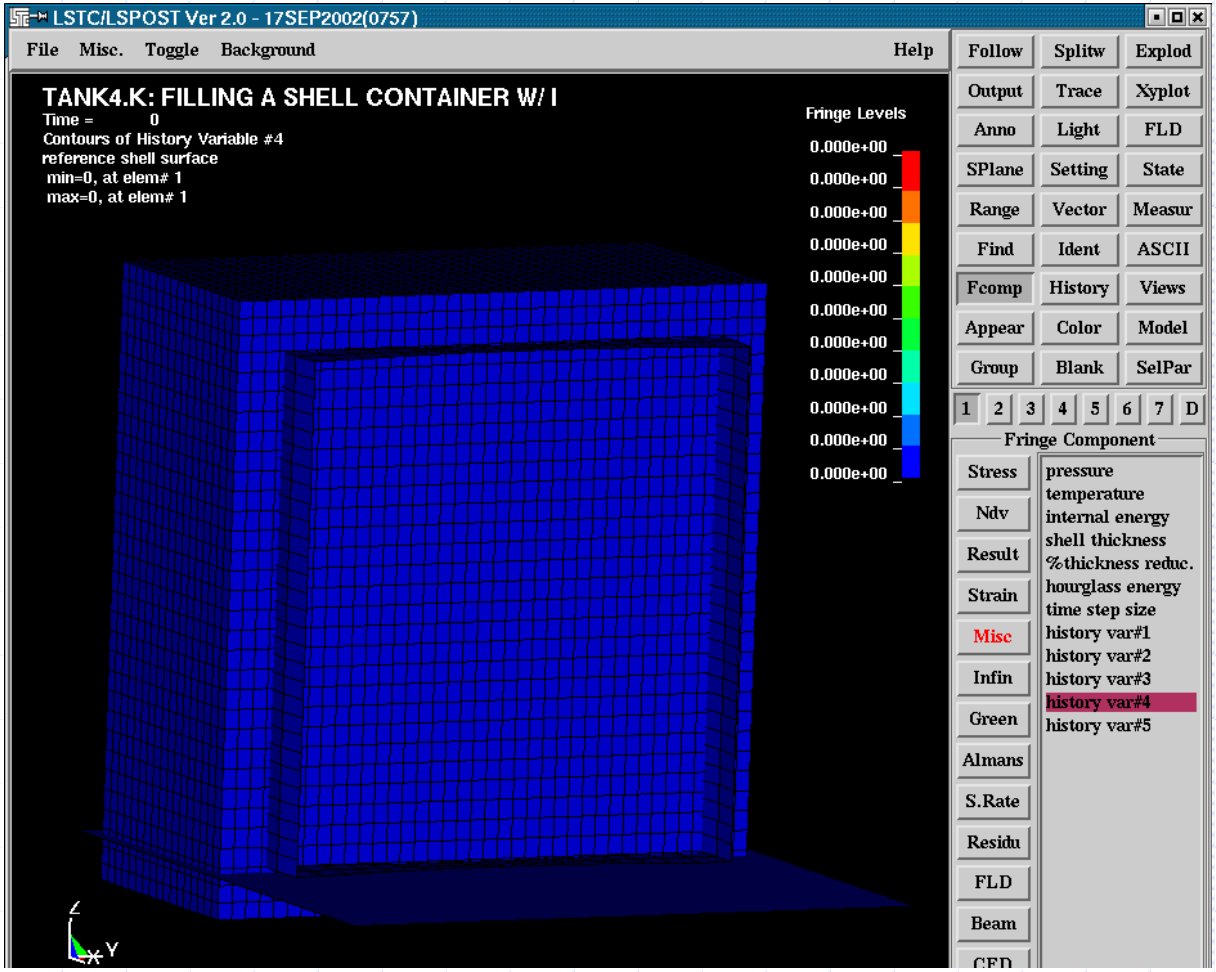
Only the liquid (AMMG2) has filled the container completely.

This is shown as the red region.

This is AMMG2

Volume-Filling with LS-DYNA ALE Method

Step 1: including cards 2 & 3 – history variable 4 = AMMG3 = vapor volfrac contour



During the 1st step:

No filling "action" has been carried out for AMMG3 (or hisvar 4)!

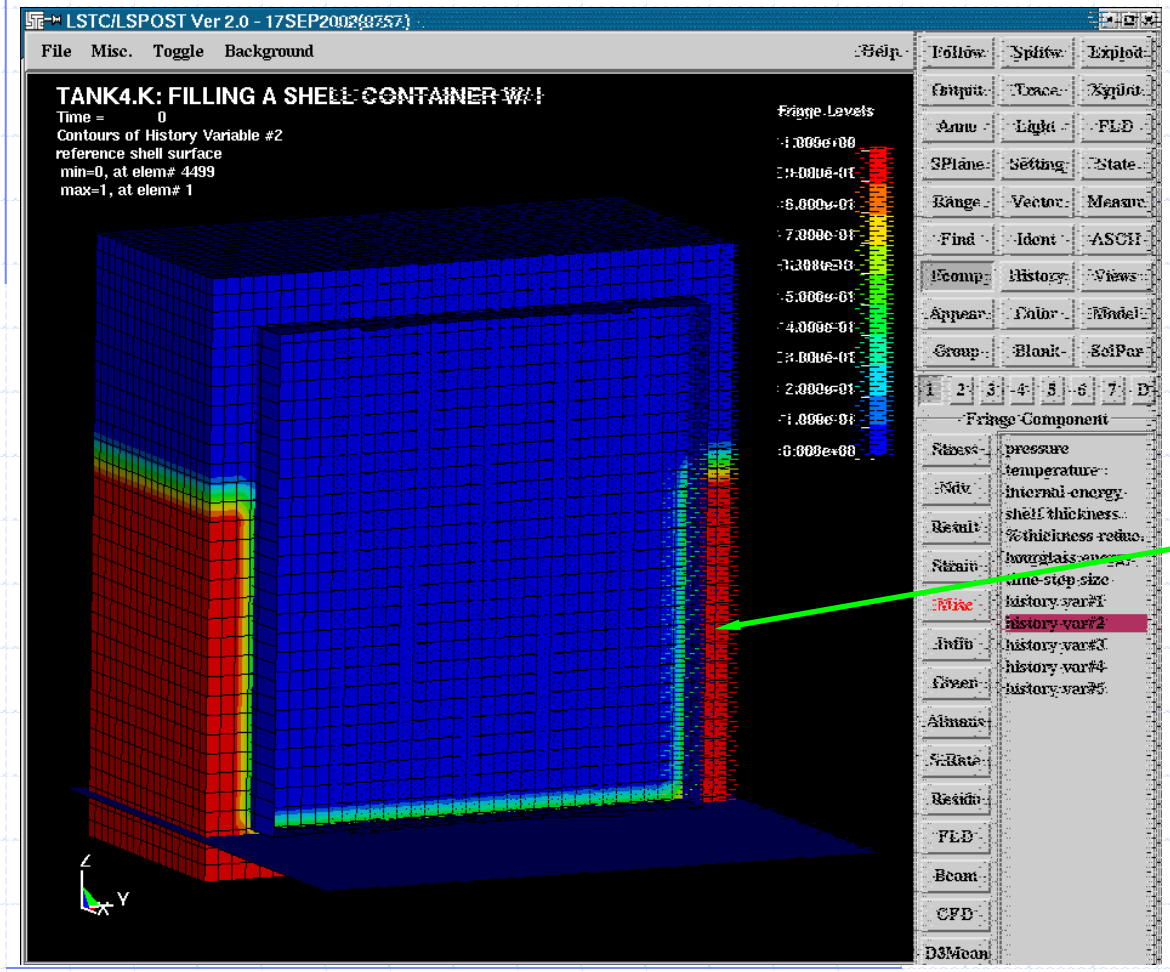
So its contour does not yet exist as shown here ➔

blue = 0 volume fraction.

The "action" of the 1st filling has done nothing about the vapor yet.

Volume-Filling with LS-DYNA ALE Method

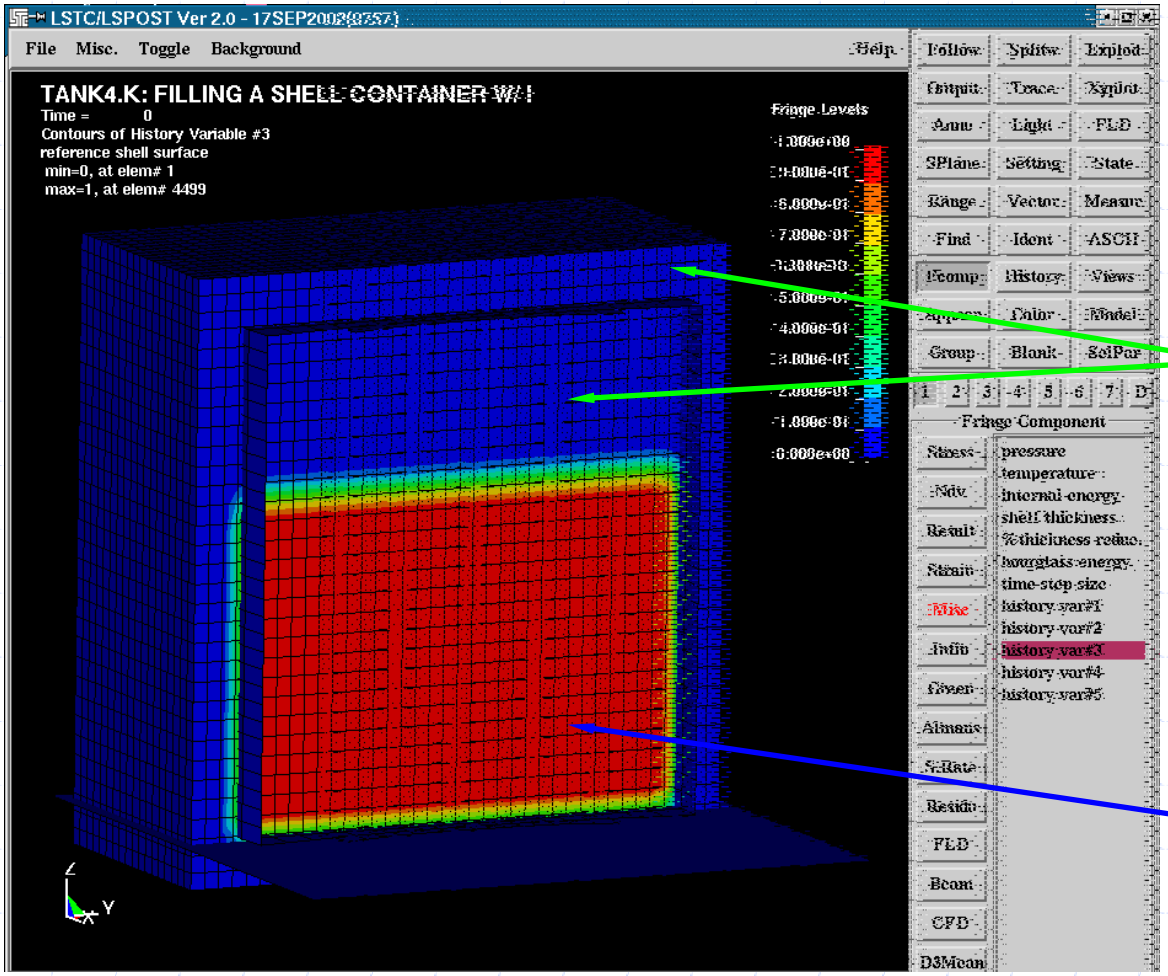
Step 1&2: including cards 2,3,4,5 – history variable 2= AMMG1= air volfrac contour



This is AMMG1 outside air

Volume-Filling with LS-DYNA ALE Method

Step 1&2: including cards 2,3,4,5 – history variable 3 = AMMG2 = liquid volfrac



During the 2nd step:

Fill all elms above a "plane" @ z=400 with AMMG3= fuel vapor both in and outside tank

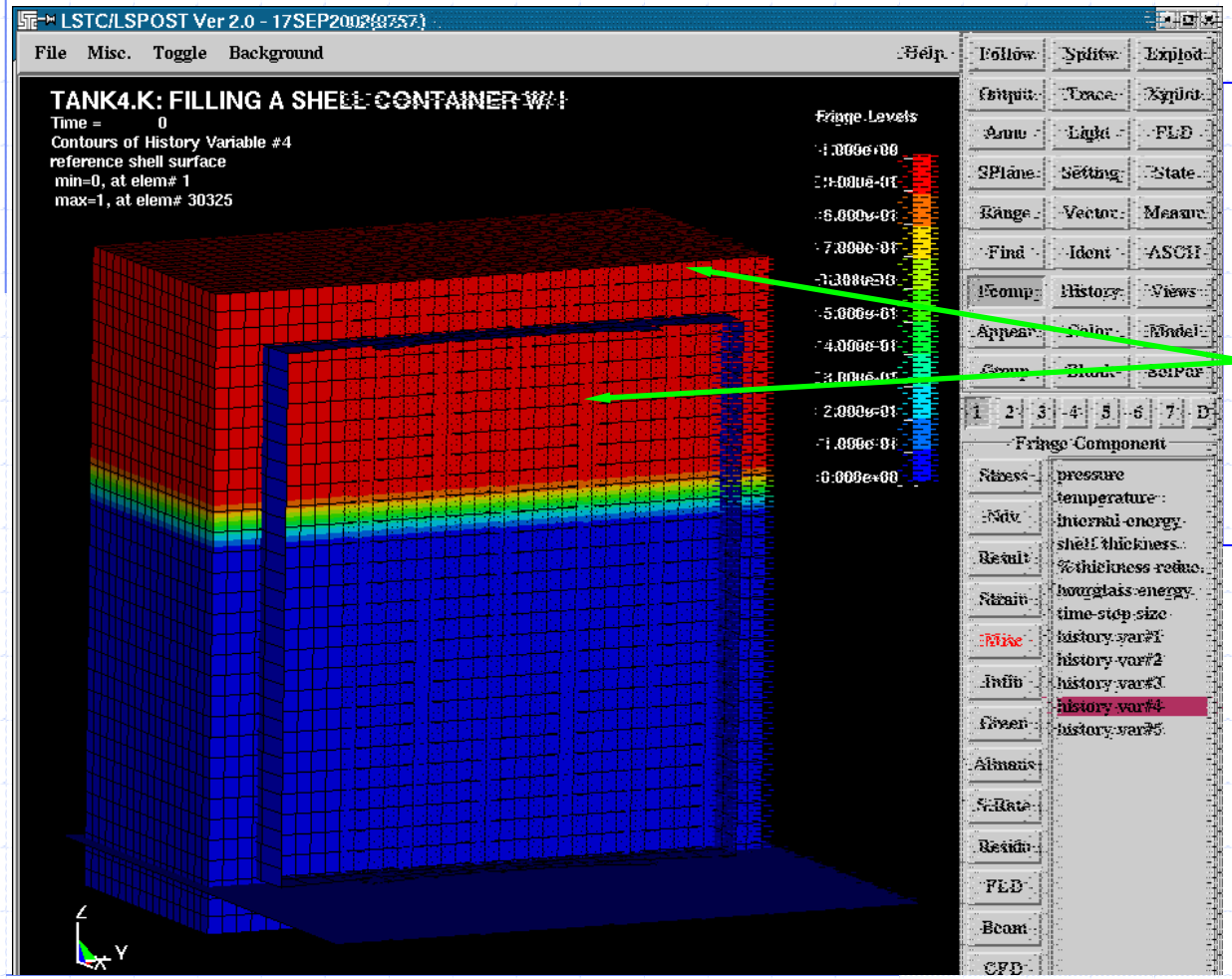
The liquid (AMMG2) now fills the container partially.

This is shown as the red region.

This is AMMG2.

Volume-Filling with LS-DYNA ALE Method

Step 1&2: including cards 2,3,4,5 – history variable 4= AMMG3 = fuel vapor volfrac



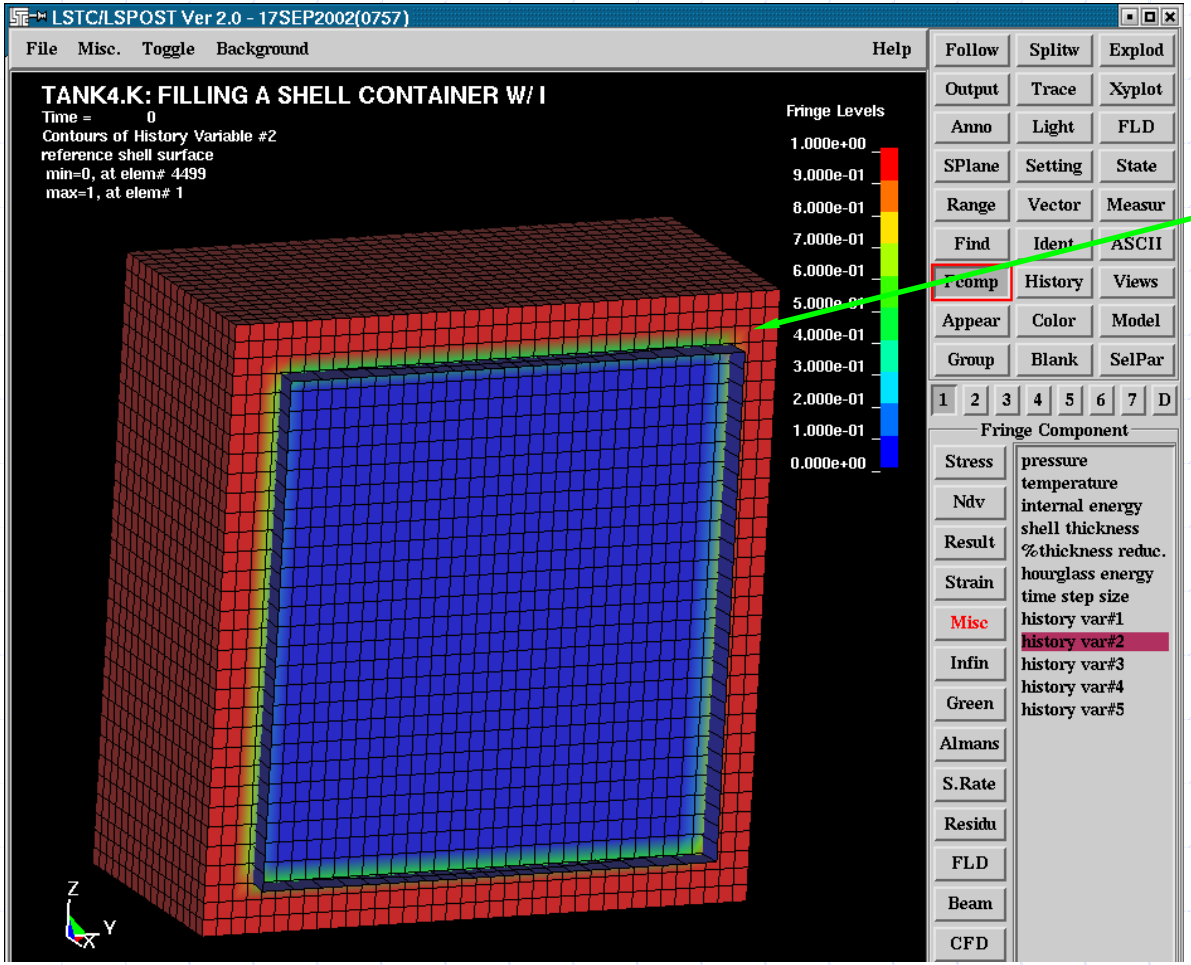
During the 2nd step:

Fill all elms above a "plane" at z=400 with AMMG3= fuel vapor both in and outside tank

This is AMMG3.

Volume-Filling with LS-DYNA ALE Method

All Steps: history variable 2= AMMG1= air volfrac contour



Final step:

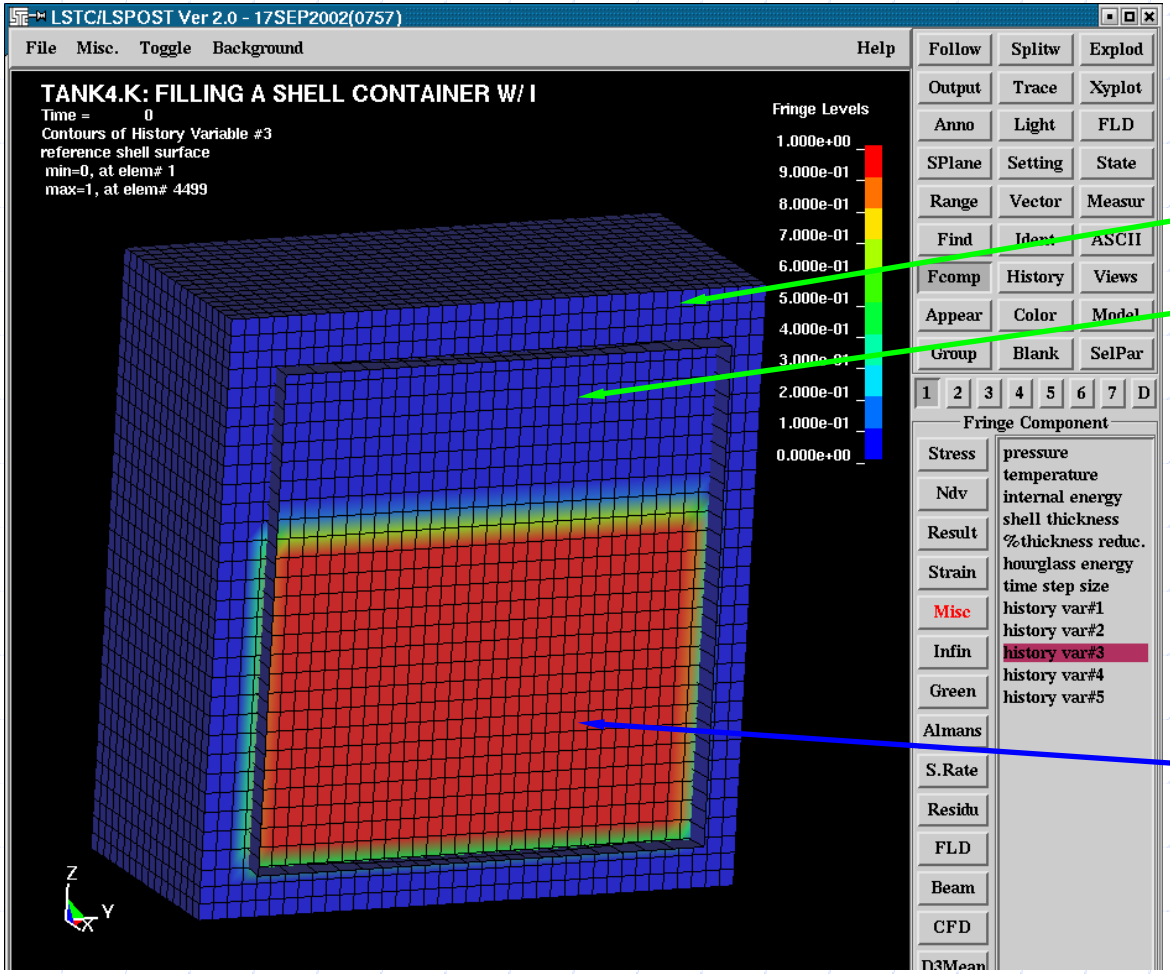
This is AMMG1 = outside air.

Step 3 fills all elms outside the container with air.

This is to correct for the action of step 2 which fills the vapor outside the container, too. (recall "fill all elms with z .ge. 400"!)

Volume-Filling with LS-DYNA ALE Method

All Steps: history variable 3 = AMMG2 = liquid volfrac



Final step:

AMMG1= outside air

AMMG3= fuel vapor

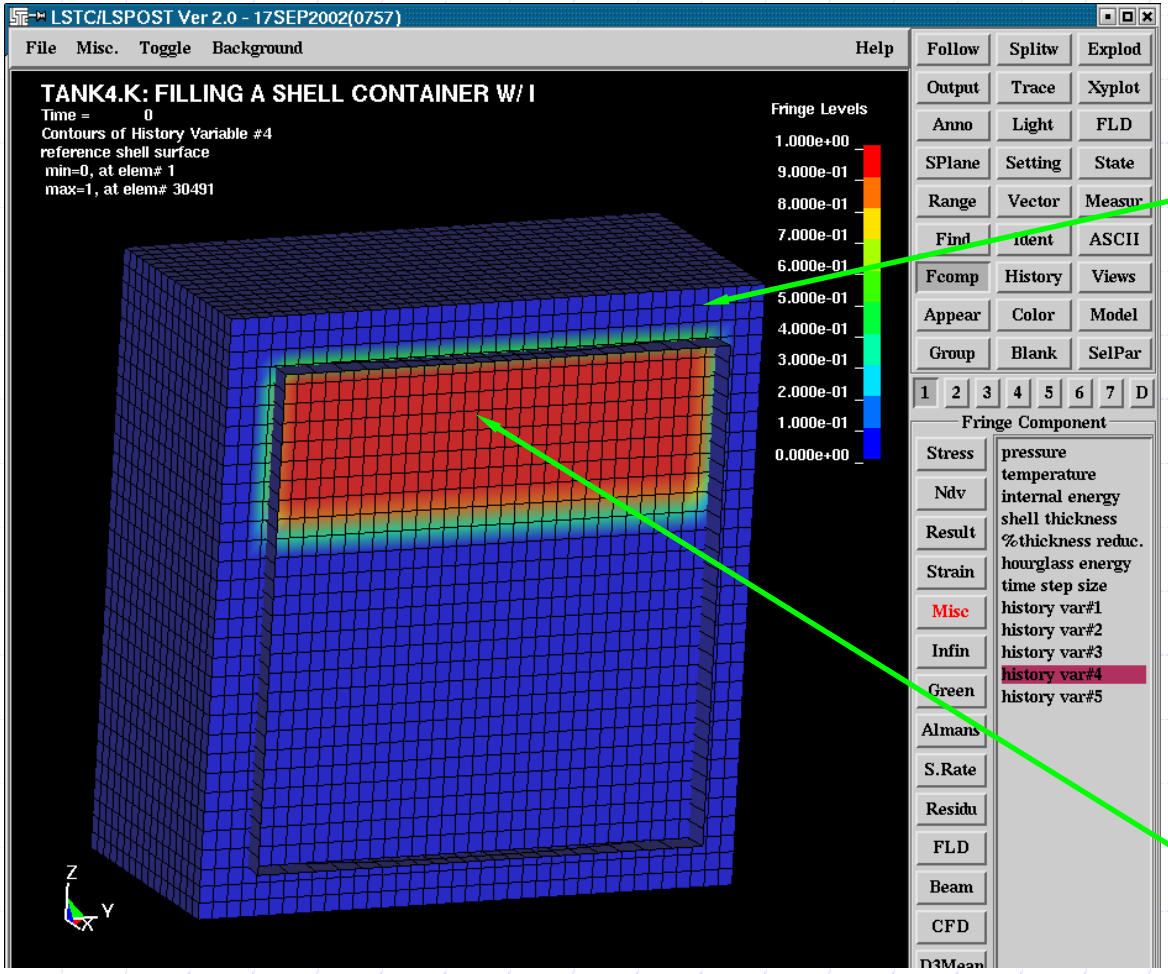
The liquid (AMMG2) now fills the container partially.

This is shown as the red region.

This is AMMG2.

Volume-Filling with LS-DYNA ALE Method

All Steps: history variable 4= AMMG3 = fuel vapor volfrac



Final step:

This is AMMG1 = outside air.

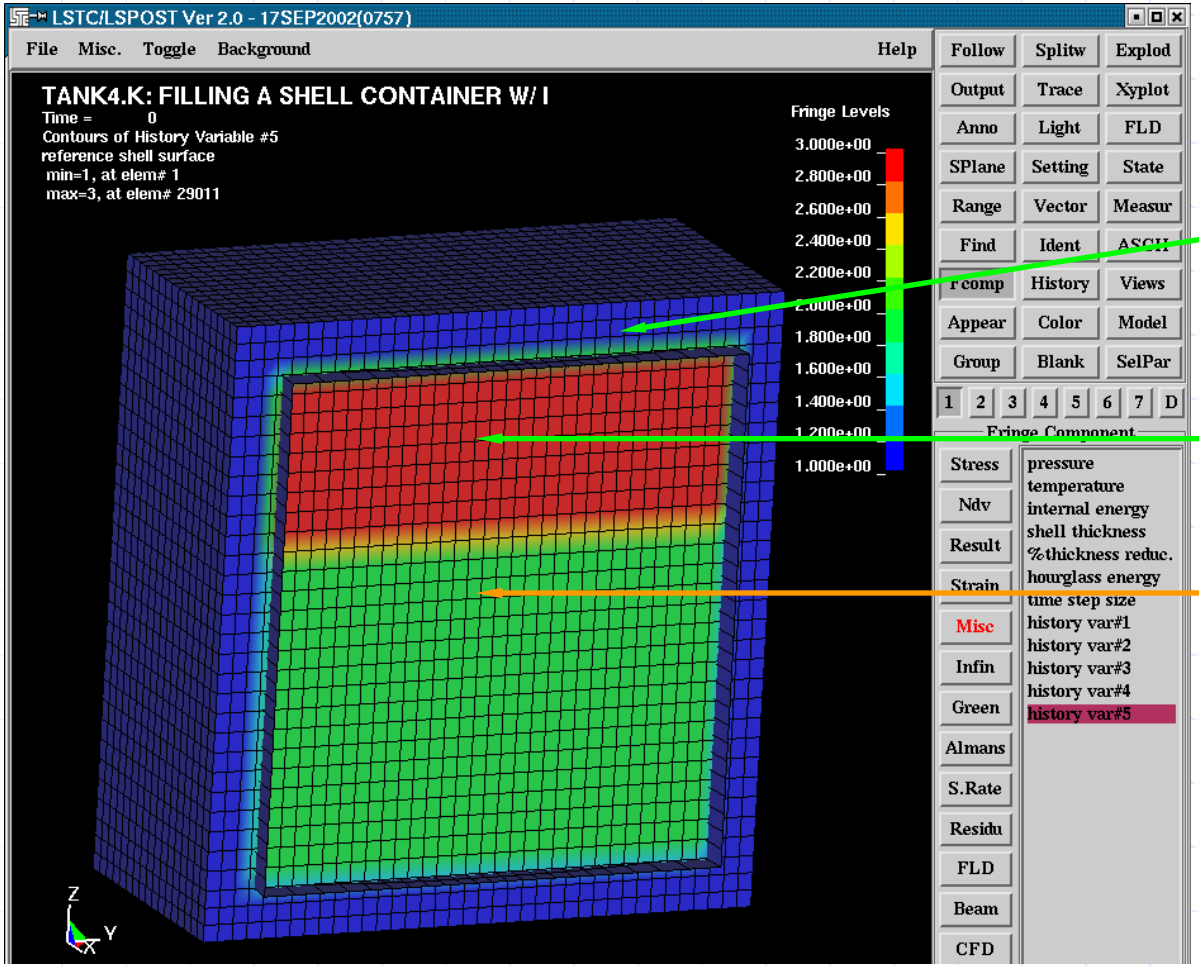
Step 3 fills all elms outside the container with air.

This is to correct for the action of step 2 which fills the vapor outside the container, too. (recall "fill all elms with z .ge. 400"!)

This is AMMG3 = fuel vapor inside tank.

Volume-Filling with LS-DYNA ALE Method

All Steps:



Final step:

This is AMMG1 = outside air.

This is AMMG3 = fuel vapor inside tank.

This is AMMG2 = liquid fuel inside tank.