LS-DYNA

Training class in
ALE and fluid-structure interaction

Lars Olovsson

LSTC

September 2006

Class contents

Day 1

- Introduction
- Lagrangian Formulation
- Single material Euler/ALE
- Multi-material Eulerian formulation
- Multi-material ALE-formulation
- Workshop

Day 2

- Fluid-structure interaction
- Some useful materials and EOS
- Workshop

Introduction

- Why numerical modeling?
- Different formulations
  - Lagrangian formulation
  - ALE-formulation
  - Multi-material Eulerian formulation
  - Multi-material ALE-formulation
- Fluid-structure interaction

Why Numerical Modeling?

Q: What's the purpose of numerical modeling in structural and fluid mechanics?

A: To predict the response of mechanical systems that are exposed to specific loads or initial conditions.

Why Numerical Modeling?

Q: How can this be achieved?

Lagrangian, Eulerian or ALE formulation

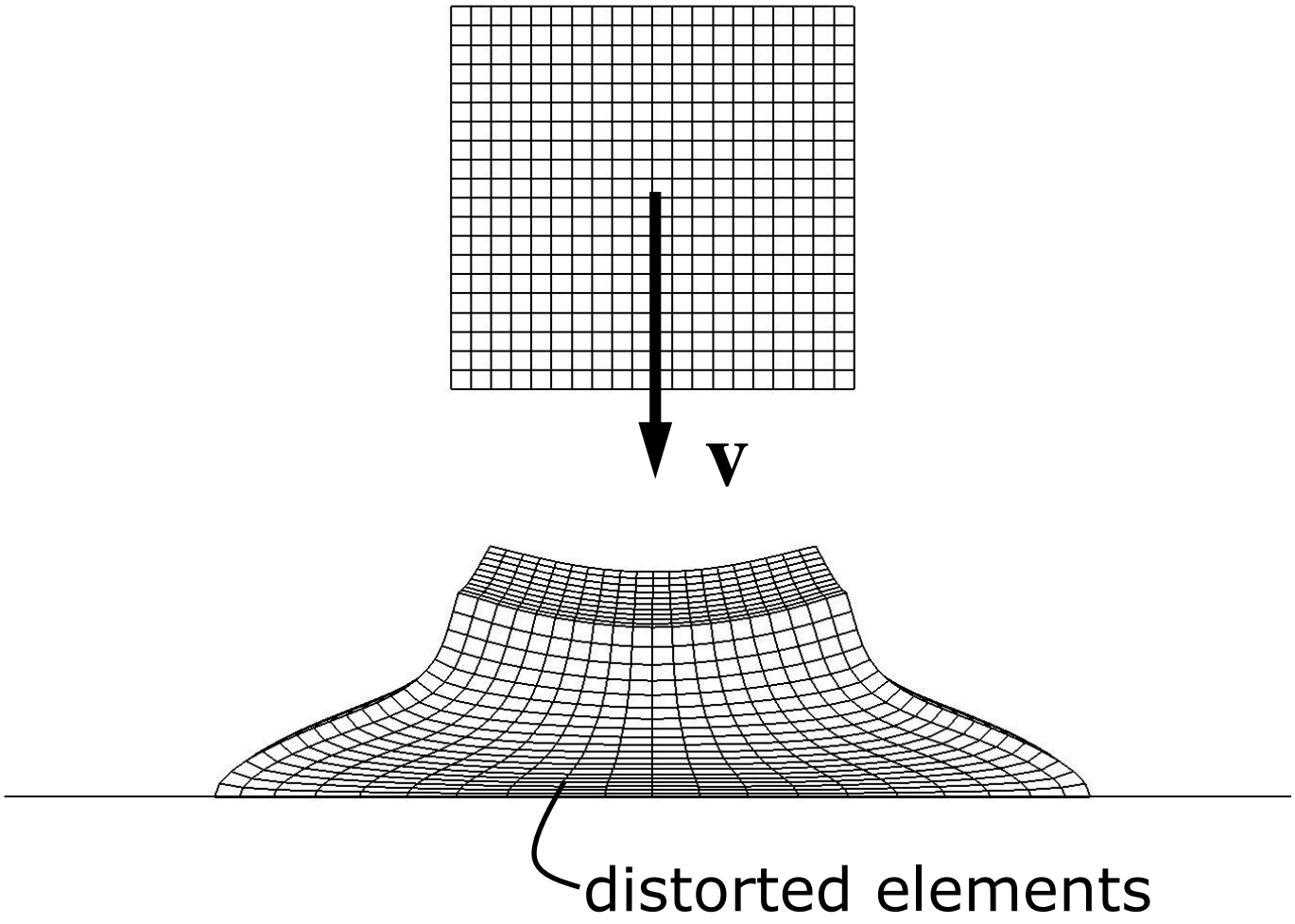
A: We first need a set of equations that realistically describe the physics of the system.

Secondly, we need to solve these equations, with appropriate boundary conditions.

Numerical solution, Finite Elements

Lagrangian formulation

The Lagrangian formulation is not suitable at very large deformations.

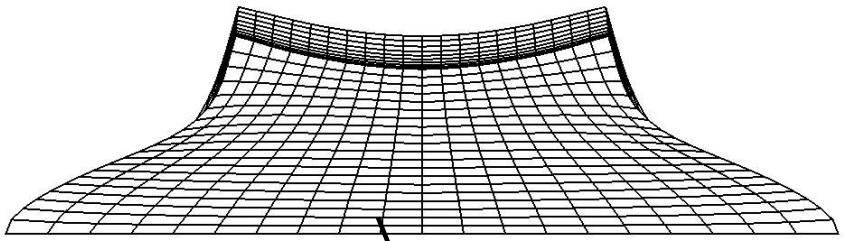


ALE formulation

The nodes don’t follow the material flow, but they are positioned to avoid badly distorted elements.

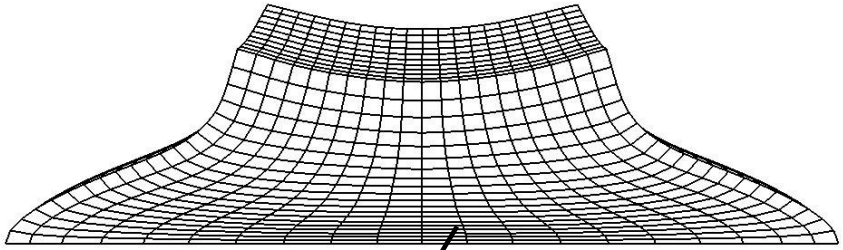
A relative motion between material and element grid leads to extra terms in the balance laws.

ALE



good

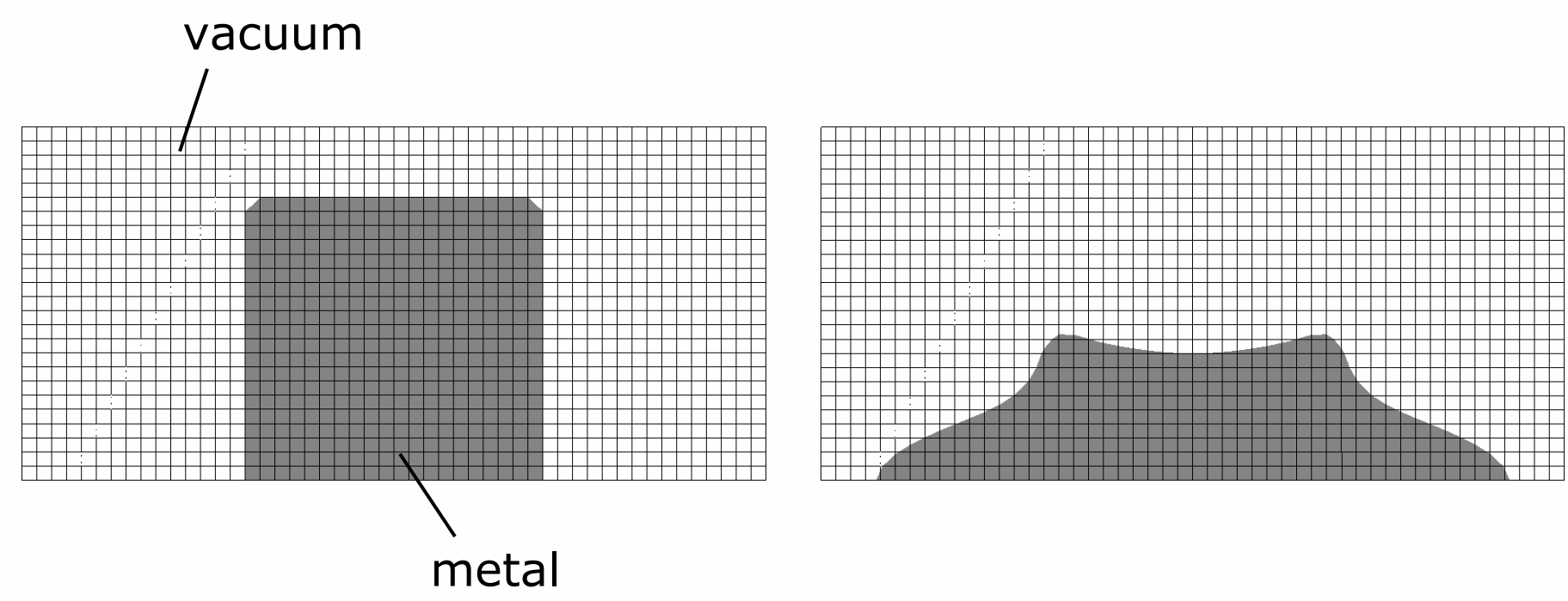
Lagrangian formulation



ugly

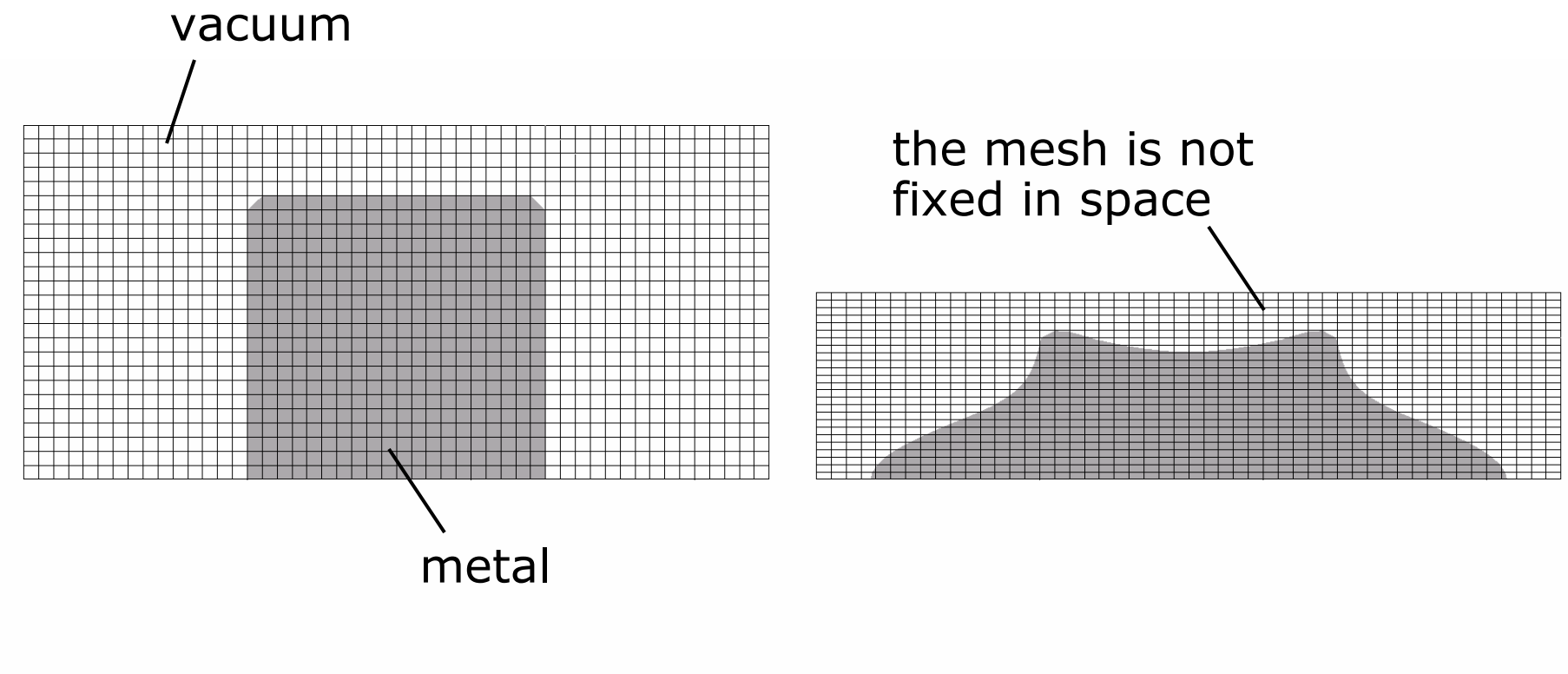
Multi-material Eulerian formulation

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



Multi-material ALE formulation

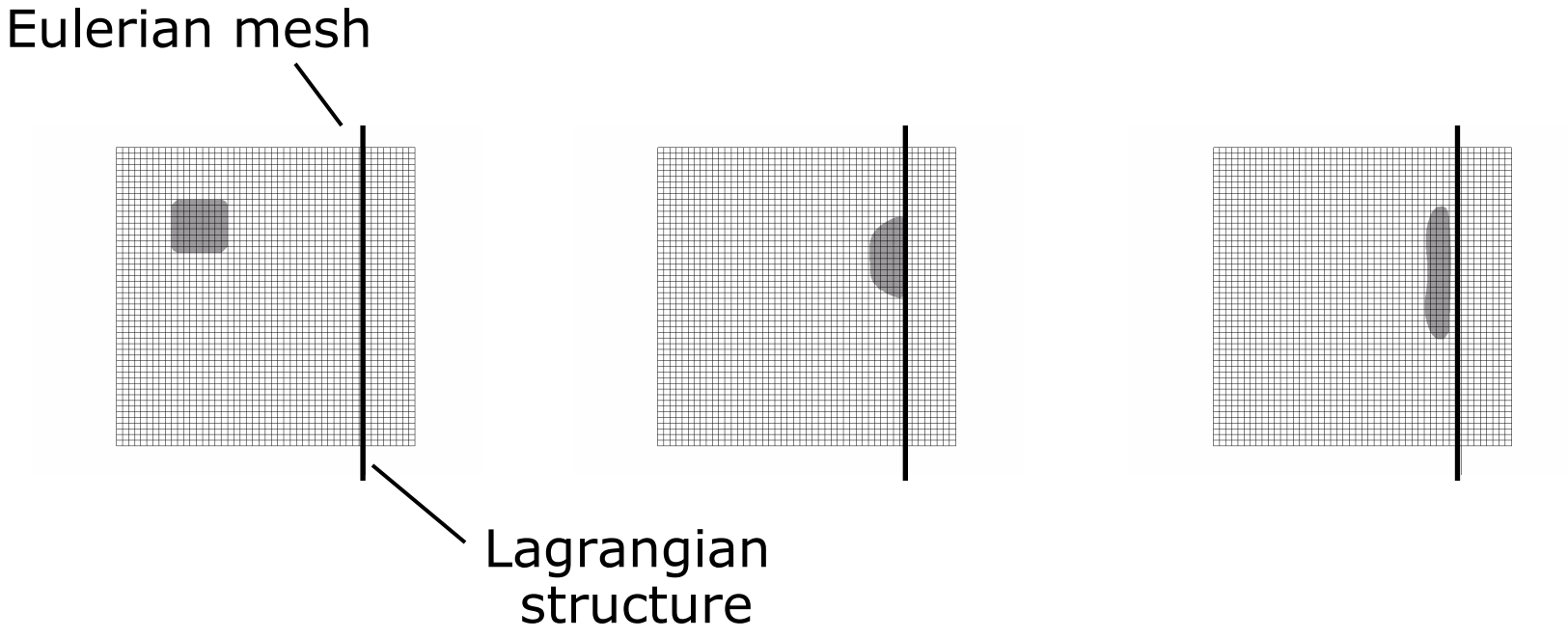
By introducing ALE in the multi-material formulation, the model can be made smaller and the numerical errors can be kept on a lower level.



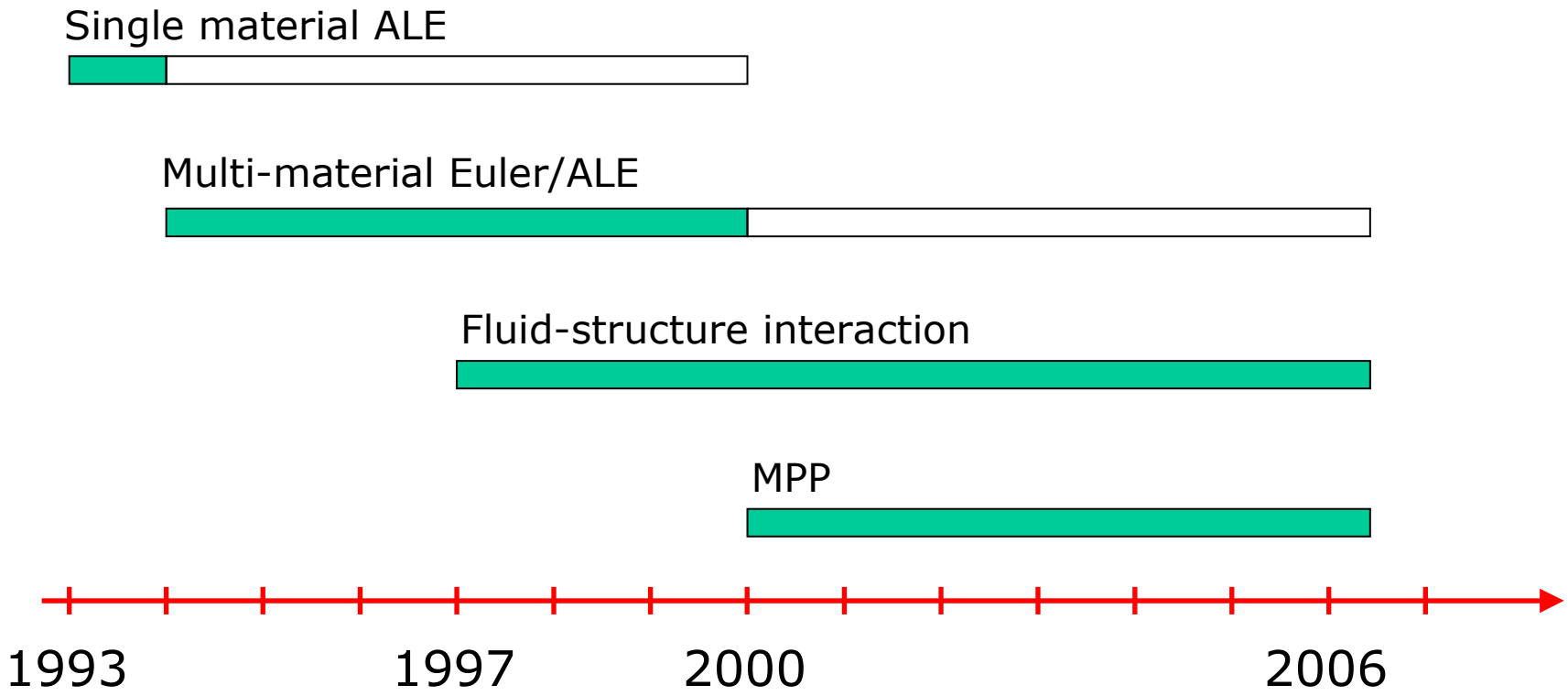
Fluid-structure interaction

It is quite often suitable to treat parts of a model as Lagrangian and other parts as Eulerian or with an ALE formulation.

A fluid-structure coupling algorithm is needed for the communication between the different parts.



ALE and FSI development in LS-DYNA



Lagrangian formulation

- Governing Equations
- Discretization in space and time
  - spatial discretization
  - time integration loop
  - internal/external forces
  - nodal accelerations
  - central difference time integration scheme
  - strain rate
  - Jaumann stress rate
  - example of constitutive model
  - bulk viscosity
- Input deck example
  - Lagrangian Taylor bar impact

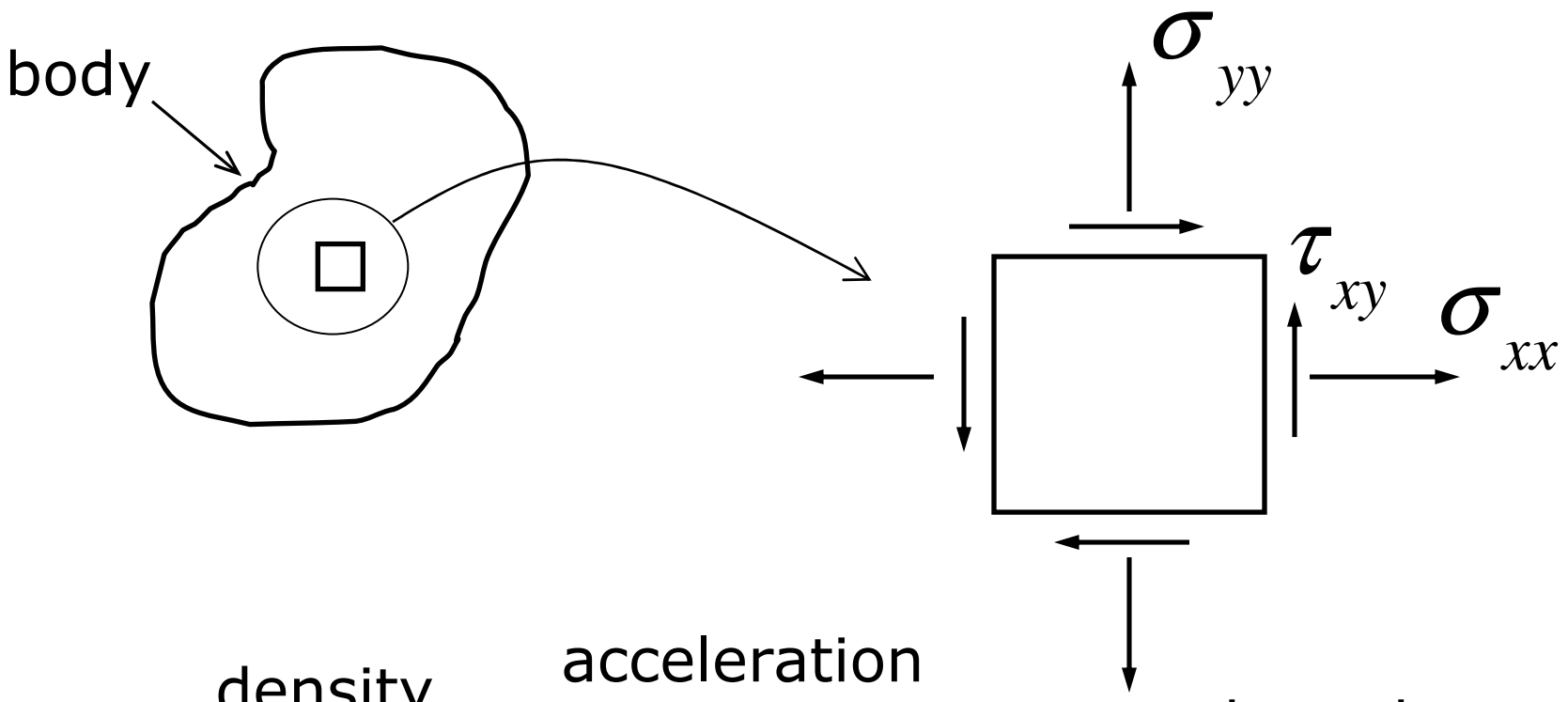
Governing equations

The idea is to solve three balance equations describing the conservation of **mass**, **momentum** and **energy**.

The solution of these equations require information about the material behavior, that is **constitutive relations**.

Governing equations

momentum balance



density

acceleration

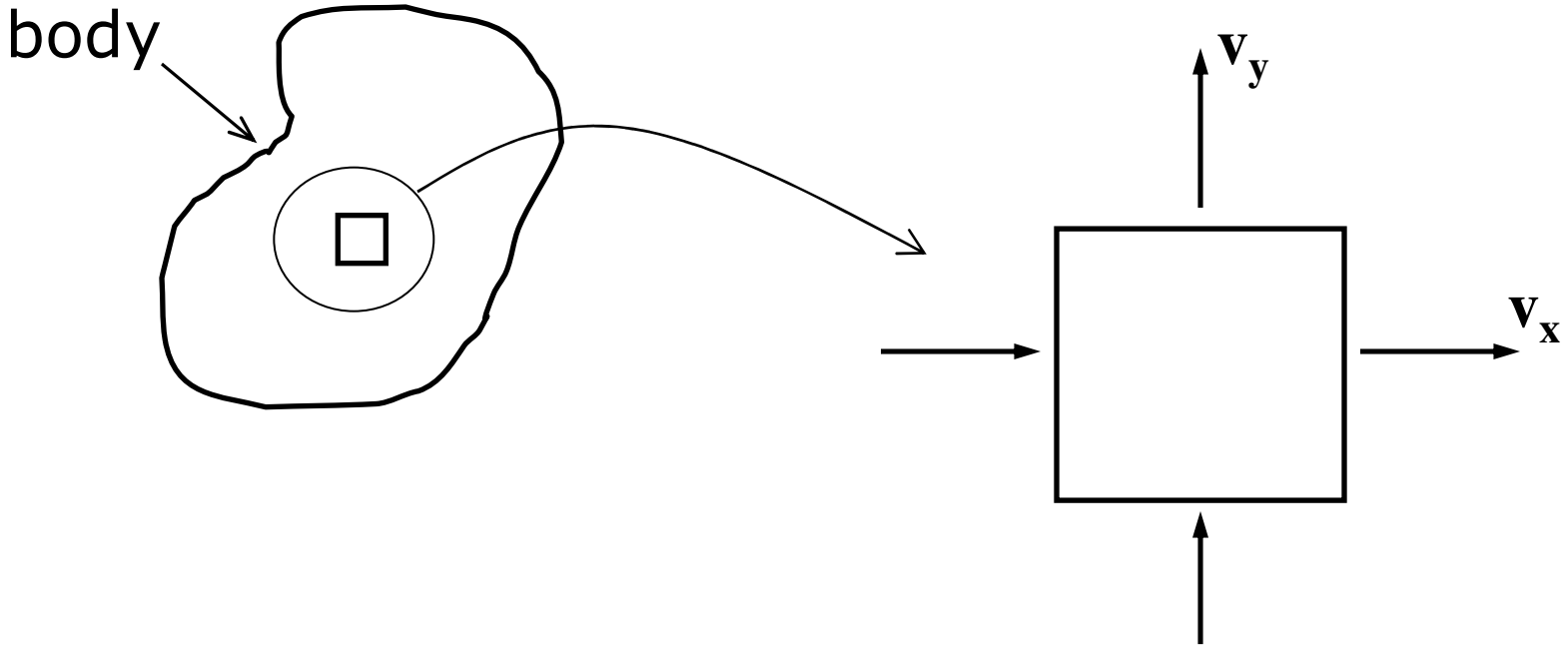
stress tensor

$$\rho \ddot{\mathbf{x}} = \rho \mathbf{g} + \operatorname{div} \sigma$$

body load

Governing equations

mass balance



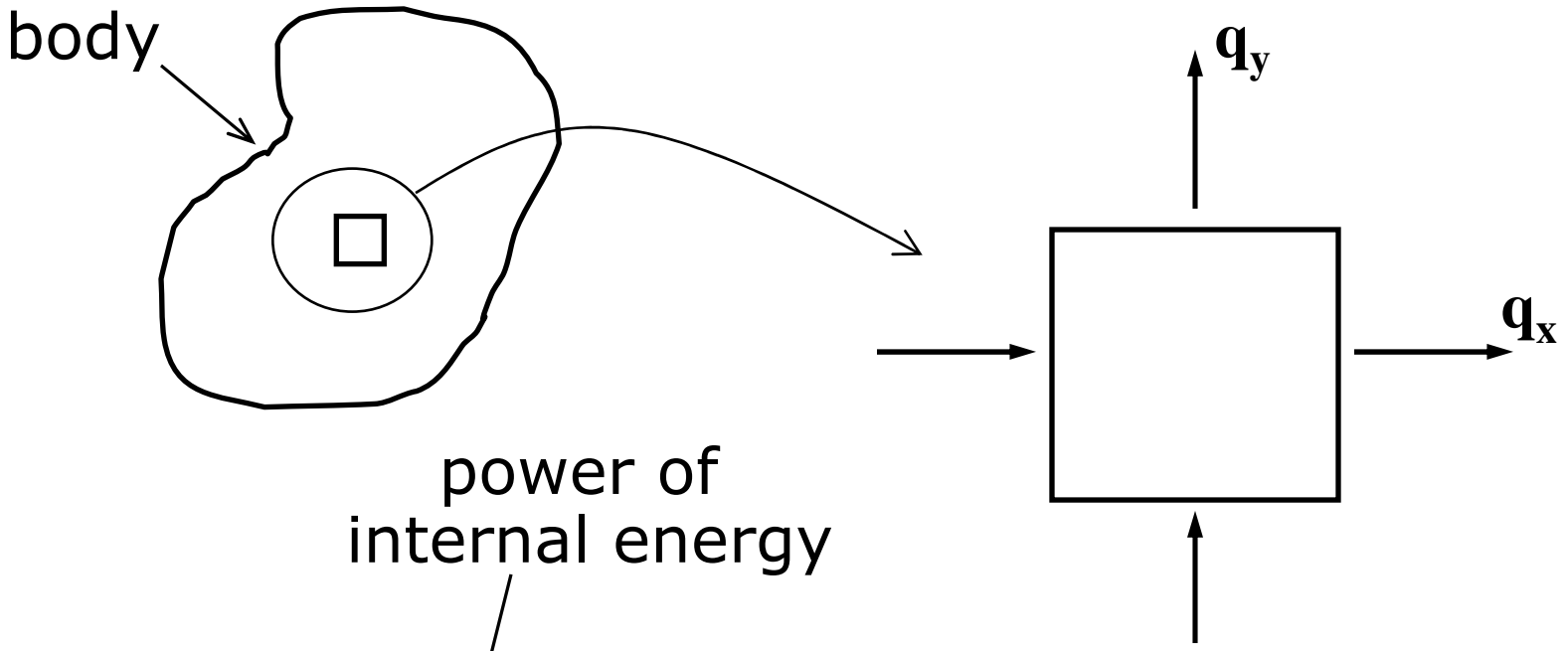
density

velocity

$$\dot{\rho} + \rho \operatorname{div} \mathbf{v} = 0$$

Governing equations

energy balance



power of internal energy

density

rate of deformation

heat flux

$$\rho \dot{u} = \sigma : \mathbf{D} + \rho r - \nabla \cdot \mathbf{q}$$

stress tensor

heat source

Discretization

Generally the conservation equations are too complex to be solved exactly.

We therefore try to find a numerical solution.

We will discretize our system in both space and in time.

Discretization in space	➡	Finite Elements
Discretization in time	➡	central difference time integration scheme

Spatial discretization

momentum equation

$$\rho\ddot{\mathbf{x}} = \rho\mathbf{g} + \operatorname{div}\boldsymbol{\sigma}$$

+ boundary conditions

weak form

$$\int_V (\rho\ddot{\mathbf{x}} - \rho\mathbf{g} - \operatorname{div}\boldsymbol{\sigma}) \cdot \boldsymbol{\Phi}\, dV = 0 \qquad \textbf{(Eq. A)}$$

+ boundary conditions

discretization and numerical integration

internal and external force vectors

$$\mathbf{M}\ddot{\mathbf{d}} = \mathbf{F}_i + \mathbf{F}_e$$

mass matrix

nodal accelerations

Spatial discretization

energy equation

$$\rho \dot{u} = \sigma : \mathbf{D} + \rho r - \nabla \cdot \mathbf{q}$$

+ boundary conditions

weak form

$$\int_V (\rho \dot{u} - \sigma : \mathbf{D} - \rho r + \nabla \cdot \mathbf{q}) \cdot \varphi \, \mathrm{d}V = 0$$

+ boundary conditions

discretization and numerical integration

internal and external thermal force vectors

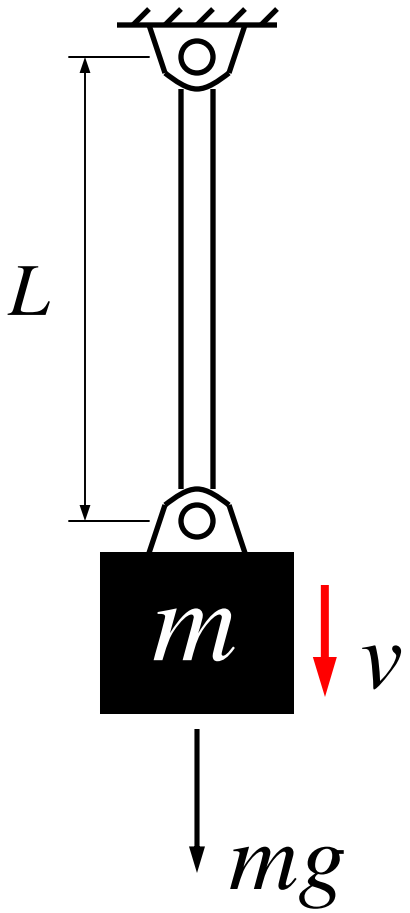
$$\mathbf{M}^{\theta} \dot{\theta} = \mathbf{F}_{\mathbf{i}}^{\theta} + \mathbf{F}_{\mathbf{e}}^{\theta}$$

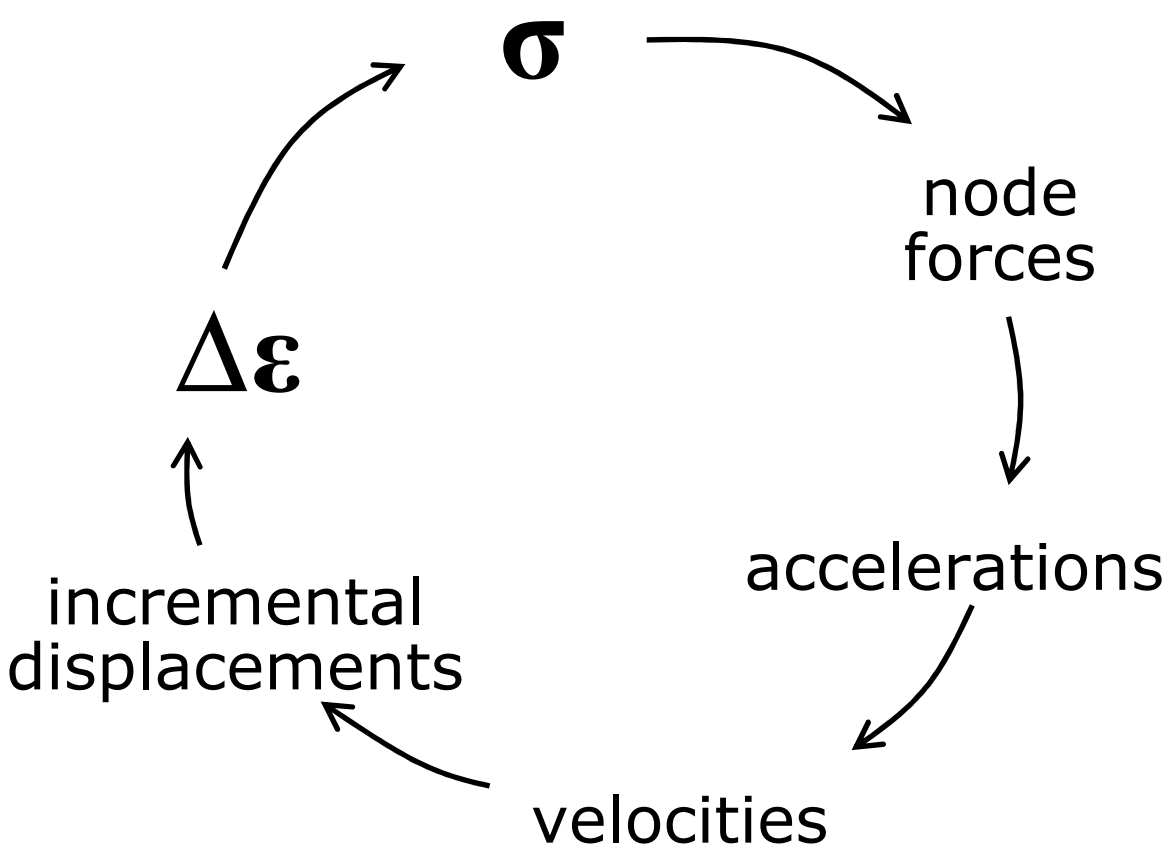
heat capacity matrix

temperature

Time integration loop

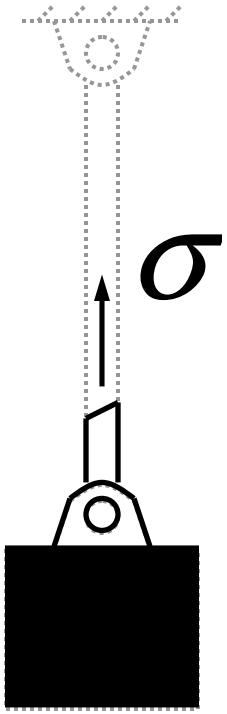
momentum equation

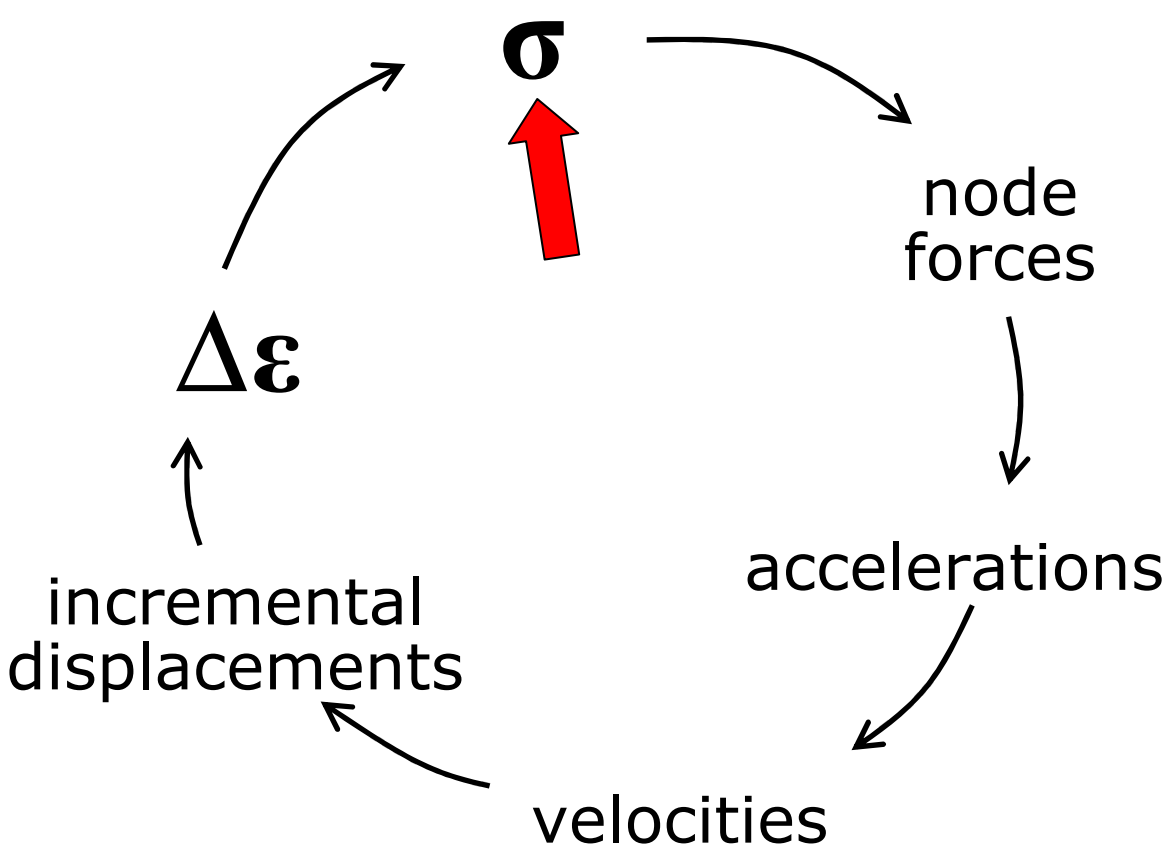




Time integration loop

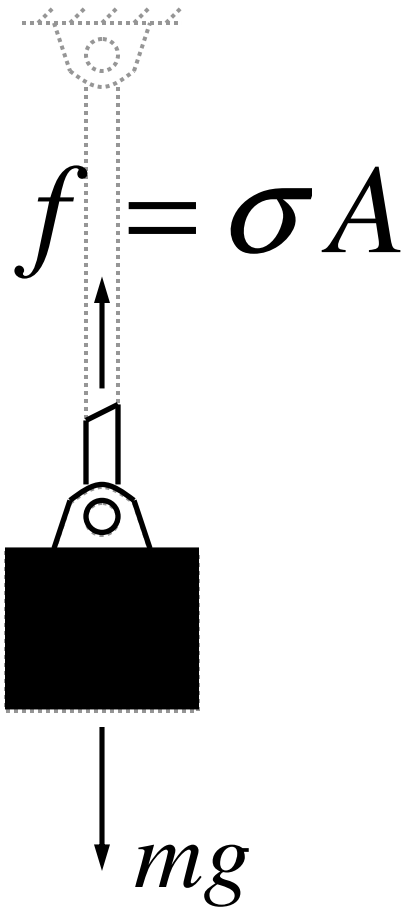
momentum equation

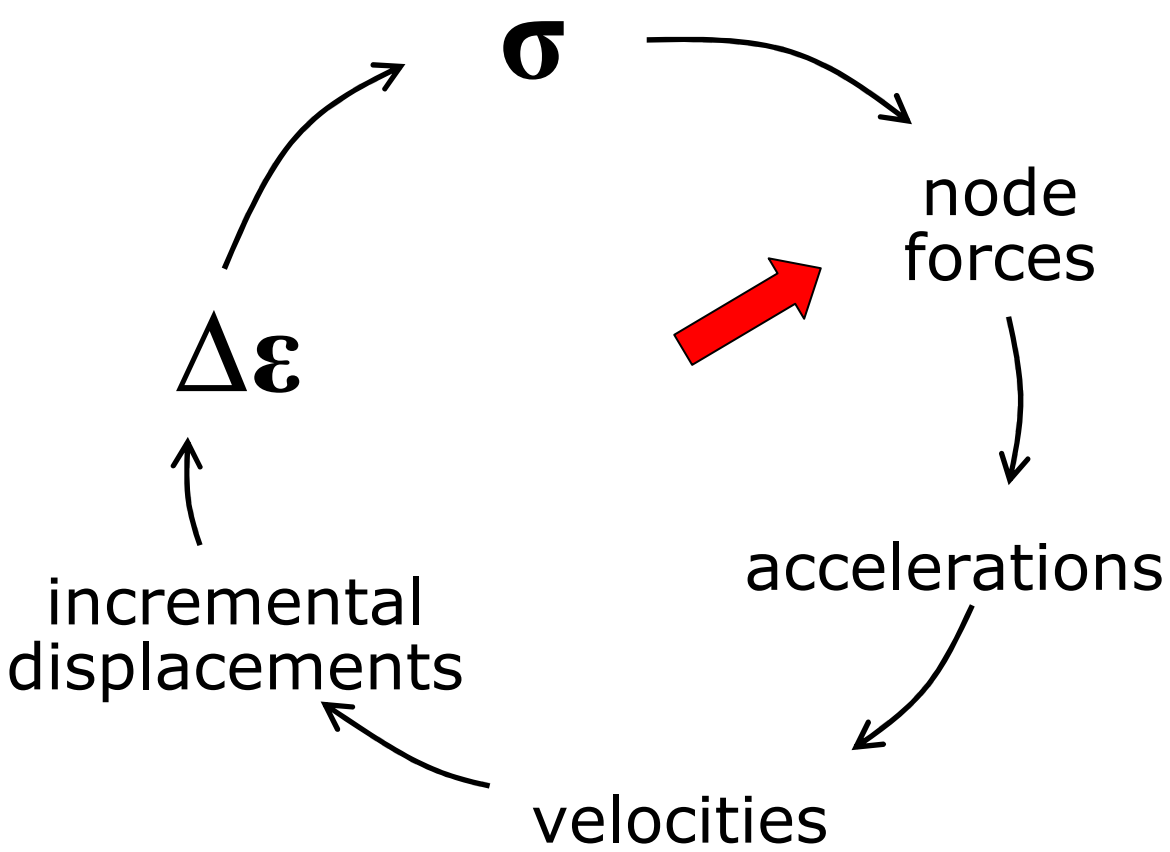




Time integration loop

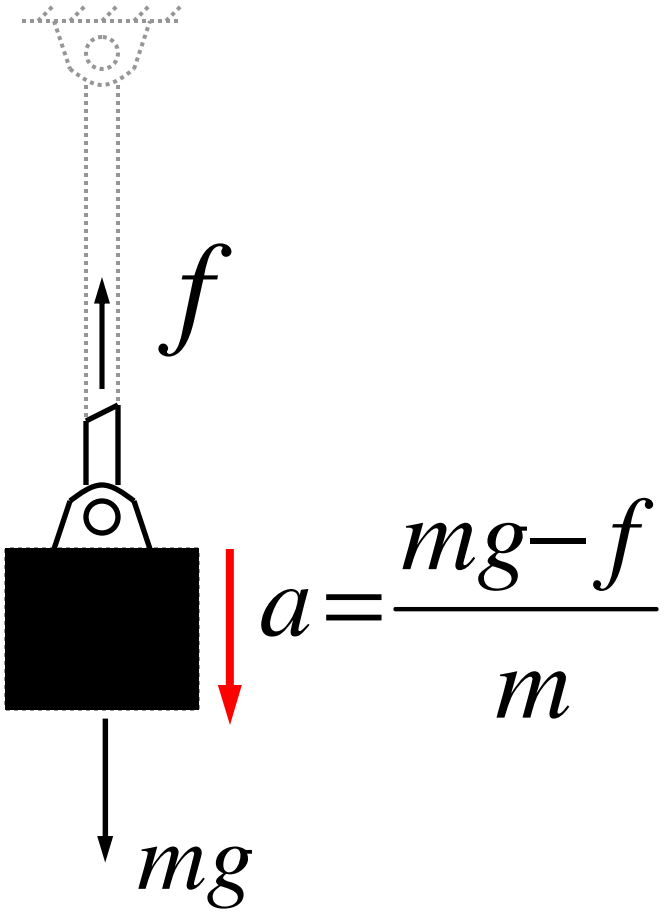
momentum equation



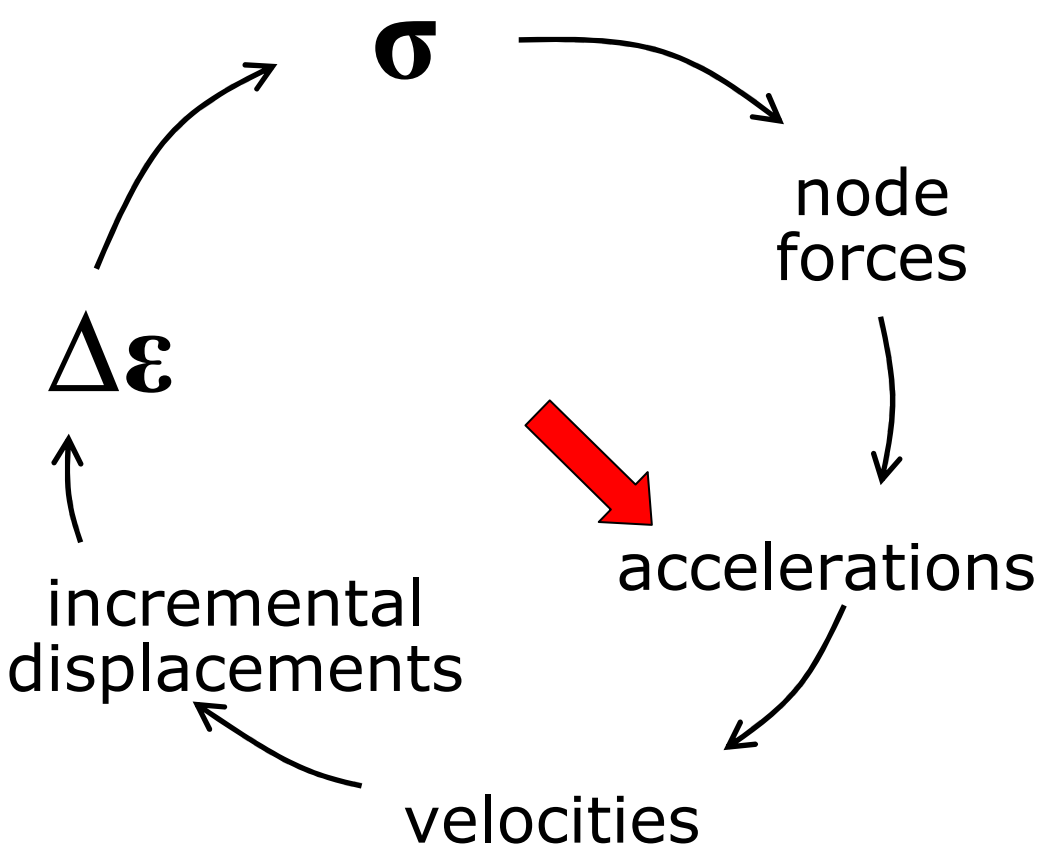


Time integration loop

momentum equation

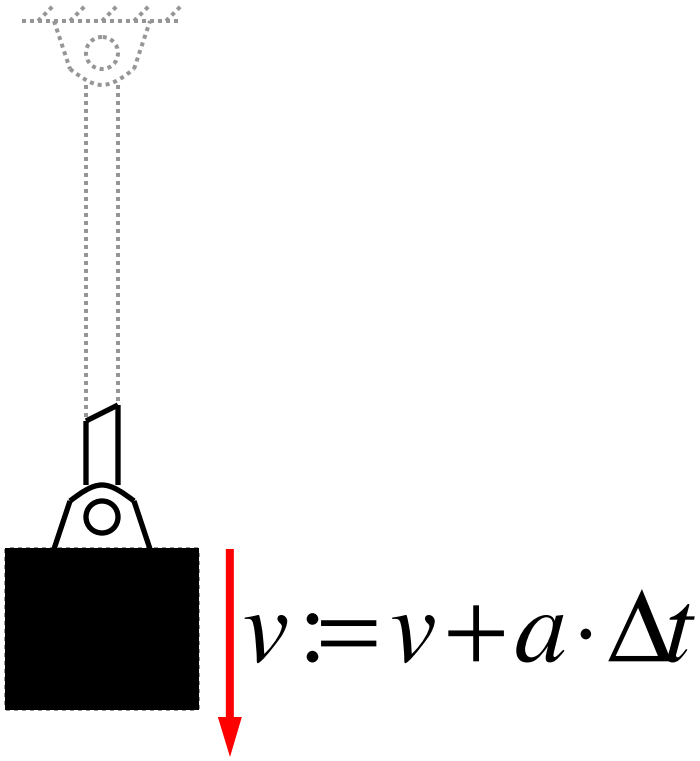


$$a = \frac{mg - f}{m}$$

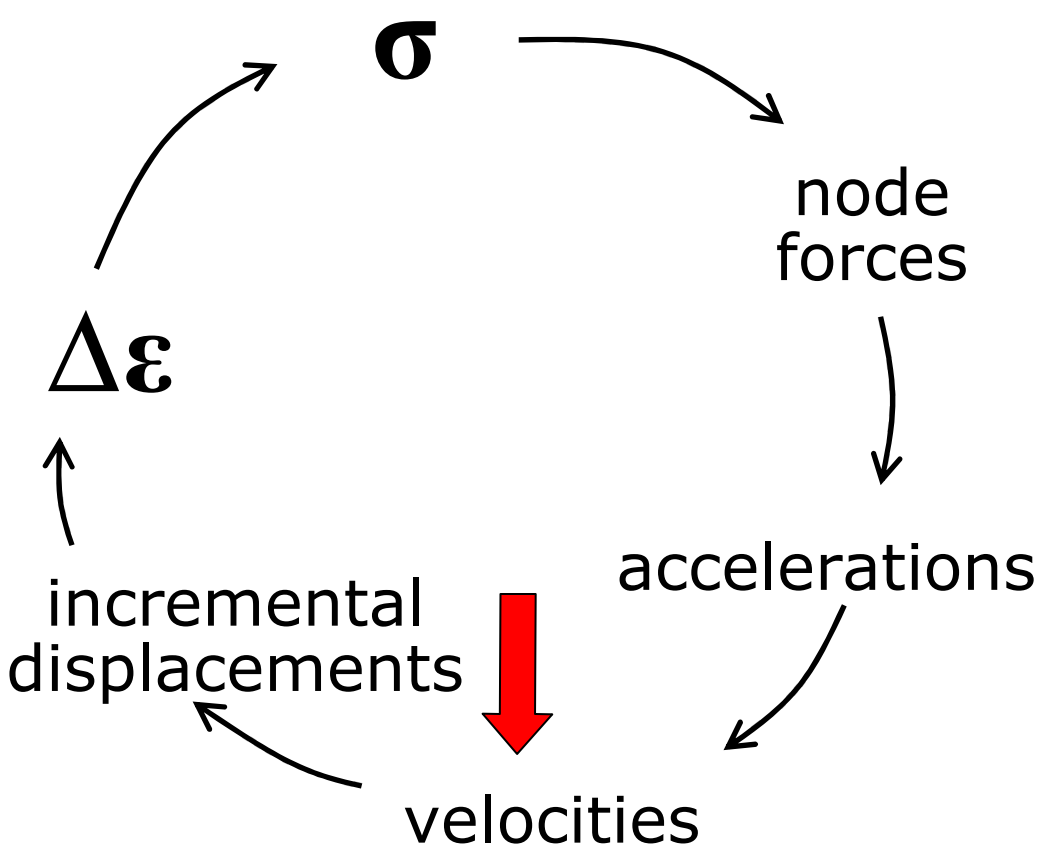


Time integration loop

momentum equation

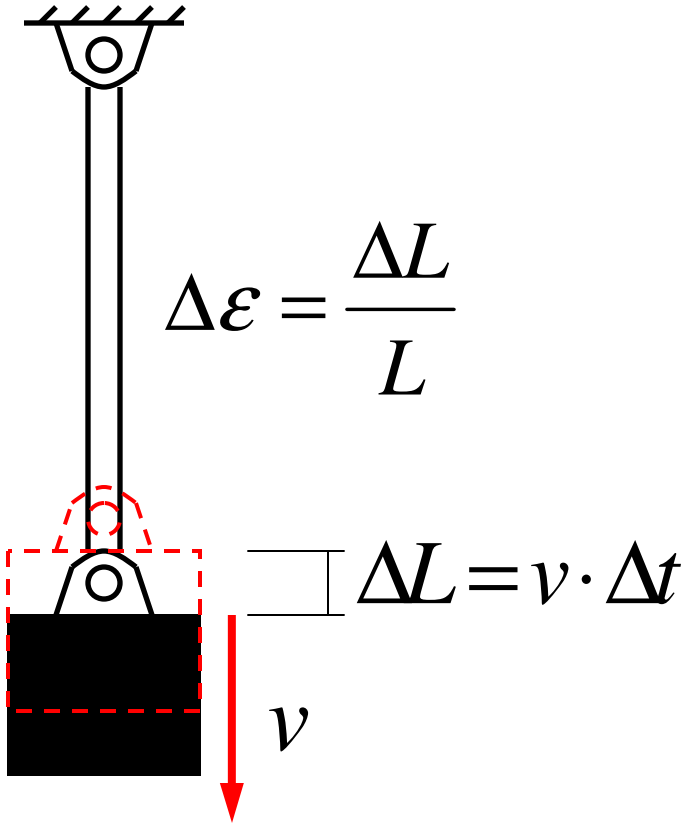


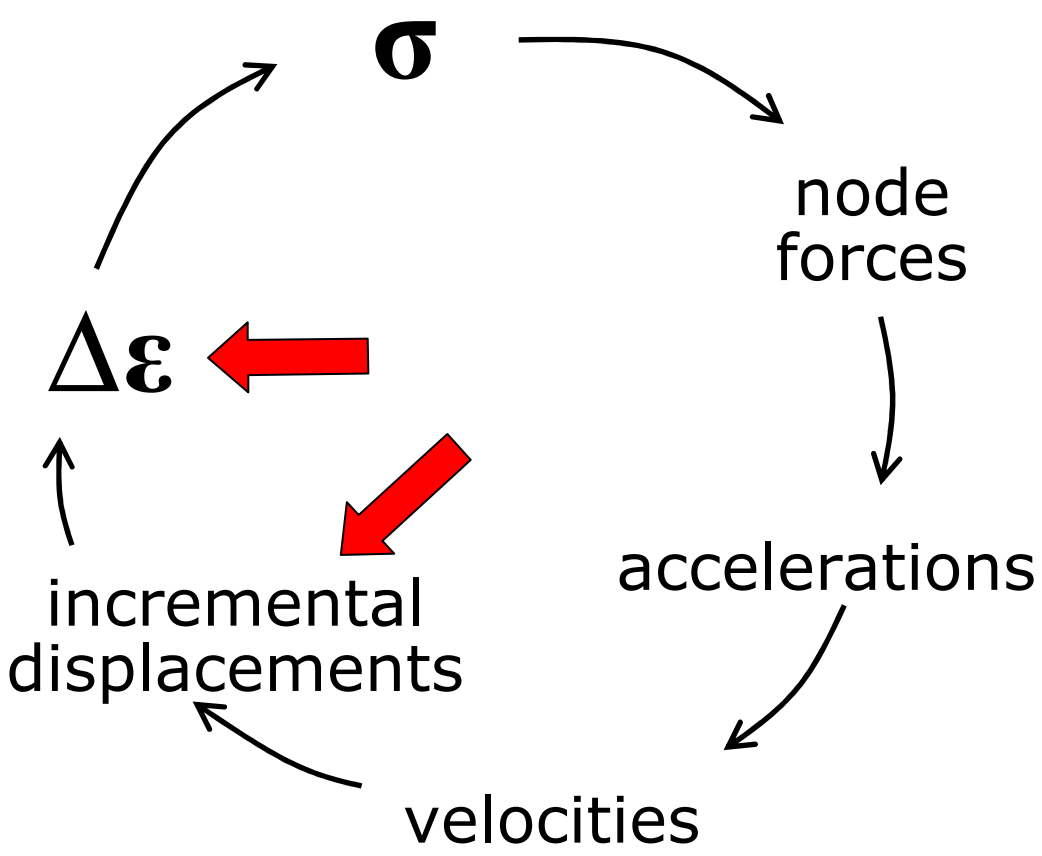
$$v := v + a \cdot \Delta t$$



Time integration loop

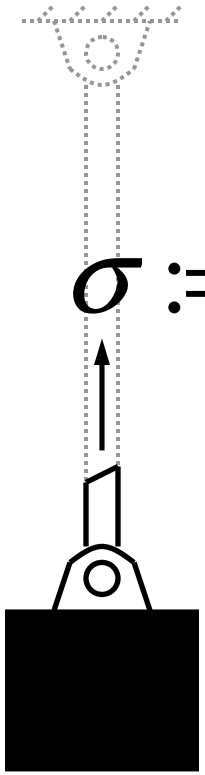
momentum equation



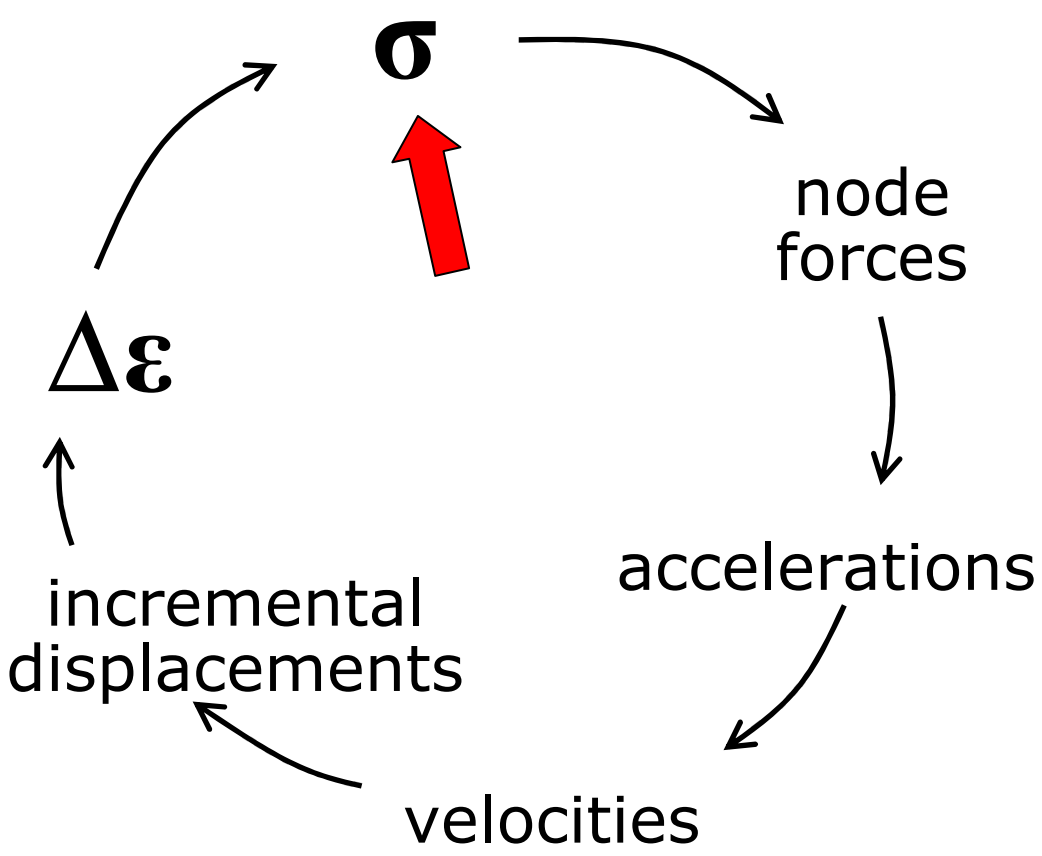


Time integration loop

momentum equation



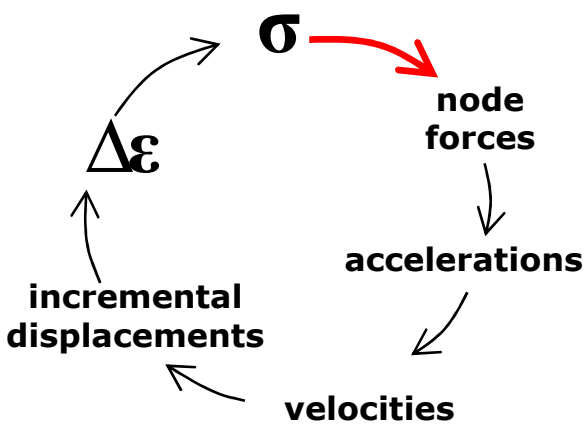
$$\sigma := \sigma + E \cdot \Delta\varepsilon$$



Internal forces

The internal forces come from a partial integration of **(Eq. A)**.

$$\int_V \operatorname{div}\boldsymbol{\sigma}\cdot\boldsymbol{\Phi}\,dV = \int_B (\sigma\mathbf{n})\cdot\boldsymbol{\Phi}\,dB - \underbrace{\int_V \boldsymbol{\sigma} : (\nabla\boldsymbol{\Phi})\,dV}_{\mathbf{F}_i}$$



Internal forces

$\mathbf{F}_i$ is the sum of internal forces for each separate element. One has, for one element.

$$\mathbf{f}_i^e = \int_{V^e} \mathbf{B}^t \overline{\boldsymbol{\sigma}}\, dV \qquad \textbf{(Eq. B)}$$

$\mathbf{B}$ contains the derivatives of the shape functions and $\overline{\boldsymbol{\sigma}}$ is a vector with the 6 stress components.

Internal forces

(Eq. B) is not solved exactly, but is estimated as

$$\mathbf{f}_i^e = \mathbf{B}^t \bar{\boldsymbol{\sigma}} \mathbf{V}^e \quad (\xi_k = 0)$$

That is, the integral is approximated as the value of the function at the center of the element, times the element volume.

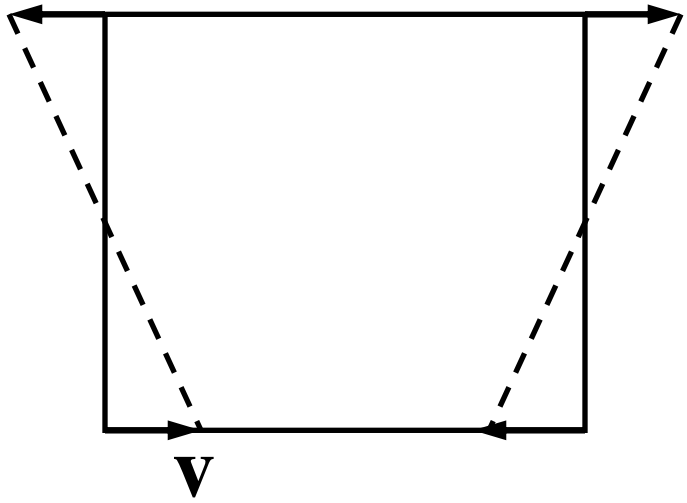
This is computationally efficient, but it leads to zero energy modes, referred to as hourglassing.

Internal forces
hourglass control

Artificial forces are applied to minimize the development of zero-energy modes. The default hourglass control is viscosity based.

The artificial forces are proportional to the nodal velocities, **v**.

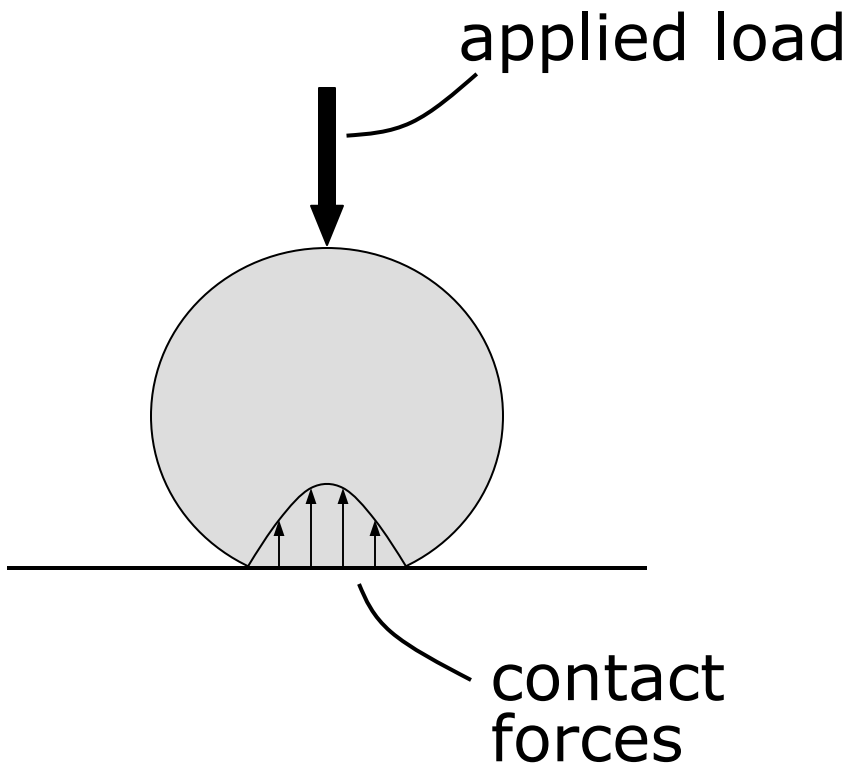
typical zero-energy deformation mode



Attention! Turn off the hourglass control when dealing with gases and fluids. It is only motivated for materials with a shear stiffness.

External forces

The external force vector consists of contributions from body loads (such as gravity), contact forces, reaction forces and other applied loads.

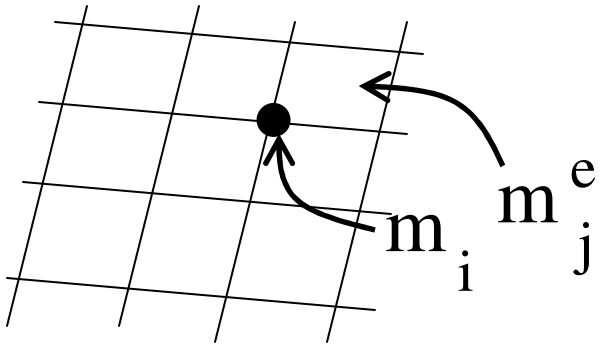


Node accelerations

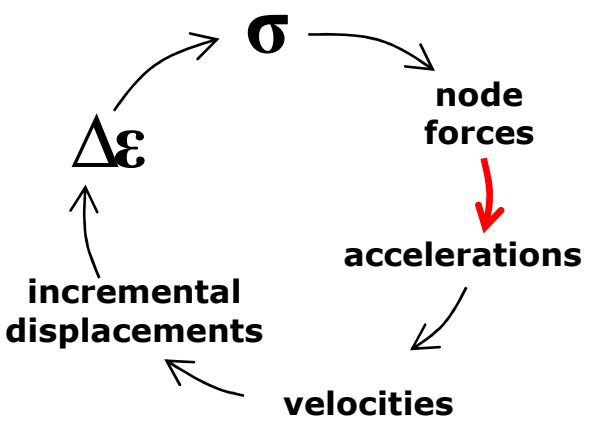
The mass matrix, $\mathbf{M}$, gives the relation between nodal accelerations and nodal forces.

$$\ddot{\mathbf{d}} = \mathbf{M}^{-1}(\mathbf{F}_i + \mathbf{F}_e)$$

LS-DYNA assumes that the mass is concentrated at the nodes. This leads to a diagonal mass matrix, that easily can be inverted.



$$m_i = \frac{1}{8}\sum_{j=1}^{n} m_j^e$$



Time integration
central difference

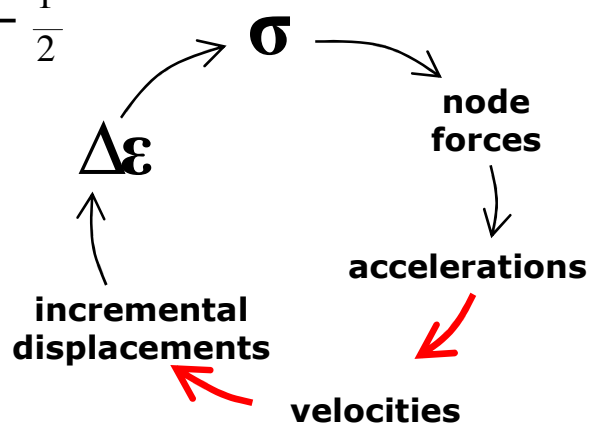
$$\ddot{\mathbf{d}}_n = \mathbf{M}^{-1}(\mathbf{F}_i + \mathbf{F}_e)$$

$$\dot{\mathbf{d}}_{n+\frac{1}{2}} = \dot{\mathbf{d}}_{n-\frac{1}{2}} + \ddot{\mathbf{d}}_n \Delta t$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \dot{\mathbf{d}}_{n+\frac{1}{2}} \Delta t$$

$n = n+1$

The central difference scheme is 2nd order accurate



Time integration

critical time step size

The central difference scheme is conditionally stable, the time step size must be smaller than a specific value.

$$\Delta t_{\text{cr}} = \frac{2}{\omega_{\max}}$$

$\omega_{\max}$ is the highest eigenfrequency of the system, and it is computationally expensive to derive.

$\omega_{\max}$ is estimated by looking at the geometry and material properties of each individual element

$$\omega_{\max} \approx \frac{2c}{\Delta x_{\min}}$$

c: sound speed

$\Delta x_{\min}$: characteristic size of smallest element

Strain rate/incremental strain

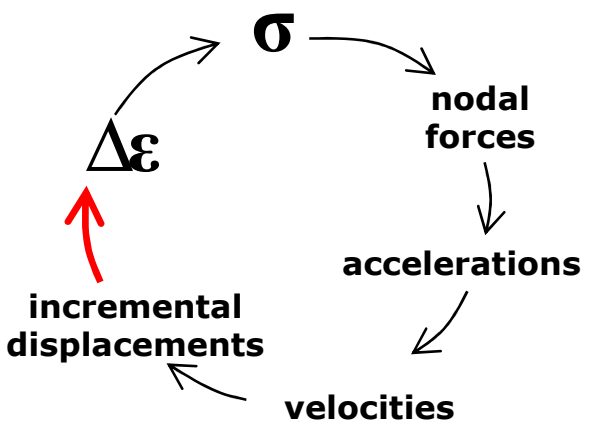
The incremental strain is defined as the strain rate times the time step size.

$$\Delta\boldsymbol{\varepsilon} = \mathbf{D}\,\Delta t$$

$\mathbf{D}$ is defined from the current velocity field $\mathbf{v}(\mathbf{x})$ as

$$\mathbf{D} = \begin{bmatrix} \mathrm{D}_{xx} & \mathrm{D}_{xy} & \mathrm{D}_{xz} \\ & \mathrm{D}_{yy} & \mathrm{D}_{yz} \\ symm & & \mathrm{D}_{zz} \end{bmatrix}$$

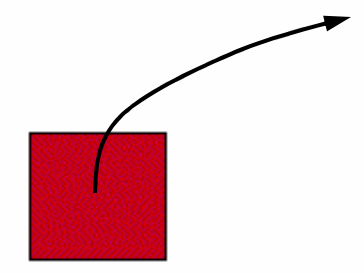
$$\mathrm{D}_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$$



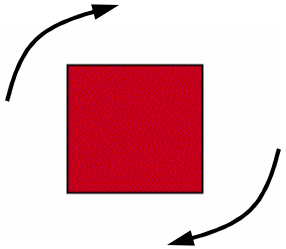
Strain rate/incremental strain

Examples:

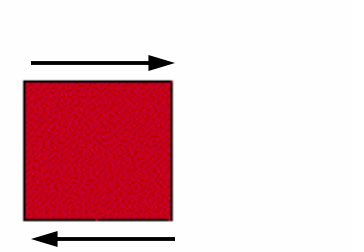
$$\mathbf{v}=\begin{Bmatrix}u\\v\end{Bmatrix}=\begin{Bmatrix}t\\1\end{Bmatrix}\Rightarrow\frac{\partial\mathbf{v}}{\partial\mathbf{x}}=\begin{bmatrix}0&0\\0&0\end{bmatrix}\Rightarrow\mathbf{D}=\mathbf{0}$$



$$\mathbf{v}=\begin{Bmatrix}y\\-x\end{Bmatrix}\Rightarrow\frac{\partial\mathbf{v}}{\partial\mathbf{x}}=\begin{bmatrix}0&1\\-1&0\end{bmatrix}\Rightarrow\mathbf{D}=\mathbf{0}$$



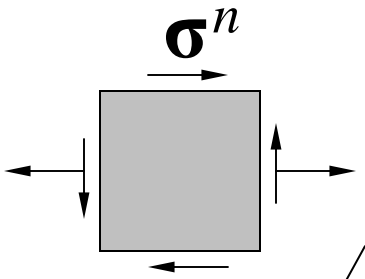
$$\mathbf{v}=\begin{Bmatrix}y\\0\end{Bmatrix}\Rightarrow\frac{\partial\mathbf{v}}{\partial\mathbf{x}}=\begin{bmatrix}0&1\\0&0\end{bmatrix}\Rightarrow\mathbf{D}=\begin{bmatrix}0&1/2\\1/2&0\end{bmatrix}$$



Jaumann stress rate

objective stress update

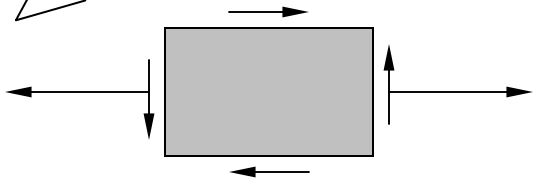
A rigid rotation should influence the spatial stress tensor.



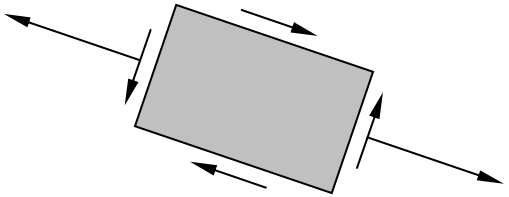
$\boldsymbol{\sigma}^n$

$$\Delta\breve{\boldsymbol{\sigma}} = \mathbf{L} : \mathbf{D}\ \Delta t$$

incremental deformation



incremental rotation, $\Delta\mathbf{R}$



$$\boldsymbol{\sigma}^{n+1} = \Delta\mathbf{R}(\boldsymbol{\sigma}^n + \Delta\breve{\boldsymbol{\sigma}})\Delta\mathbf{R}^{\mathrm{T}}$$

This transformation is expensive and LS-DYNA does, by default, not do it exactly

$\mathbf{R}$ can be derived from a polar decomposition of $\mathbf{F}$

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{RU}$$

Jaumann stress rate

objective stress update

LS-DYNA uses $\mathbf{W}$ to spin the stress tensor. This is the so called Jaumann stress rate.

tangential stiffness

rate of deformation

$$\dot{\boldsymbol{\sigma}} = \mathbf{L} : \mathbf{D}^{**} + \boldsymbol{\sigma}\mathbf{W} - \mathbf{W}\boldsymbol{\sigma} \qquad \textbf{(Eq. C)}$$

stress rate

spin tensor

$$D_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \qquad W_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i}\right)$$

Comment **

The product L:D

The product **L**:**D** is normally written in matrix form. We have the following relationship (Hooke's law) for a linear elastic material:

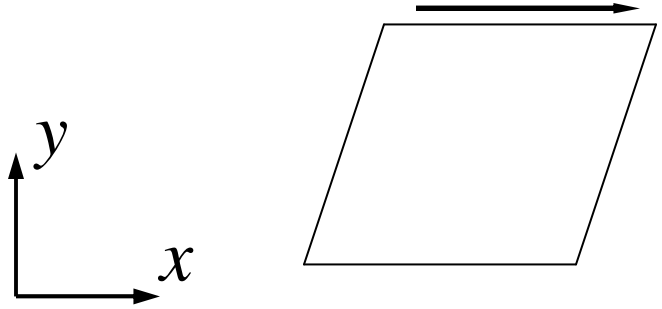
$$\begin{Bmatrix} \breve{\sigma}_{xx} \\ \breve{\sigma}_{yy} \\ \breve{\sigma}_{zz} \\ \breve{\tau}_{xy} \\ \breve{\tau}_{yz} \\ \breve{\tau}_{zx} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & (1-2\nu)/2 & & \\ & & & & (1-2\nu)/2 & \\ & & & & & (1-2\nu)/2 \end{bmatrix} \begin{Bmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ D_{xy} \\ D_{yz} \\ D_{zx} \end{Bmatrix}$$

$\breve{\boldsymbol{\sigma}}$ = **L** : **D**

Jaumann stress rate

example of simple shear

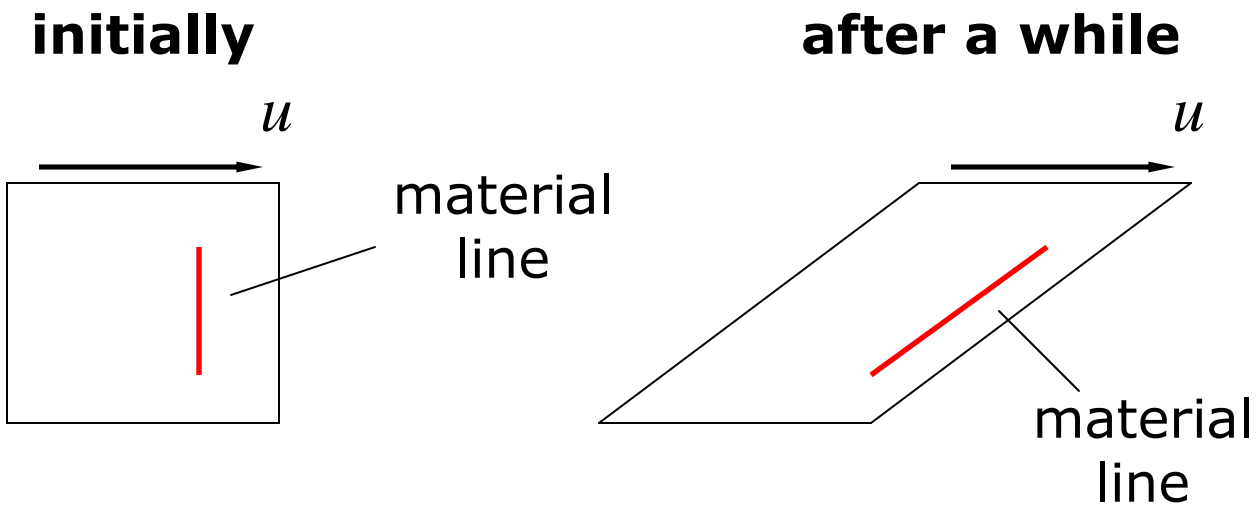
$$u = k \cdot y \quad [m/s]$$



$$\mathrm{D} = \frac{1}{2}\begin{bmatrix} 0 & k \\ k & 0 \end{bmatrix} \qquad \mathrm{W} = \frac{1}{2}\begin{bmatrix} 0 & k \\ -k & 0 \end{bmatrix}$$

Jaumann stress rate

example of simple shear



$$\mathbf{W} = \frac{1}{2}\begin{bmatrix} 0 & k \\ -k & 0 \end{bmatrix}$$

A material line will never rotate more than 90º. Consequently, the stresses should not be rotated more than 90º either. However, $\mathbf{W}$ is constant. Following **(Eq. C)**, the stress tensor will be rotated more than 90º at large shear strains.

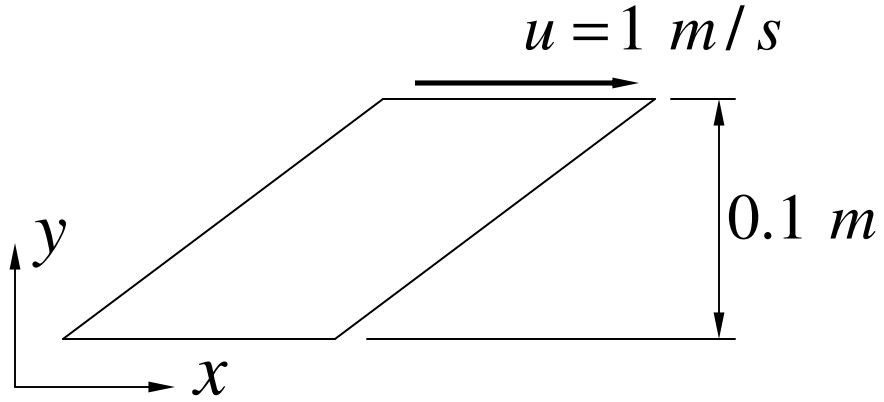
Jaumann stress rate

example of simple shear

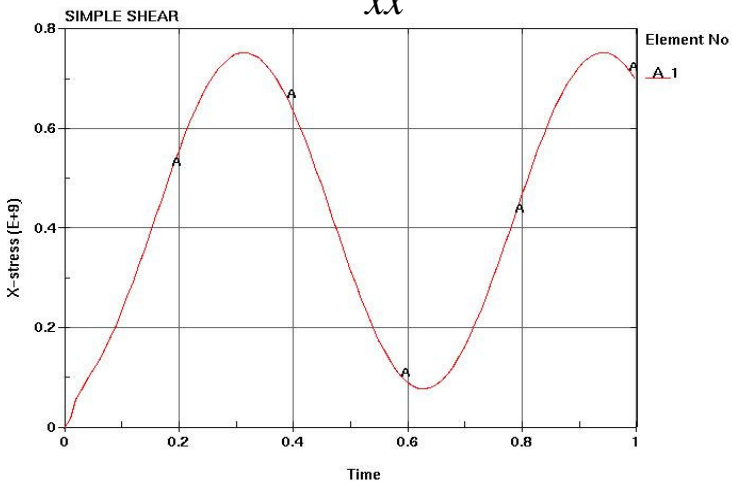
Bad estimation of rotations ➡ non-physical stress oscillations.

$$E = 1.0 \cdot 10^{10} \quad Pa$$
$$\nu = 0.3$$
$$\sigma_y = 1.0 \cdot 10^{9} \quad Pa$$
$$\phi = \sigma_{eff} - (\sigma_y + \varepsilon_{eff}^{p} \cdot 1.0 \cdot 10^{9}) \quad Pa$$
$$\sigma_{eff} = \sqrt{\frac{3}{2}(s_{ij} - s_{ij}^{*})(s_{ij} - s_{ij}^{*})} \quad Pa$$
$$\dot{s}_{ij}^{*} = 1.0 \cdot 10^{9} \dot{\varepsilon}_{ij}^{p} \quad Pa$$

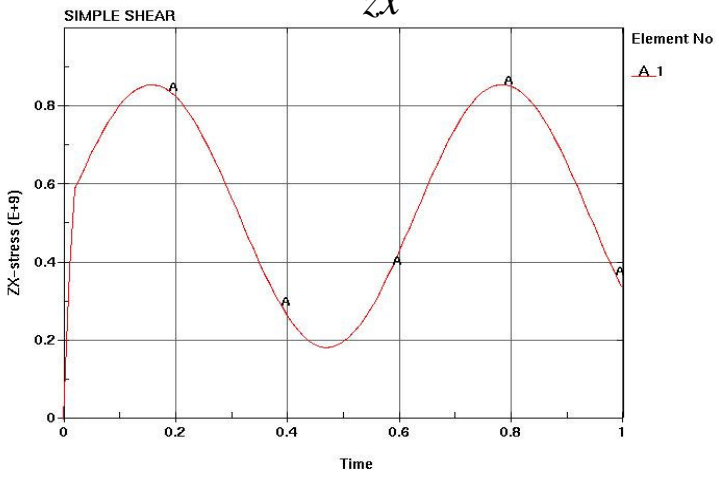
deviatoric stress



σ_{xx}

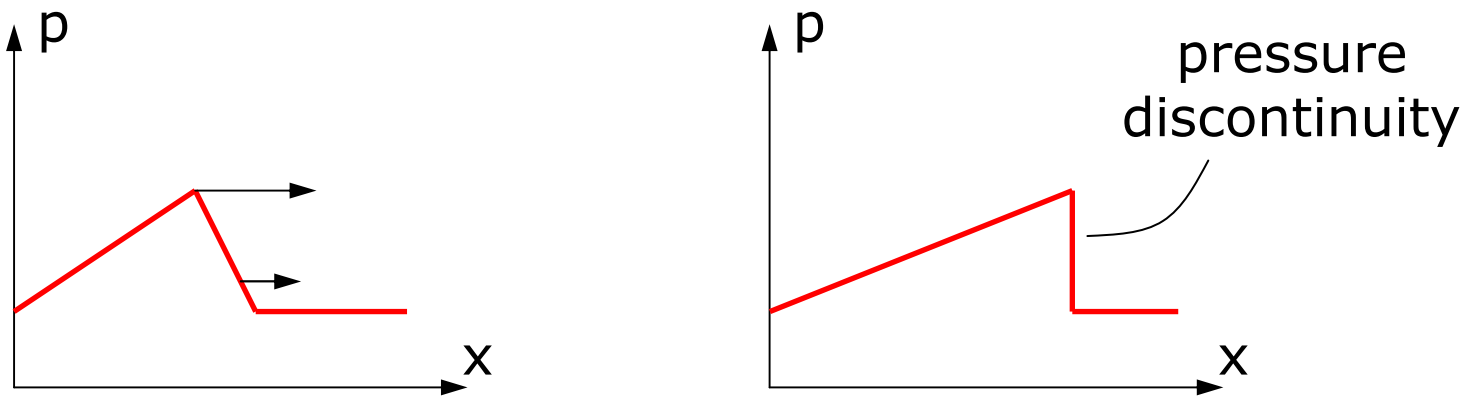


σ_{zx}



Bulk viscosity

Shock waves result from the fact that sound speed increases with increasing pressure.

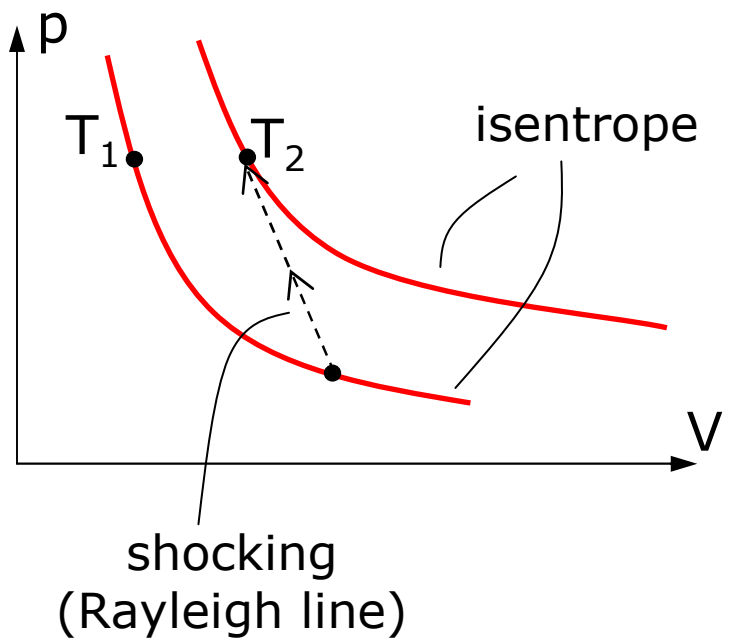


Assuming an isentropic compression as the material gets shocked, the governing equations predict a singularity and an inversion of the material. Numerically this is manifested as a noisy pressure response.

Bulk viscosity

In reality shocked material does not undergo an isentropic compression.

$T_1 < T_2$



Bulk viscosity

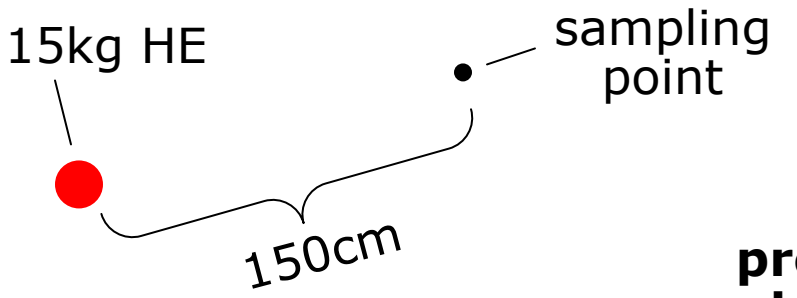
The problem of not following an isentrope is taken care of by an artificial viscosity term, q, known as bulk viscosity.

$$q = \rho l (C_0 l D_{kk}^2 + C_1 a D_{kk})$$

$l = V^{1/3}$	measure of element size
ρ	density
D_{kk}	trace of strain rate
a	speed of sound
C_0, C_1	dimensionless constants

The bulk viscosity damps out the highest frequencies and the ripple around the pressure front can be eliminated. The drawback is a smeared out pressure front.

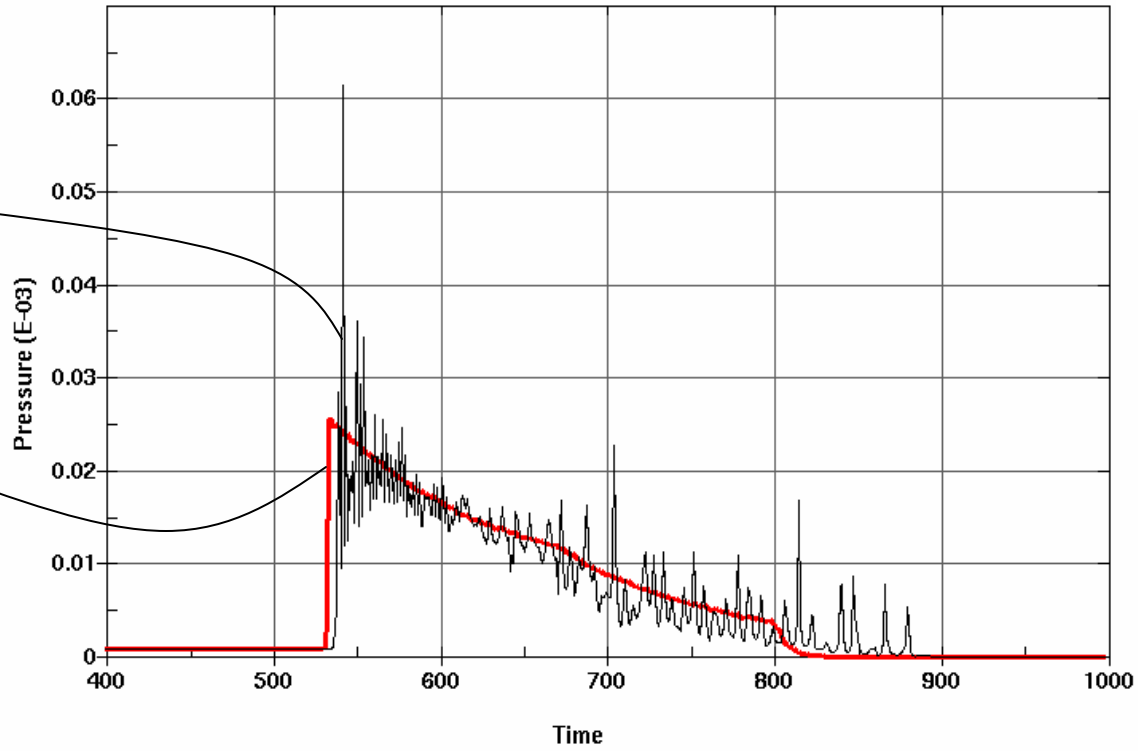
Bulk viscosity



pressure time history at the sampling point

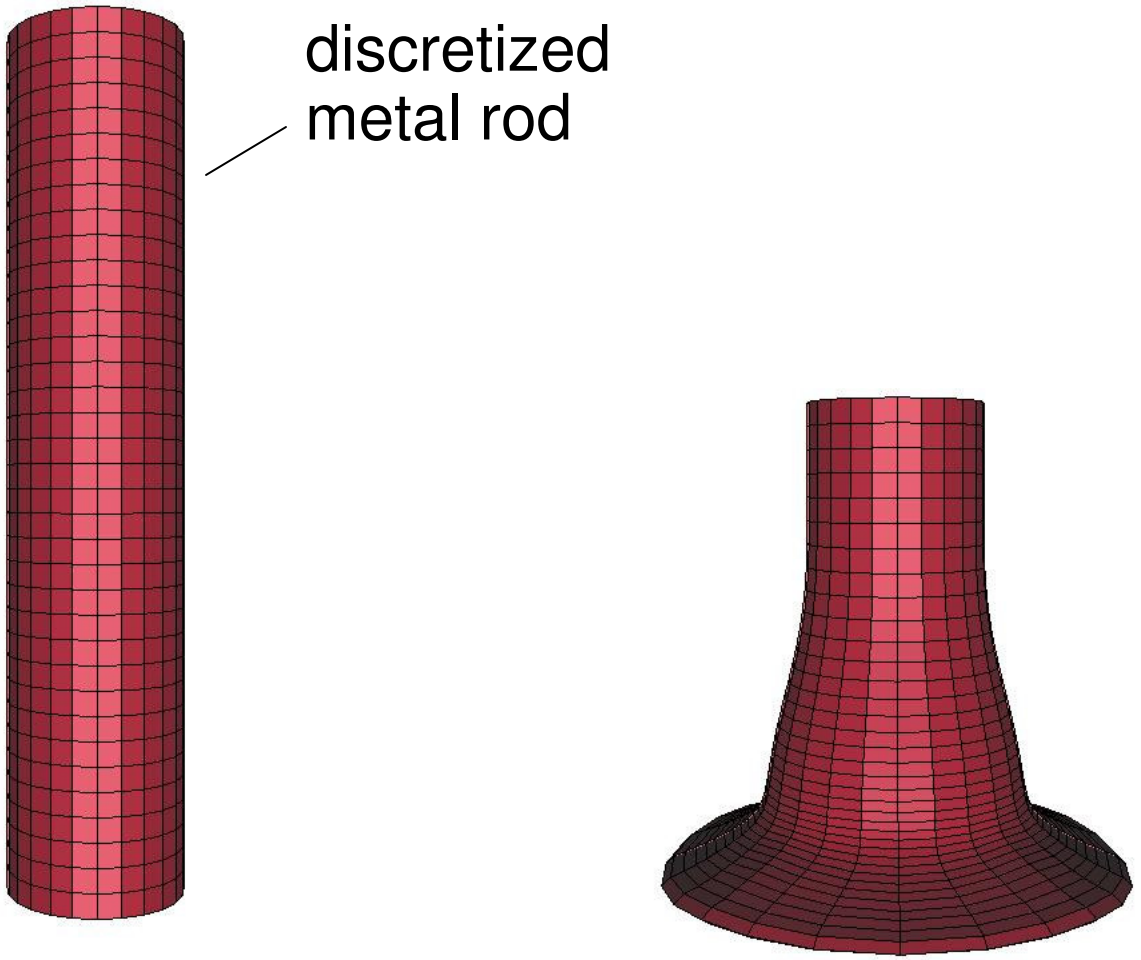
without bulk viscosity

with bulk viscosity



LS-DYNA example

Taylor bar impact



LS-DYNA example

Taylor bar impact

```
*KEYWORD
*TITLE
Lagrangian taylor bar
*CONTROL_TERMINATION
    8.0e-5
*CONTROL_TIMESTEP
       0.0       0.6
*DATABASE_BINARY_D3PLOT
    1.0e-6
*MAT_PLASTIC_KINEMATIC
         1    8930.0   1.17e11      0.35     4.0e8     1.0e8       1.0

*SECTION_SOLID
         1         0
*PART

         1         1         1
*RIGIDWALL_PLANAR
         1
       0.0   -0.0112       0.0       0.0    0.9888       0.0       0.0
*INITIAL_VELOCITY

       0.0    -280.0       0.0
```

LS-DYNA example

Taylor bar impact

```
*NODE
       1-1.084202172E-19 2.119999938E-02 1.599999960E-03       1       0
       2-5.333333393E-04 2.119999938E-02 1.599999960E-03       0       0
   .
   .
   .
    1774-2.771285828E-03-1.119999960E-02 1.599992393E-03       0       0
*ELEMENT_SOLID
       1       1       1       2       6       5      17      18      22      21
       2       1       2       3       7       6      18      19      23      22
   .
   .
   .
     972       1    1754    1154    1153    1758    1770    1170    1169    1774
*END
```

Single material Euler/ALE

- Governing Equations
- Operator split technique
  - time integration loop
  - mesh smoothing
  - advection schemes
- Keyword commands
  - **`*CONTROL_ALE`**
  - **`*SECTION_SOLID`**
  - **`*BOUNDARY_AMBIENT_EOS`**
- Input deck examples
  - ALE Taylor bar impact
  - viscous flow through pipe (Eulerian)

Governing equations

The relative motion between material and mesh complicates the evolution of all history variables, both element and node centered ones.

time derivative of history variable
in Lagrangian reference system

$$\dot{\phi} = \phi' + \nabla\phi \cdot (\dot{\mathbf{x}} - \mathbf{v})$$

time derivative of history variable
in ALE reference system

Governing equations

In the Eulerian and ALE-formulations, the nodes do not follow the material flow. There is flux of material between the elements. This complicates the governing equations.

Energy equation:

material velocity

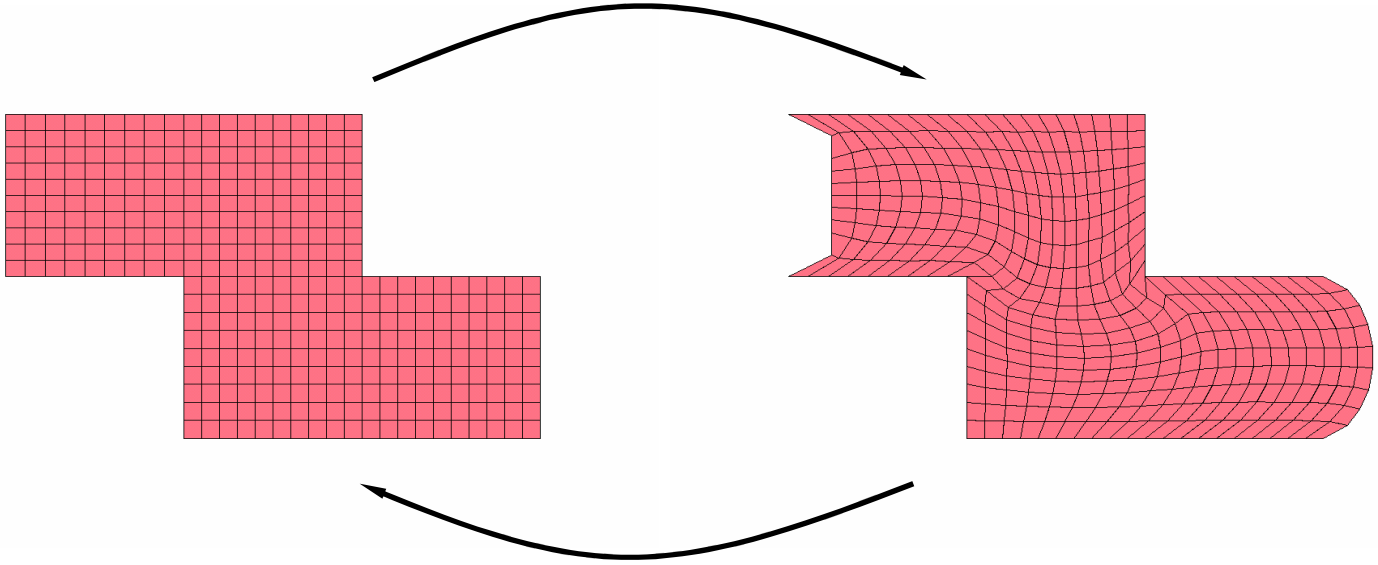
$$\rho \dot{u} = \rho \nabla u \cdot (\dot{\mathbf{x}} - \mathbf{v}) + \boldsymbol{\sigma} : \mathbf{D} + \rho r - \nabla \cdot \mathbf{q}$$

mesh velocity

Operator split technique

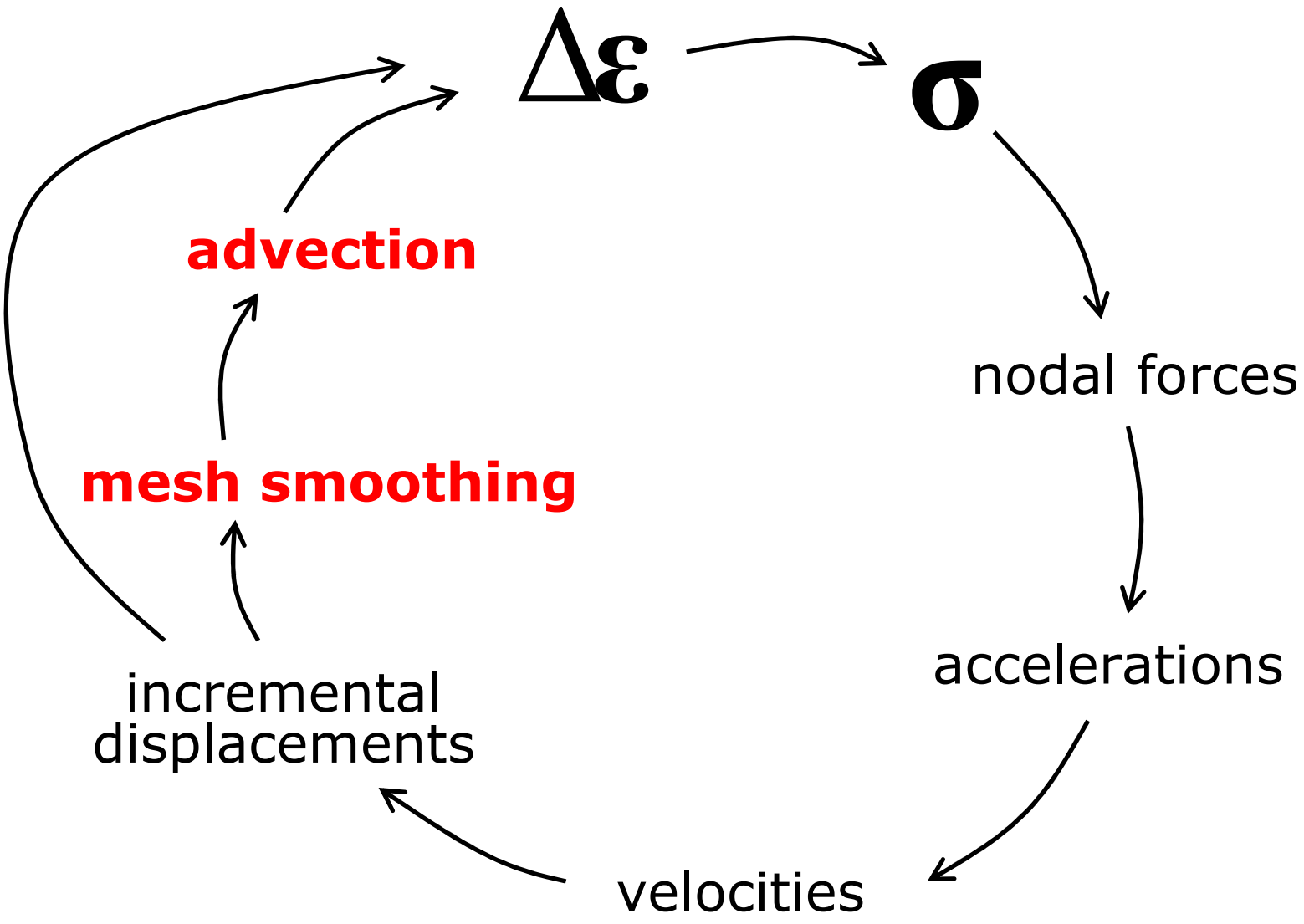
LS-DYNA first computes the Lagrangian time derivative and updates the history variables. Subsequently the relative motion between mesh and material is computed and the history variables are updated once more. This is a so called **operator split technique**.

pretend it is a Lagrangian formulation
and take a few time steps



move the nodes to new positions and map the solution
from the old configuration onto the new one

Time integration loop



Time integration loop

critical time step size

Due to the implemented advection schemes, the mass flow velocity influences the critical time step size.

speed of sound

characteristic size of element

$$\Delta t_{\text{cr}} \approx \min_{\text{nel}} \left[\frac{\Delta x^e}{c}, \frac{\Delta x^e}{4 v^{flux}} \right]$$

flux velocity

$$v^{flux} = \max_i \left(\left\| \dot{\mathbf{x}}_i - \mathbf{v}_i \right\| \right)$$

material velocity

mesh velocity

The definition of Δt_{cr} ensures that a particle won't flow across more than half an element in one time step. This is important for the advection accuracy.

Mesh smoothing

The repositioning of nodes is called mesh smoothing. In an Eulerian formulation the nodes are moved back to their initial positions.

In a single material ALE formulation nodes on the boundary can only be moved tangentially to the mesh surface. This heavily constrains the flexibility of the single material ALE mesh smoothing schemes.

The only practically useful methods implemented in LS-DYNA are four very similar iterative methods. We will take a quick look at two of those.

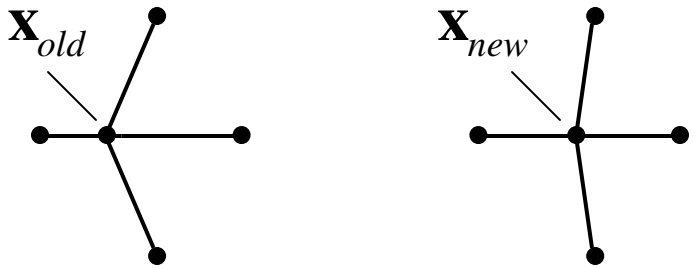
Mesh smoothing

There are different automatic iterative schemes aiming at finding ideal positions for the nodes. Ideal in the sense of minimizing element distortions. The most commonly used methods in LS-DYNA are called **simple average** and **equipotential smoothing**.

Simple average The updated node coordinate is the average coordinate of the surrounding nodes.

$$\mathbf{x}_{new} = \frac{1}{n}\sum_{1}^{n}\mathbf{x}_{old,n}$$

surrounding nodes

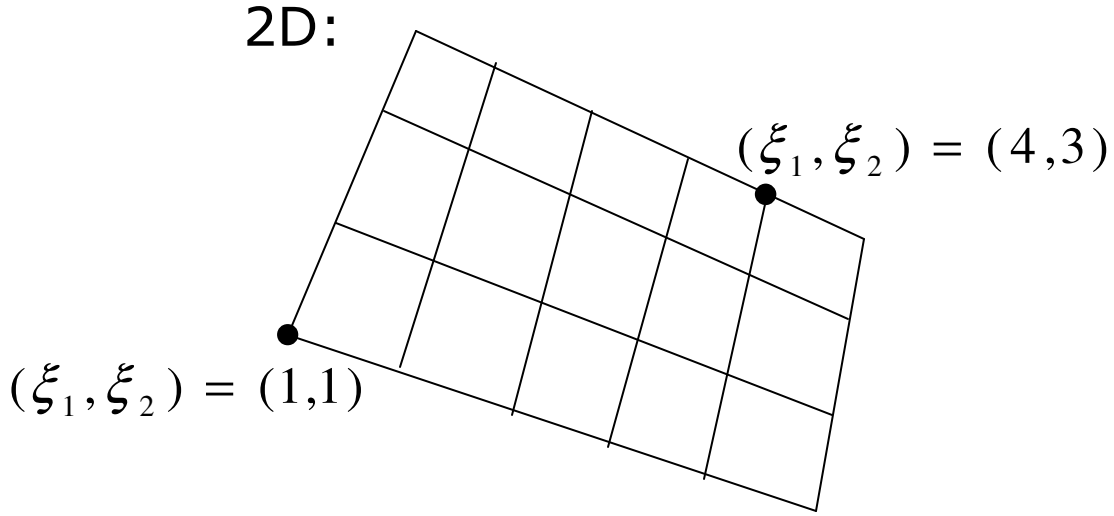


Mesh smoothing

Equipotential Requires a parametric mesh, where the node to be moved is connected to 8 elements (3D). The node location is obtained by solving Laplace's equation.

$$\Delta \xi_i = \frac{\partial^2 \xi_i}{\partial x^2} + \frac{\partial^2 \xi_i}{\partial y^2} + \frac{\partial^2 \xi_i}{\partial z^2} = 0 \qquad [i = 1,2,3]$$

2D:



Advection

After repositioning the nodes, all history variables and velocities need to be mapped from the old configuration onto the new one.

This is referred to as the **advection step**.

There are two different advection algorithms in LS-DYNA, one is spatially 1st order accurate and one is 2nd order accurate.

The 1st order method is the **Donor Cell** scheme and the 2nd order one is the **van Leer** scheme.

Both methods use the half index shift (HIS) scheme for advection of node centered variables (velocities and temperatures).

Advection

A good advection scheme should be **monotonic**, **conservative** and as little **dissipative** and **dispersive** as possible.

monotonic – The advection doesn't introduce higher max- and smaller min-values in the history variable fields.

$$\max_{\mathbf{x} \subset V}(\phi^{new}) \leq \max_{\mathbf{x} \subset V}(\phi^{old})$$

$$\min_{\mathbf{x} \subset V}(\phi^{new}) \geq \min_{\mathbf{x} \subset V}(\phi^{old})$$

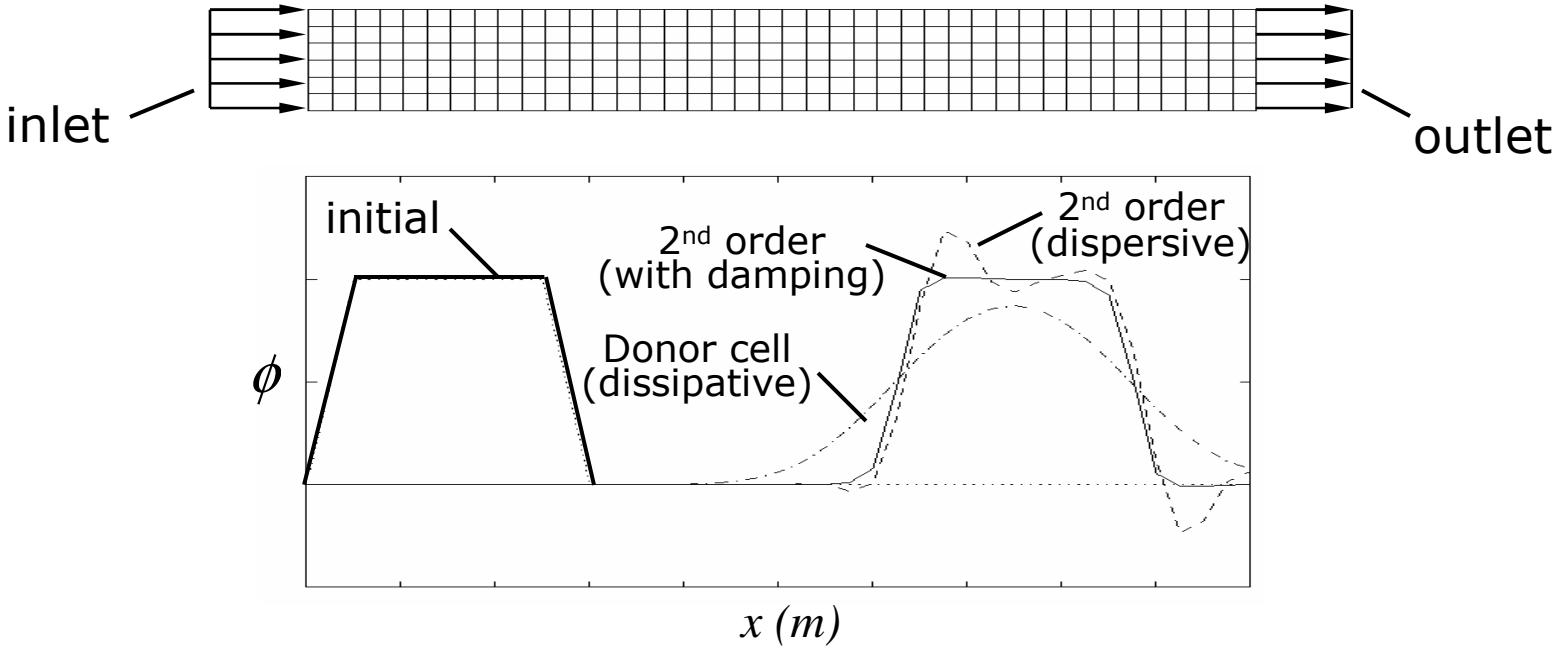
conservative – The advection doesn't change the total mass, momentum etc. of the system.

$$\int_V \phi^{new} \mathrm{d}V = \int_V \phi^{old} \mathrm{d}V$$

Advection

dissipative – The solution variable fields are smeared out.

dispersive – High spatial frequencies in the solution variable field travel slower than the mass flow velocity



Advection
Donor cell scheme

The Donor cell scheme is monotonic, conservative and fast.

Unfortunately it is only 1st order accurate and strongly dissipative. This severely limits the usefulness of the scheme.

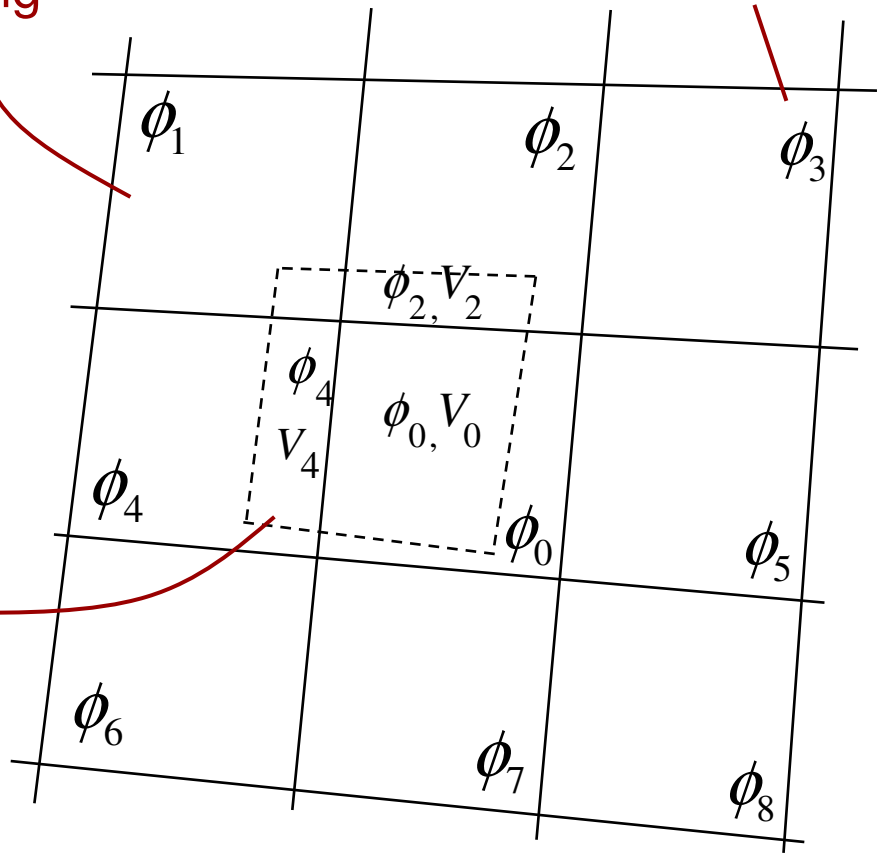
The Donor cell scheme is also dispersive. However, the dispersive errors are hidden by strong dissipation. High frequencies that travel too slow are quickly damped out.

Advection
Donor cell scheme

elements before node repositioning

element centered history variable

$$\phi_0^{new} = \frac{\phi_0 V_0 + \phi_2 V_2 + \phi_4 V_4}{V_0 + V_2 + V_4}$$



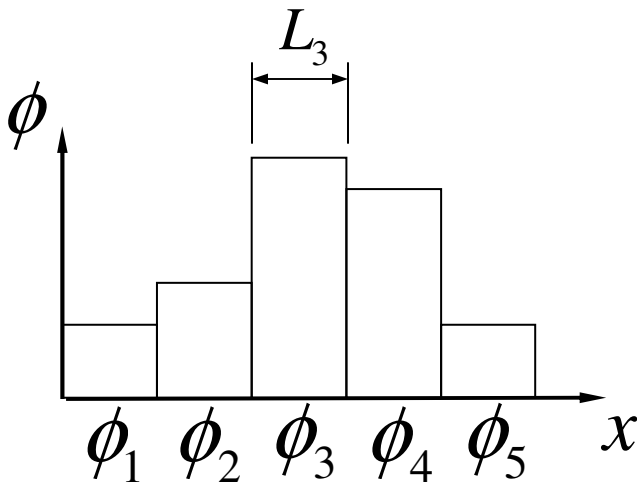
center element after node repositioning

Advection

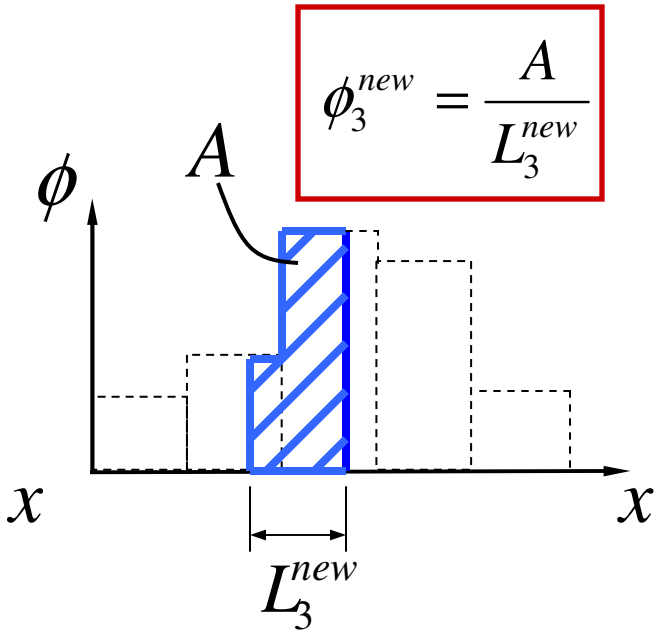
Donor cell scheme

A one-dimensional Donor cell example will help us understanding the van Leer scheme.

before mesh smoothing



new configuration

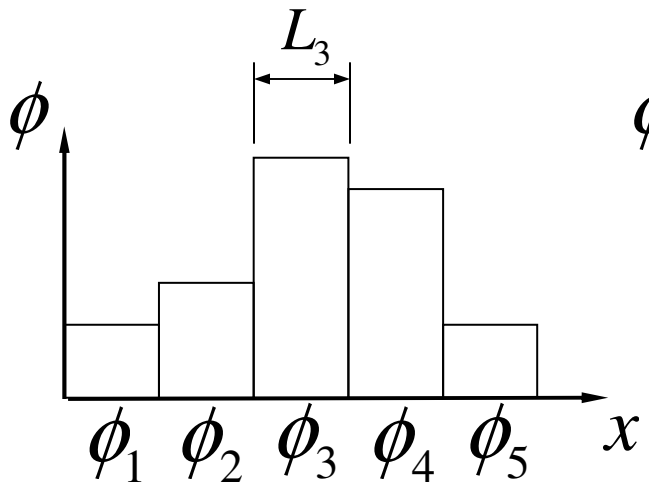


Advection
van Leer scheme

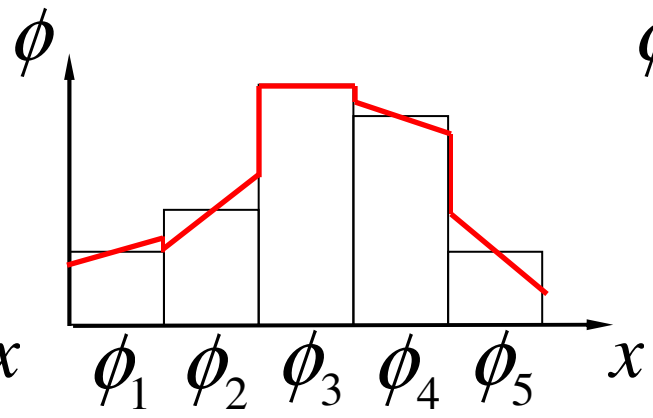
The van Leer scheme is monotonic, conservative, 2nd order accurate, but much slower than the Donor cell method. It is based on the assumption of a mesh with rectangular elements. Distorted elements introduce some 2nd order errors and the scheme actually becomes 1st order accurate.

The basic idea is to reconstruct an assumed a linear variation of the history variable fields. These assumed fields are mapped from the old mesh onto the new configuration.

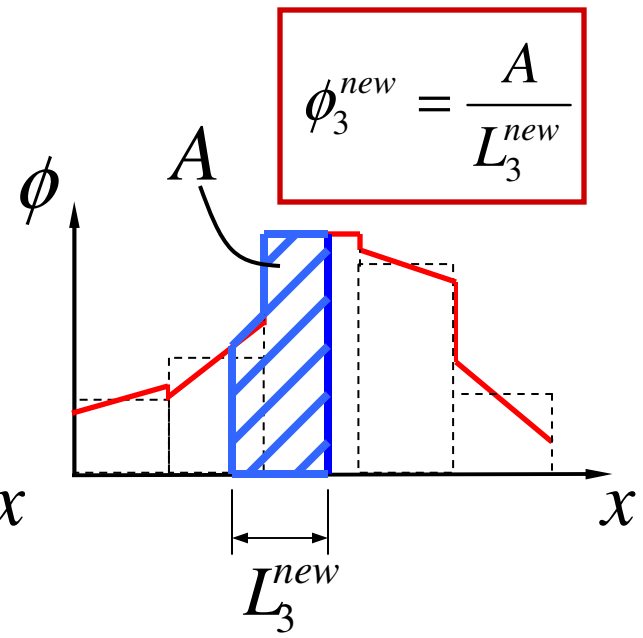
before mesh smoothing



piecewise linear field



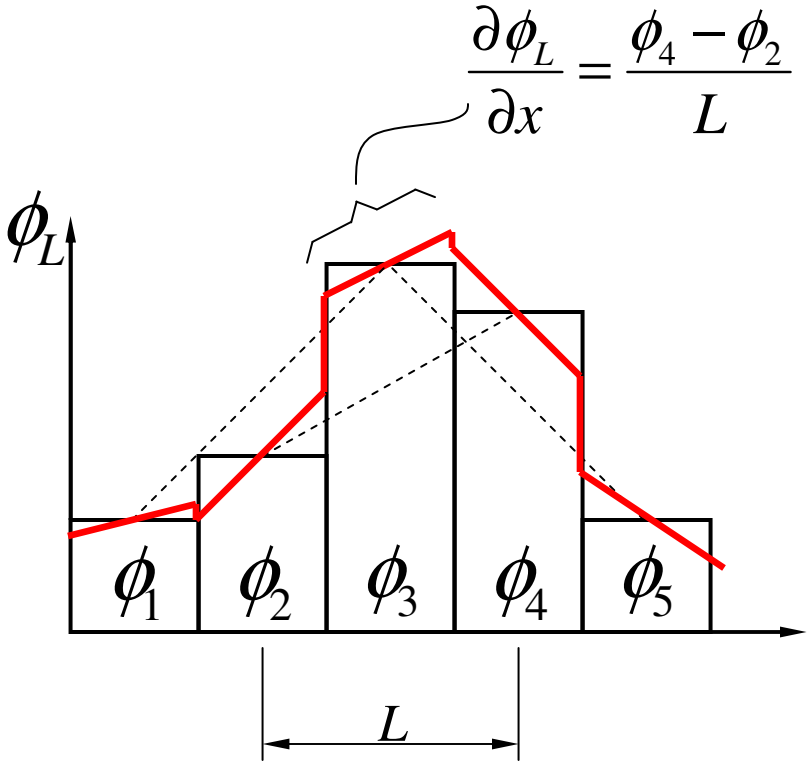
new configuration



Advection

van Leer scheme

The first step is to use central differences to construct the piecewise linear solution variable field, ϕ_L.



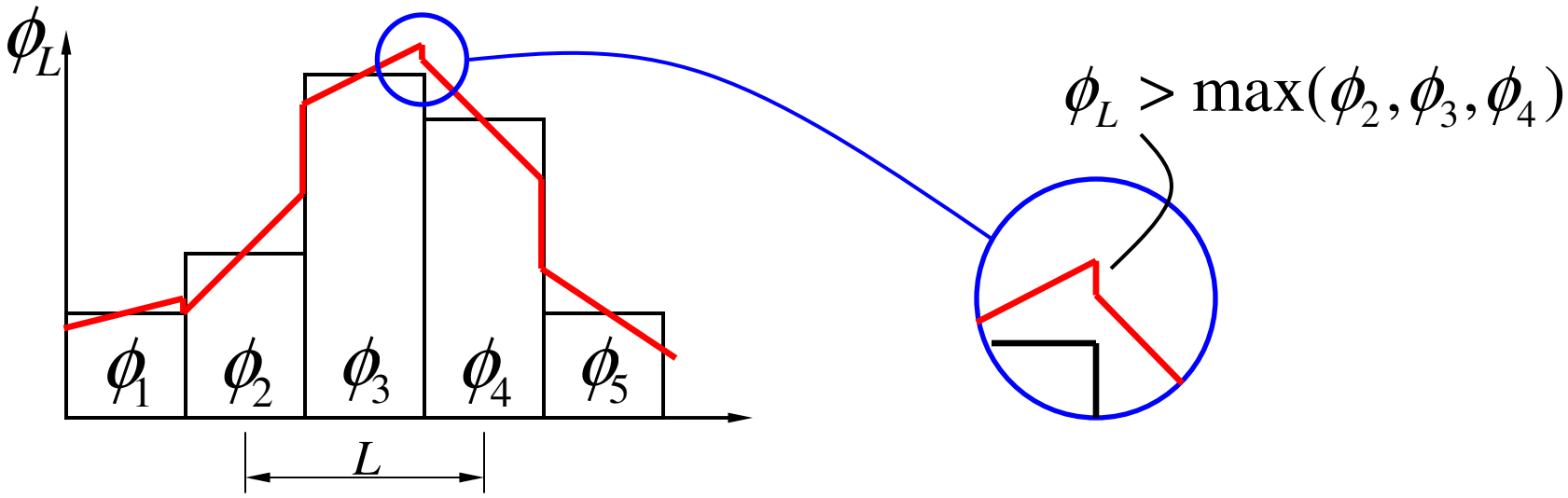
Advection
van Leer scheme

The linear field, ϕ_L, might contain local max/min's exceeding the real max/min values.

In the example below, fluxing from element 4 to element 3 will introduce a new maximum:

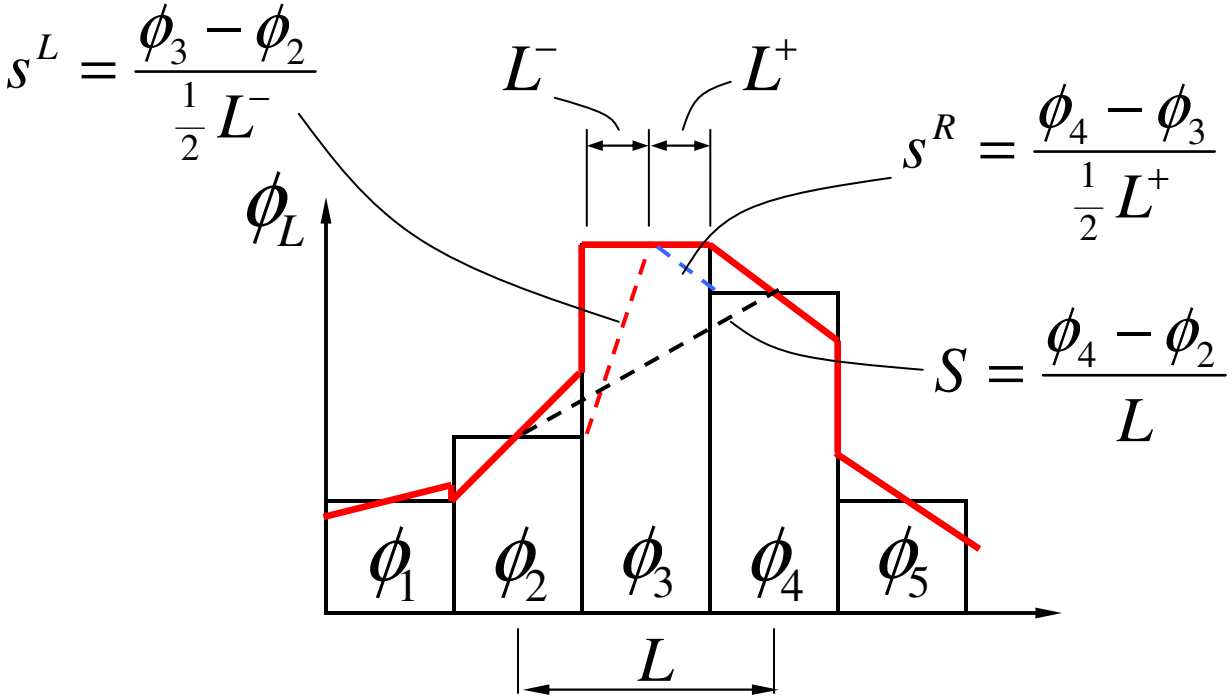
$$\phi_3^{new} > \max_{i=1,nel}(\phi_i)$$

That is, the monotonicity will be violated, unless the linear field is damped where necessary.



Advection
van Leer scheme

Monotonicity can be ensured by introducing measures of the maximum allowed slopes to the left, s^L,and to the right, s^R.



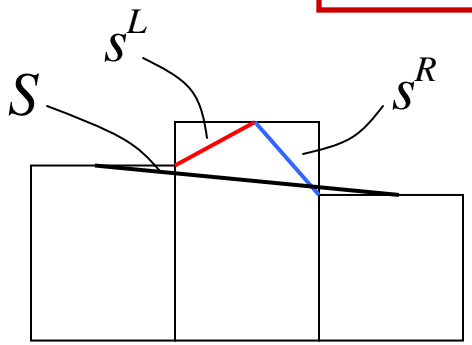
$$\frac{\partial \phi_L}{\partial x} = \frac{1}{2}(\mathrm{sgn}(s^L) + \mathrm{sgn}(s^R)) \cdot \min(|s^L|, |s^R|, |S|)$$

Advection

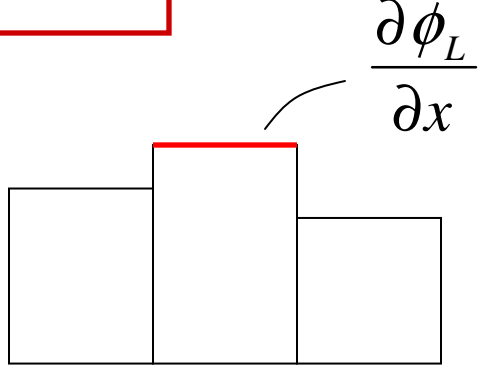
van Leer scheme

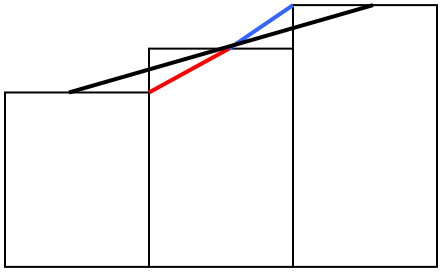
$$\frac{\partial \phi_L}{\partial x} = \frac{1}{2}(\mathrm{sgn}(s^L) + \mathrm{sgn}(s^R)) \cdot \min(|s^L|, |s^R|, |S|)$$

examples

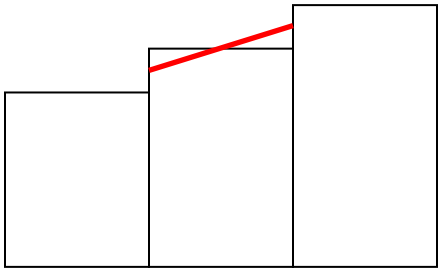


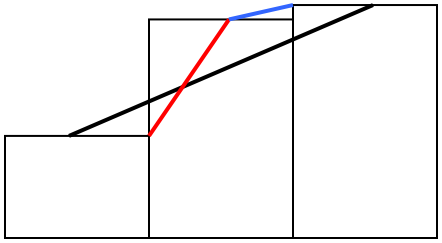
$$\mathrm{sgn}(s^L) \neq \mathrm{sgn}(s^R) \Rightarrow \frac{\partial \phi_L}{\partial x} = 0$$



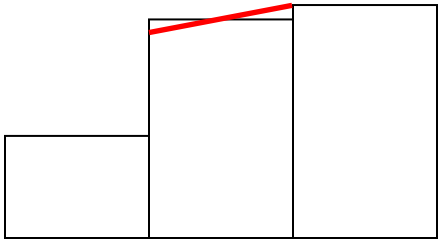


$$\min(|s^L|, |s^R|, |S|) = |S| \Rightarrow \frac{\partial \phi_L}{\partial x} = S$$



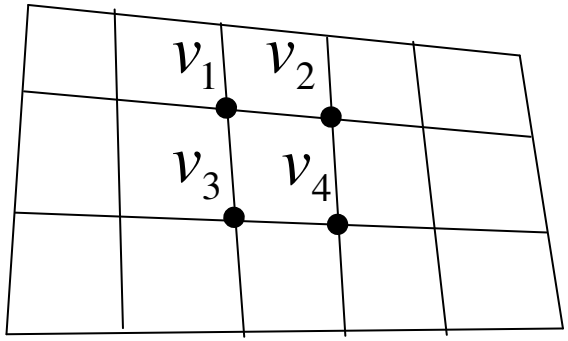


$$\min(|s^L|, |s^R|, |S|) = |s^R| \Rightarrow \frac{\partial \phi_L}{\partial x} = s^R$$

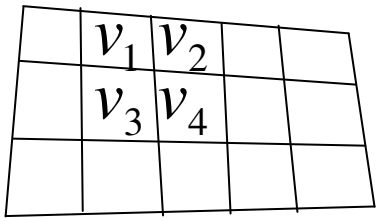


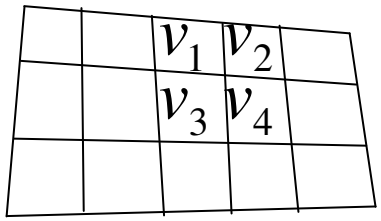
Advection
half index shift

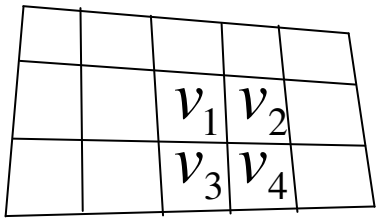
Node centered variables need a special treatment in the advection. The complexity of the applied scheme is motivated by the wish to have a conservative method. For example, it is desirable to preserve the total momentum of the system, when advecting velocities.

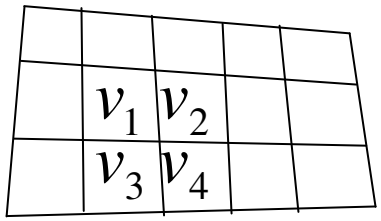


1. Move the velocities to the integration points (4 different positions in 2D and 8 in 3D).
2. Advect using the standard Donor cell or van Leer scheme.
3. Move velocities back to the nodes.









Advection
energy conservation

The total internal energy is conserved in the advection, for materials with an EOS. However, the kinetic energy is not.

The advection schemes are designed to conserve the total momentum (on the expense of kinetic energy).

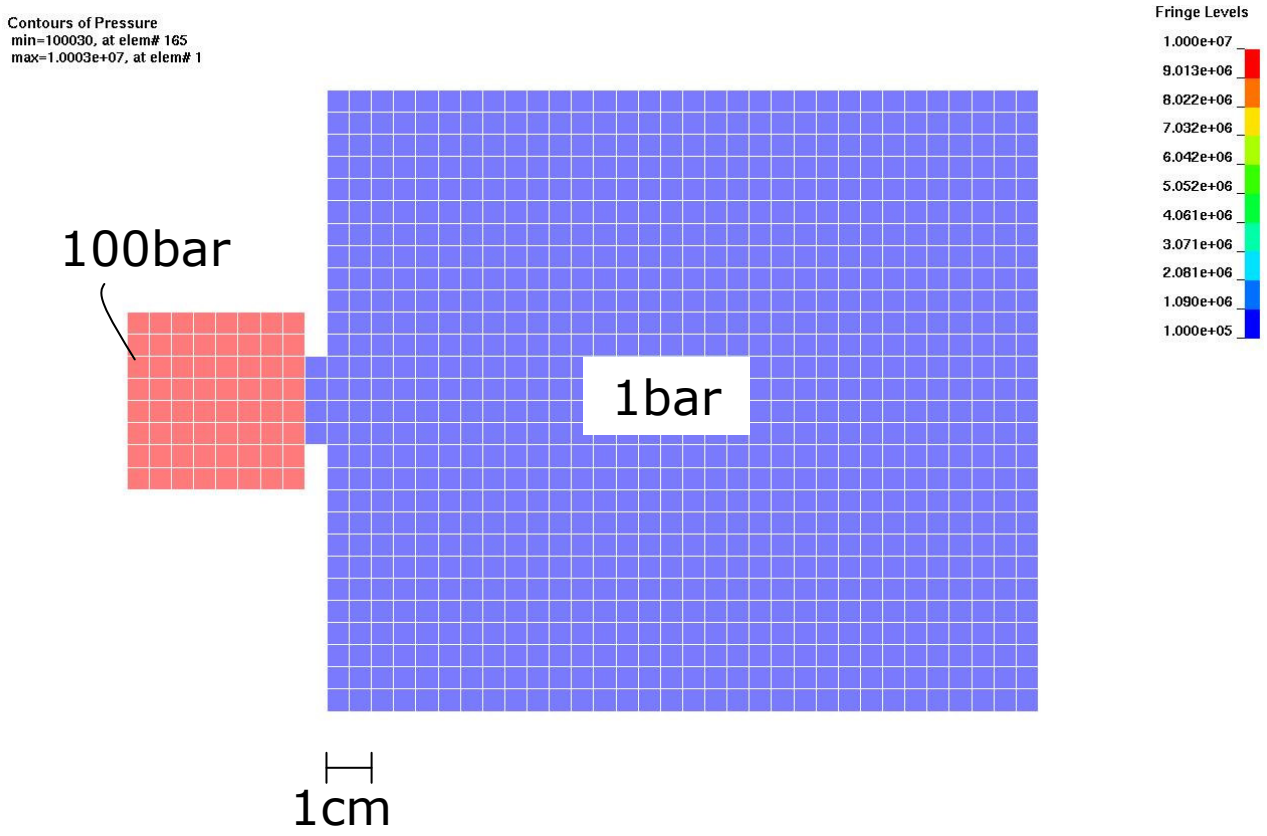
This is not acceptable in situations where the results are highly dependent on the total energy of the system.

For this reason, a new advection option (**METH=3**) has been developed. The new algorithm automatically converts dissipated kinetic energy into internal energy, such that the total energy is conserved.

The option in only activated for materials with an EOS (and for ***MAT_GAS_MIXTURE**)

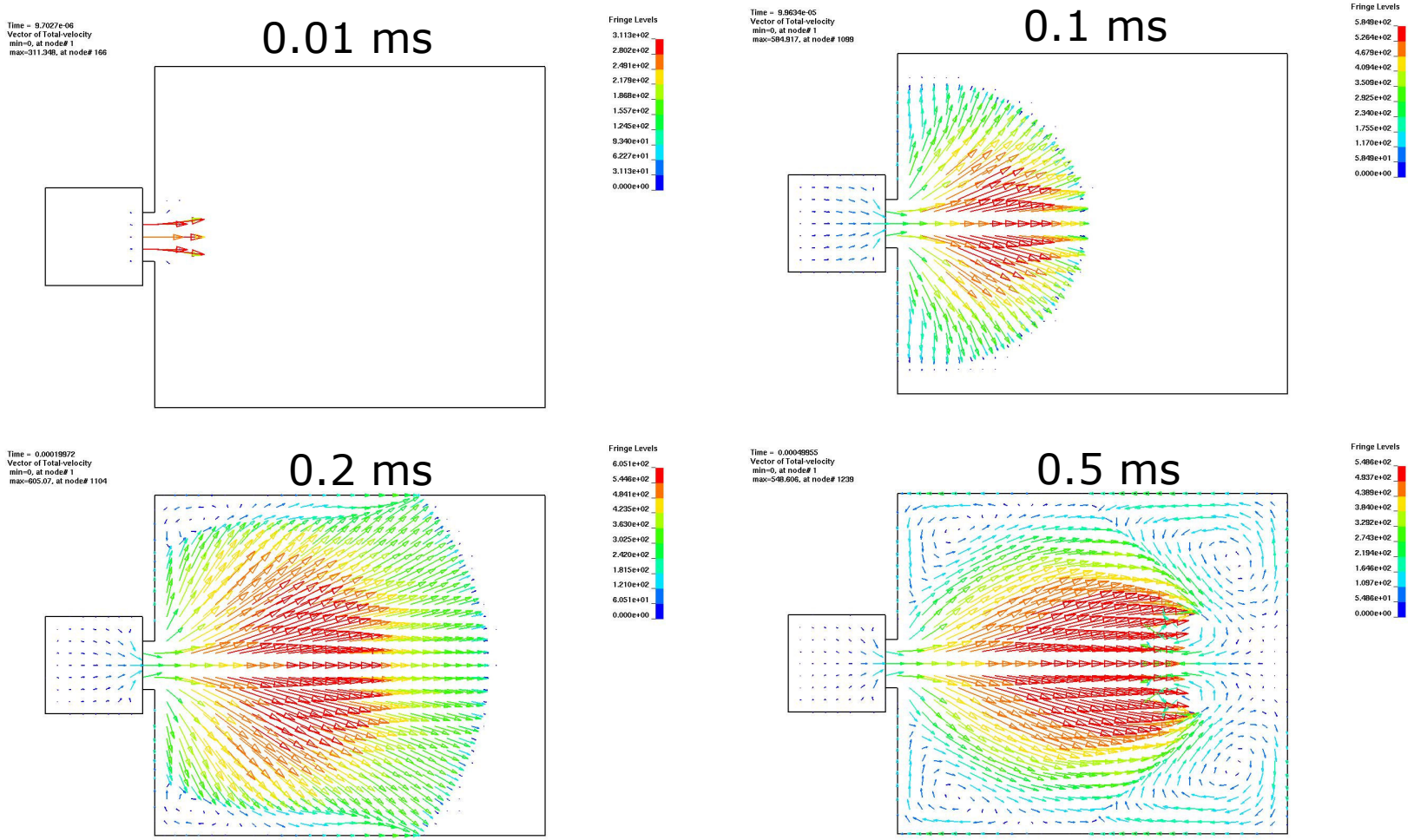
Advection
energy conservation

Example: Tank test



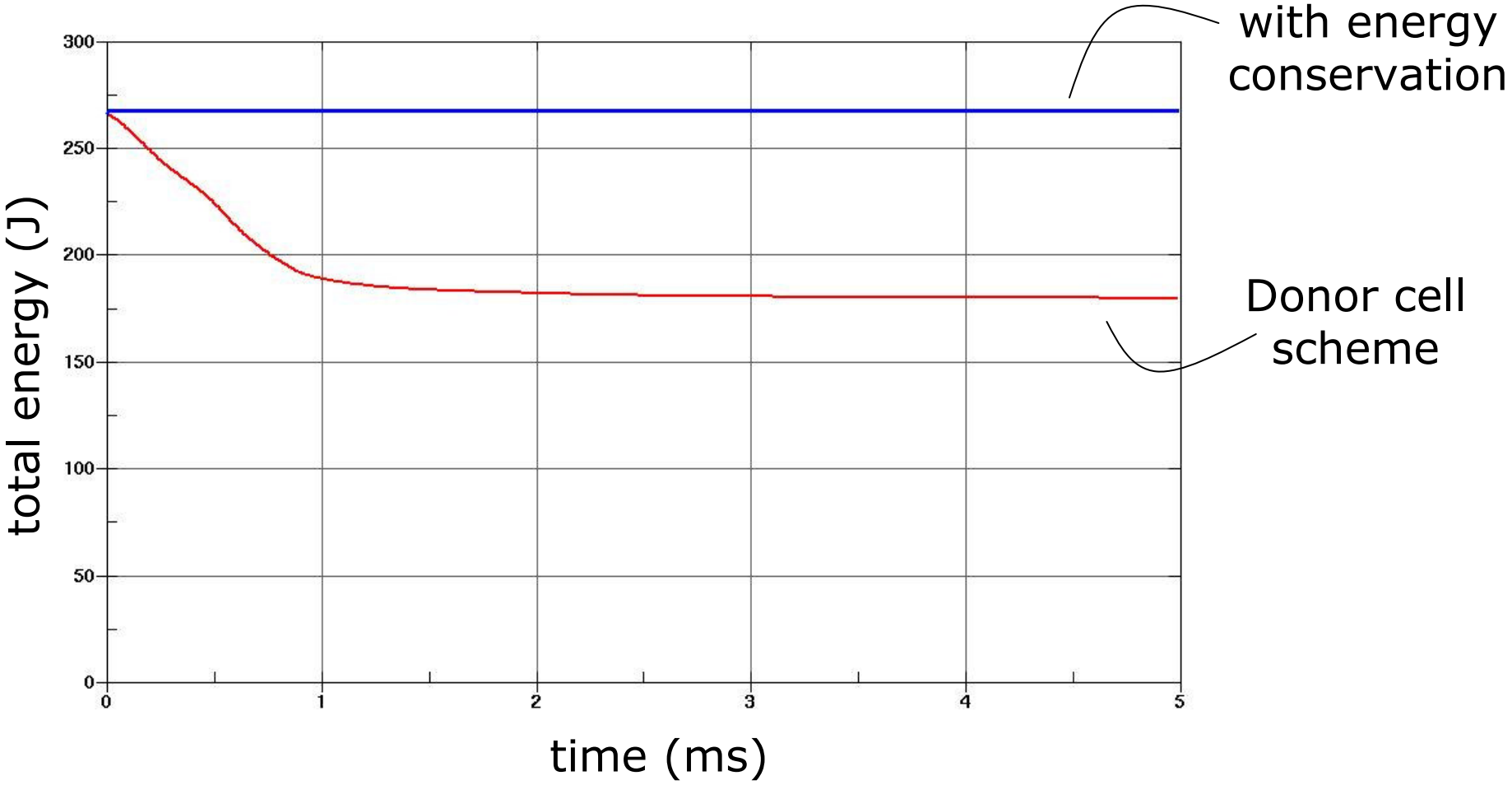
Advection
energy conservation

Example: Tank test



Advection
energy conservation

Example: Tank test



*CONTROL_ALE

DCT	NADV	METH	AFAC	BFAC	CFAC	DFAC
START	END	AAFAC	VFAC	PRIT	EBC	PREF
			CHKR			

DCT (only active in 2D ALE)

NADV Number of time steps between mesh smoothing and advection

METH Advection method (1=Donor Cell, 2=van Leer)

AFAC, **BFAC**, **CFAC**, **DFAC** Classical mesh smoothing parameters

START Birth time for ALE

END Death time for ALE

AAFAC (obsolete)

VFAC Void factor for element formulation 12

PRIT Pressure equilibrium flag

EBC Flag for automatically applied BC's along the boundary of the ALE domain

PREF Reference pressure (applied to all free boundaries of the ALE mesh)

CHKR Checkerboard pattern prevention parameter

*SECTION_SOLID

SECID ELFORM AET

SECID Section ID

ELFORM Element formulation

Eq. 1: Reuced integrated Lagrangian

Eq. 2: S/R Lagrangian

Eq. 11: ALE/Eulerian

AET Ambient element type

Eq. 4: Prescribed state at inflow boundary

preserve initial state through out the analysis or prescribe state with the command *BOUNDARY_AMBIENT_EOS

*BOUNDARY_AMBIENT_EOS

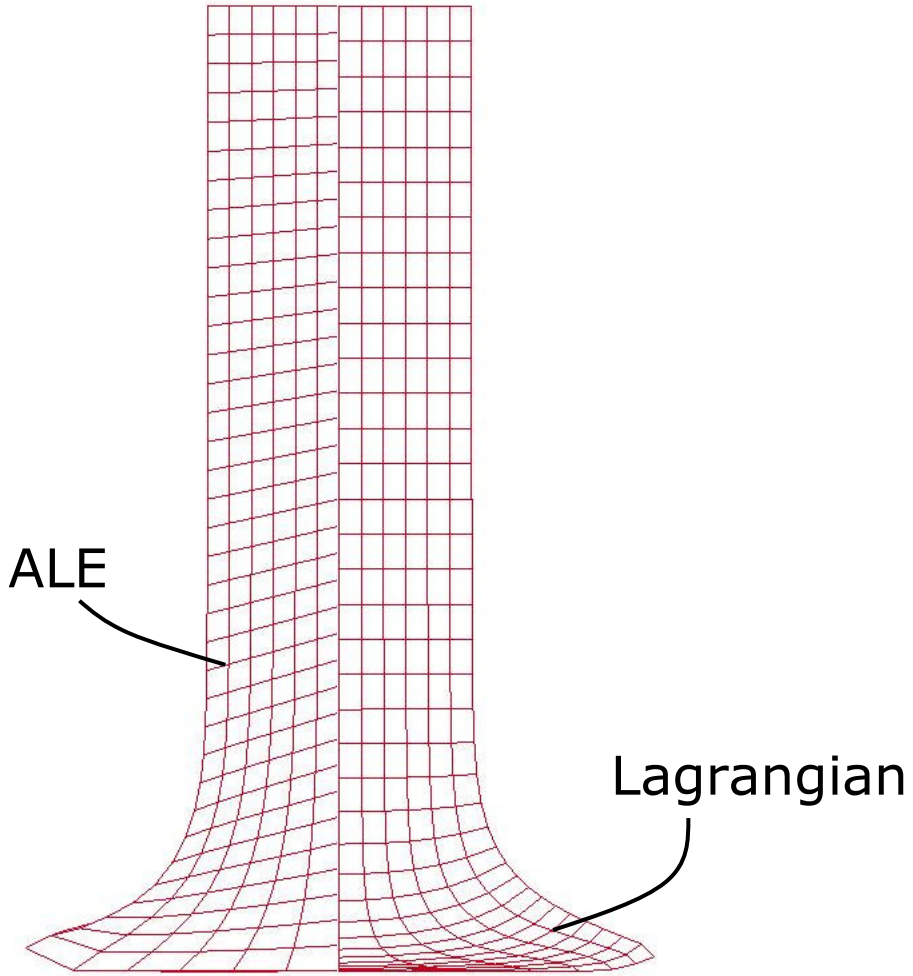
PID LCID1 LCID2

PID Part ID

LCID1 Load curve ID for internal energy

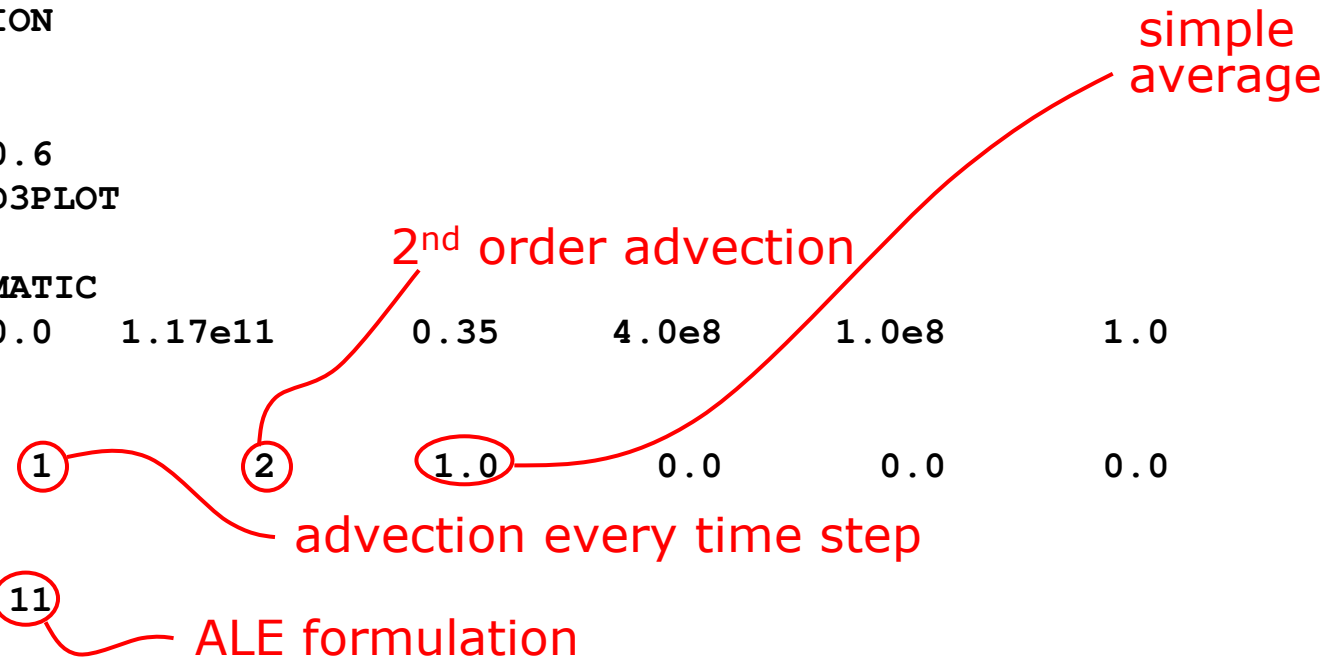
LCID2 Load curve ID for relative volume

LS-DYNA example
Taylor bar impact



LS-DYNA example
Taylor bar impact

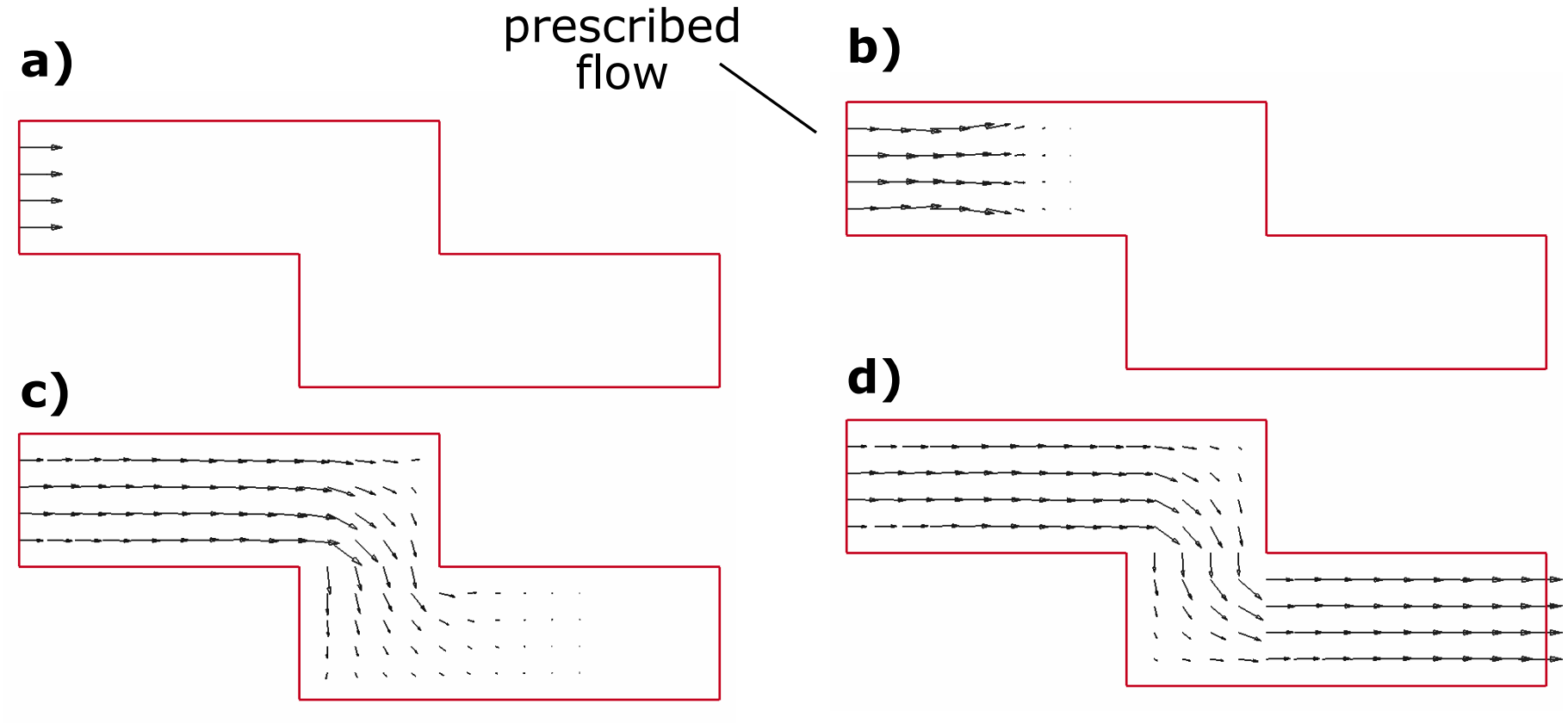
```
*KEYWORD
*TITLE
ALE taylor bar
*CONTROL_TERMINATION
    8.0e-5
*CONTROL_TIMESTEP
         0       0.6
*DATABASE_BINARY_D3PLOT
    1.0e-6
*MAT_PLASTIC_KINEMATIC
         1    8930.0   1.17e11      0.35     4.0e8     1.0e8       1.0

*CONTROL_ALE
         0         1         2       1.0       0.0       0.0       0.0

*SECTION_SOLID
         1        11
*PART

         1         1         1
*RIGIDWALL_PLANAR
         1         0         0
       0.0   -0.0112       0.0       0.0    0.9888       0.0       0.0
*INITIAL_VELOCITY

       0.0    -280.0       0.0
```



LS-DYNA example
Taylor bar impact

```
*NODE
       1-1.084202172E-19 2.119999938E-02 1.599999960E-03       1       0
       2-5.333333393E-04 2.119999938E-02 1.599999960E-03       0       0
   .
   .
   .
    1774-2.771285828E-03-1.119999960E-02 1.599992393E-03       0       0
*ELEMENT_SOLID
       1       1       1       2       6       5      17      18      22      21
       2       1       2       3       7       6      18      19      23      22
   .
   .
   .
     972       1    1754    1154    1153    1758    1770    1170    1169    1774
*END
```

LS-DYNA example
viscous flow



LS-DYNA example
viscous flow

```
*KEYWORD
*TITLE
FLow through box
*CONTROL_ALE
         0         1         1

*CONTROL_TERMINATION
     0.005
*CONTROL_TIMESTEP
         0       0.7
*DATABASE_BINARY_D3PLOT
    0.0001
*MAT_NULL
         1    1000.0   -1.0e10        0.3
*EOS_GRUNEISEN
         1    1500.0

*SECTION_SOLID
         1        11
*SECTION_SOLID
         2        11         4
```

1 time step between advection

1st order advection

prescribed state at inlet

LS-DYNA example
viscous flow

```
*PART

         1         1         1         1
*PART
         2         2         1         1
*BOUNDARY_AMBIENT_EOS
         2         1         2
*DEFINE_CURVE
         1
 0.0, 0.0
 1.0, 0.0
*DEFINE_CURVE
         2
 0.0, 0.99
 1.0, 0.99
*NODE
       1 0.000000000E 5.000000000E-01 0.000000000E       7       0
    .
    .
     384 2.600000143E 5.000000000E-01 1.000015E-01       7       0
*ELEMENT_SOLID
       1       2       1       2       4       3      13      14      16      15
    .
    .
     155       1     313     314     324     323     373     374     384     383
*END
```

prescribed energy and compression

relative volume (load curve ID)

internal energy (load curve ID)

Multi-material Eulerian formulation

- Introduction
- Composite stress
- Internal forces
- Pressure equilibrium
- Interface reconstruction
- Keyword commands

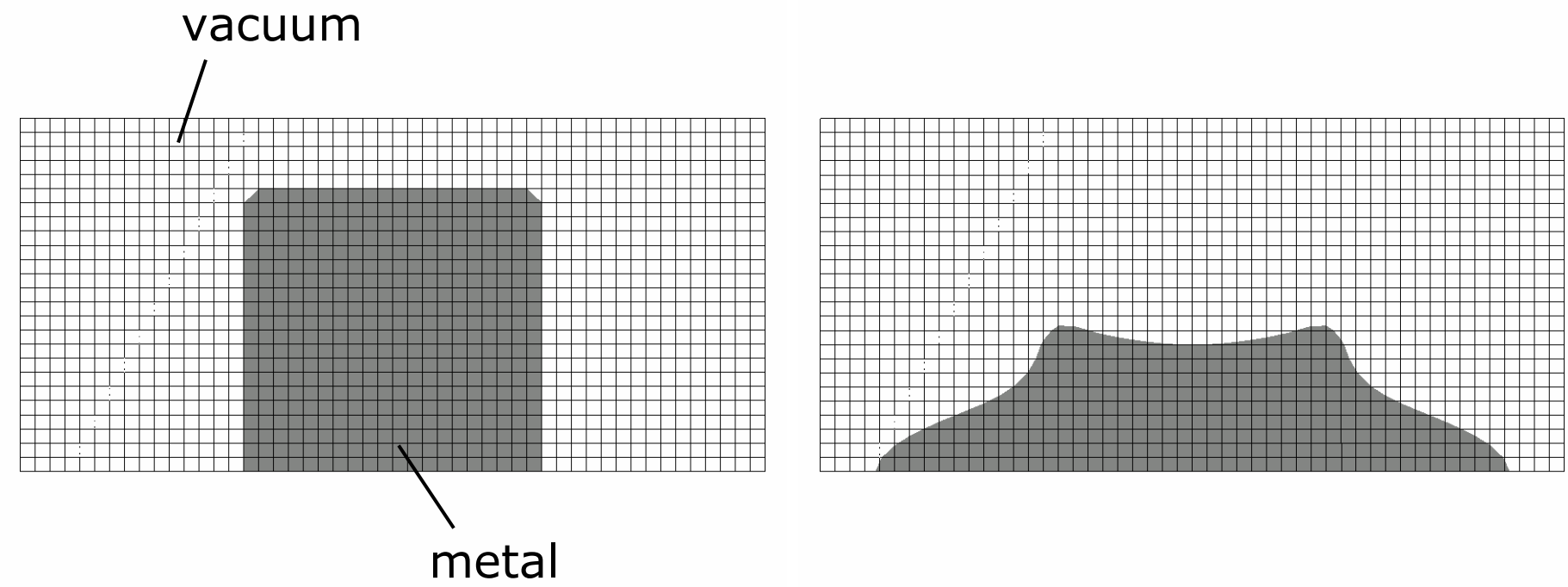
```
*CONTROL_ALE
*ALE_MULTI-MATERIAL_GROUP
*INITIAL_VOLUME_FRACTION
```

Introduction

The multi-material Eulerian formulation is a method where two or more different materials can be mixed within the same fixed mesh.

Each element in the Eulerian mesh contains a certain volume fraction of each material.

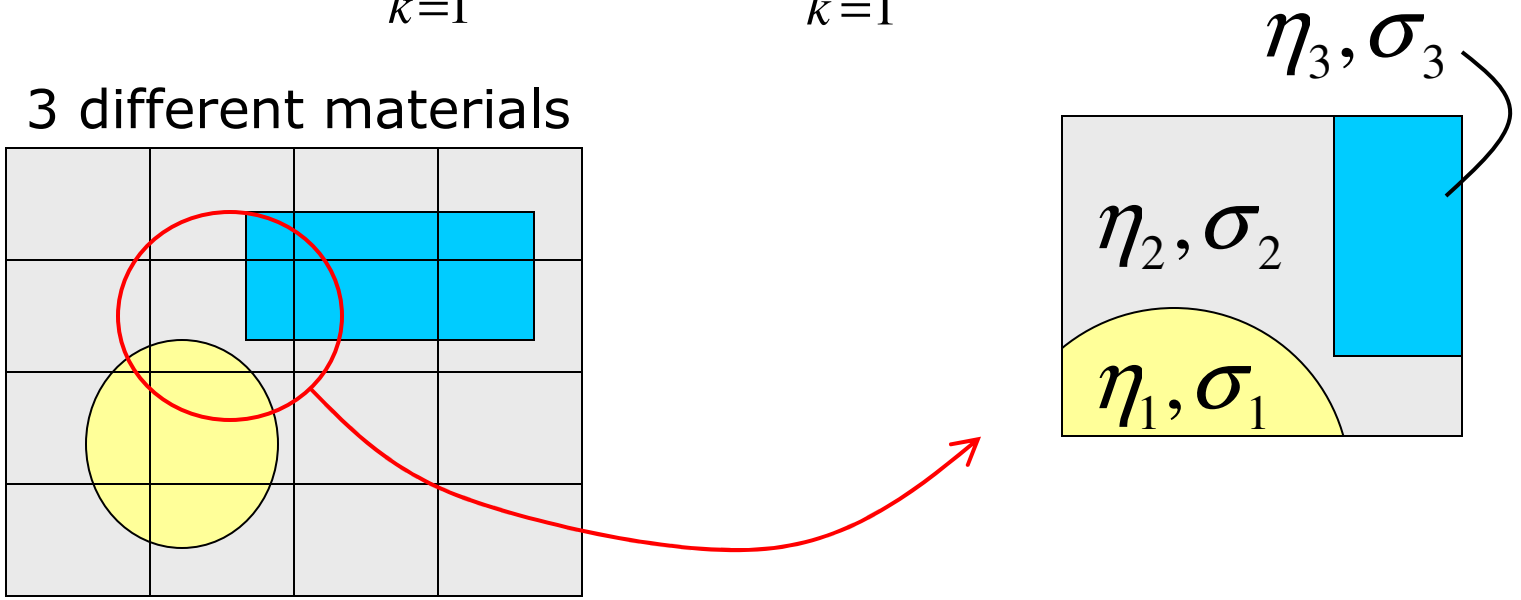
Boundaries are defined internally as straight cuts through mixed elements.



Composite stress tensor

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

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



Internal forces

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

internal element force

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

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

derivatives of shape functions

composite stress vector

element volume

Pressure equilibrium

In a multi-material formulation, each element might contain a mixture of different materials.

By default, all materials inside one element are assumed being exposed to the same strain rate.

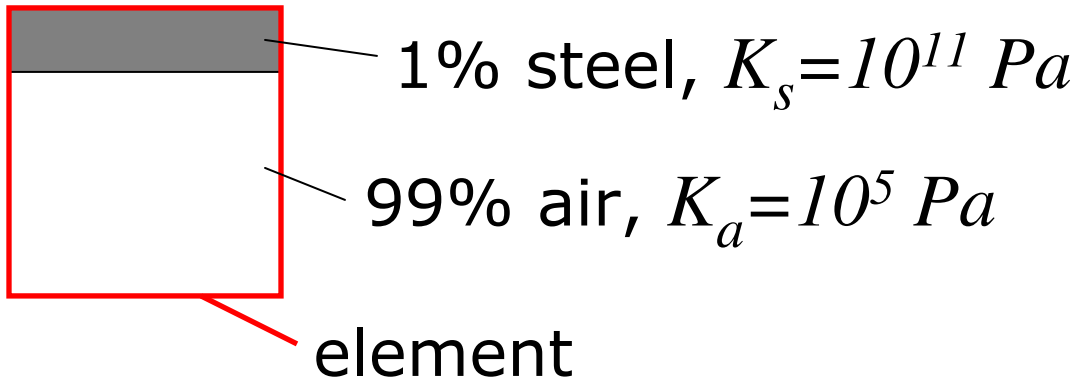
This simplification can cause troubles, such as dropping time steps and unrealistic strain levels.

Pressure equilibrium

The example below shows a situation where the default assumed uniform strain rate is bad.

Example:

An element contains a mixture of steel (1%) and air (99%)

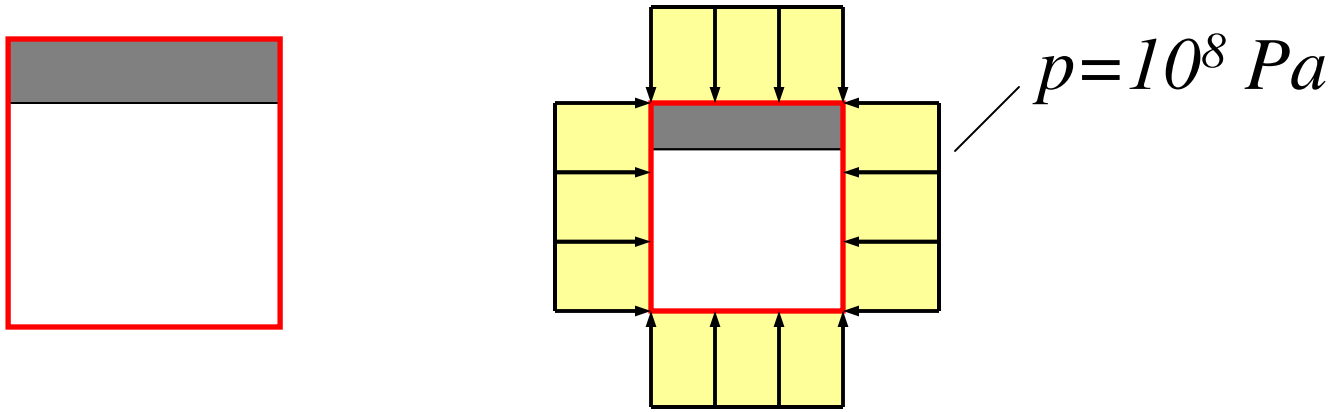


The composite bulk modulus of the element, K_c, is a volume fraction weighted average of the steel and of the air

$$K_c=0.01\cdot K_s+0.99\cdot K_a=10^9\ Pa$$

Pressure equilibrium

Assume that the element is exposed to a uniform pressure of $100\ MPa = 10^8\ Pa$



Assuming linear relationships, the applied pressure will lead to a volumetric compression of approximately 10%. Remember that both materials are exposed to the same deformation!

The pressure in the steel becomes, $p_s = 0.1 \cdot Ks = 10000\ MPa$.

This unrealistically high pressure might lead to dropping time steps and to strange results.

Pressure equilibrium

In reality, the air should compress more than the steel.

The pressure iteration algorithm assumes that the different materials have the same hydrostatic pressure.

That is, with the pressure iteration algorithm activated, different materials are exposed to different compressions.

volumetric strain rate of element

$$\operatorname{tr}\mathbf{D} = \sum_{k=1}^{n} \eta_k \operatorname{tr}\mathbf{D}_k$$

volume fraction of material k

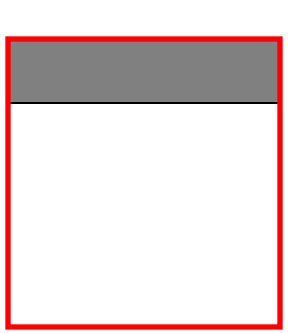
volume strain rate of material k

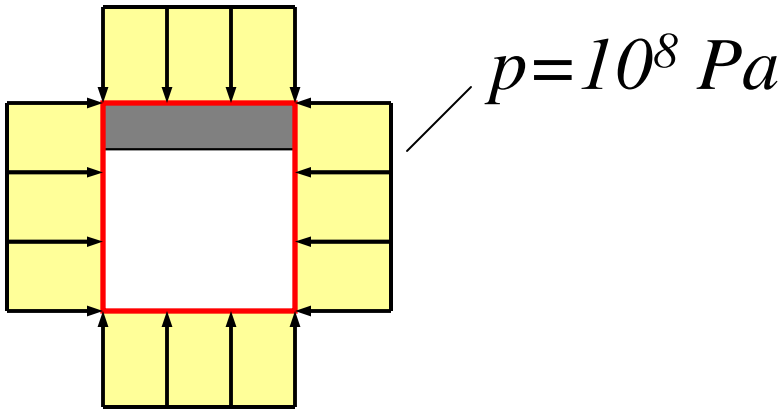
$$\frac{\operatorname{tr}\mathbf{D}_i}{\operatorname{tr}\mathbf{D}_j} = \frac{K_j}{K_i}$$

the volumetric compression of a material is proportional to one over the material bulk modulus

Pressure equilibrium

Activating the pressure iteration algorithm (**PRIT=1**), the steel-air example will behave differently.





$$p_s = p_a = p \Rightarrow$$

Steel compression = $p_s/K_s = 0.1\%$

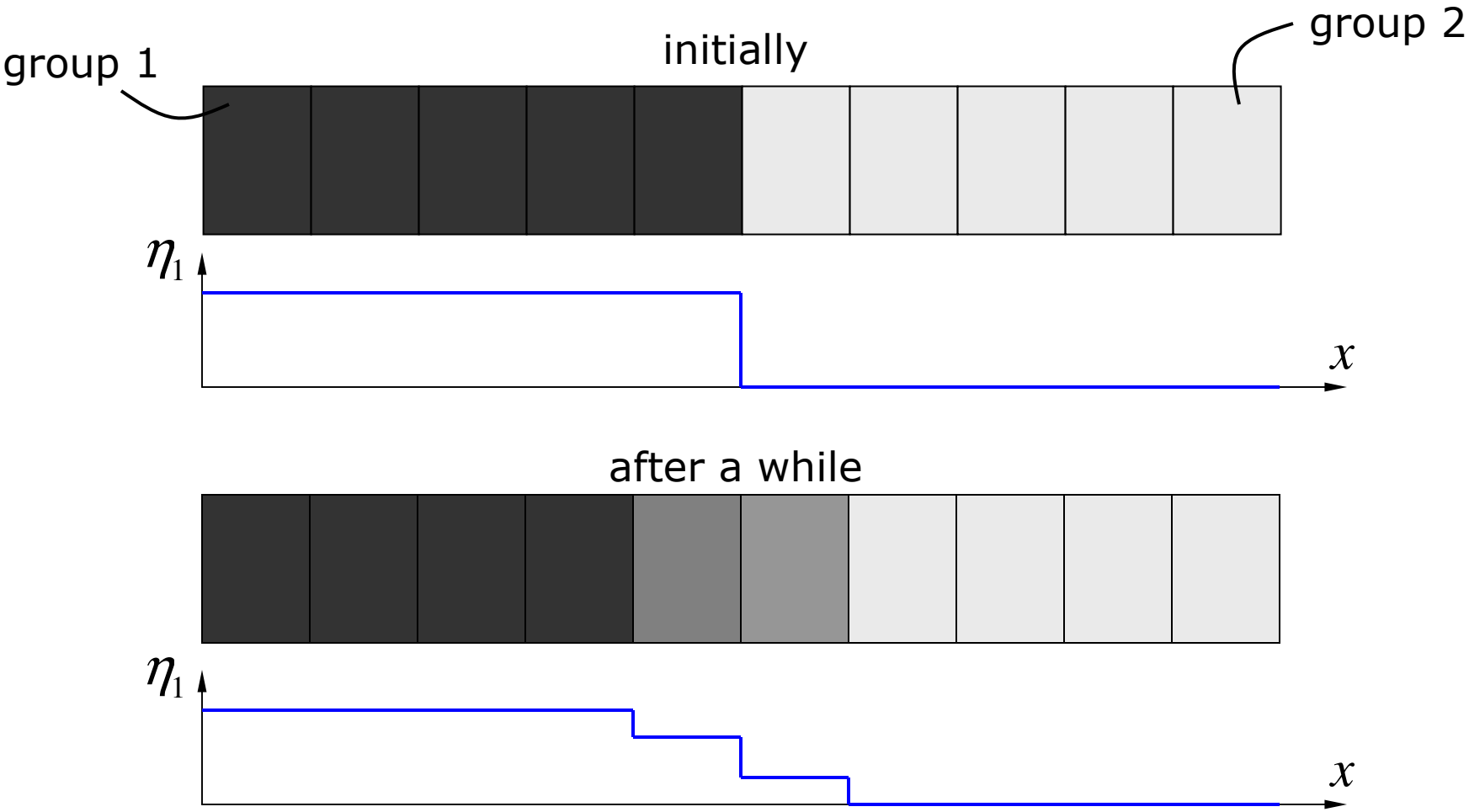
Air compression = $p_a/K_a = 10^5\ \%$ (assuming a constant bulk modulus)

***CONTROL_ALE**

DCT	NADV	METH	AFAC	BFAC	CFAC	DFAC
START	END	AAFAC	VFAC	**PRIT**	EBC	PREF

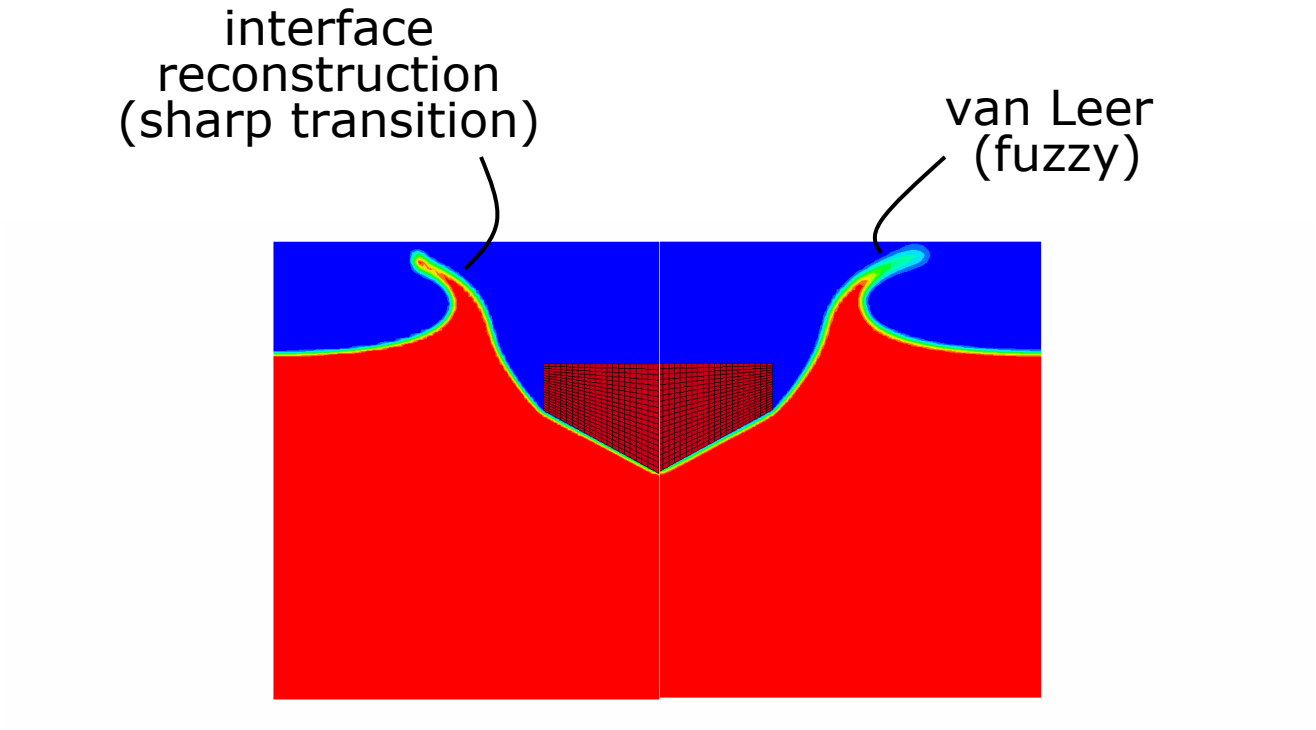
Interface reconstruction

As mass is fluxing between elements, the initially sharp material boundaries smear out. That is, there is a transition region where the volume fractions drop from 1 to 0.



Interface reconstruction

We want to keep the transition region as sharp as possible. For this reason, the volume fractions are advected differently from the other history variables. That is, we don't use the Donor cell or van Leer schemes to estimate the volume fraction flux.

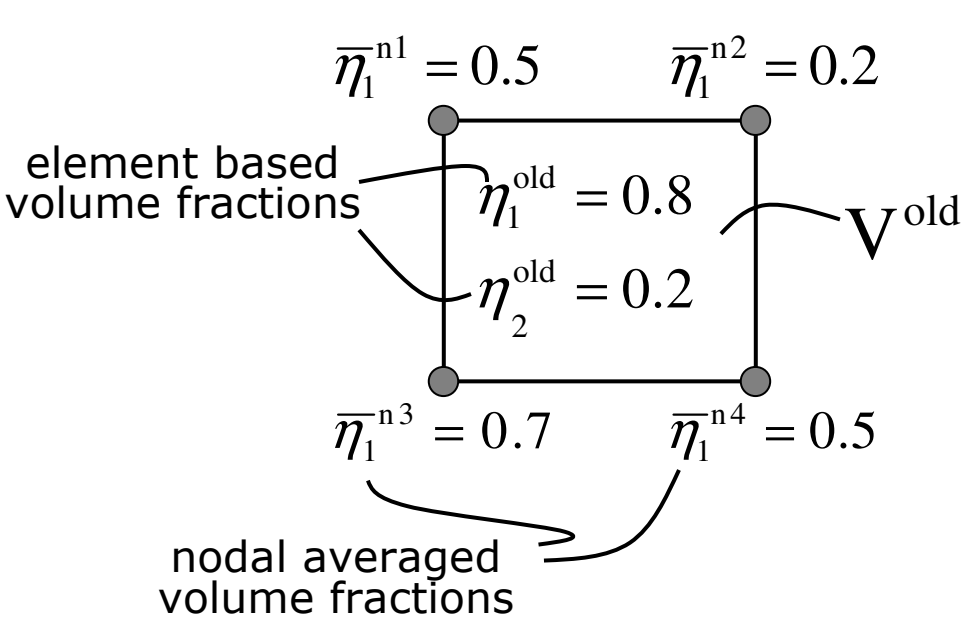


Interface reconstruction

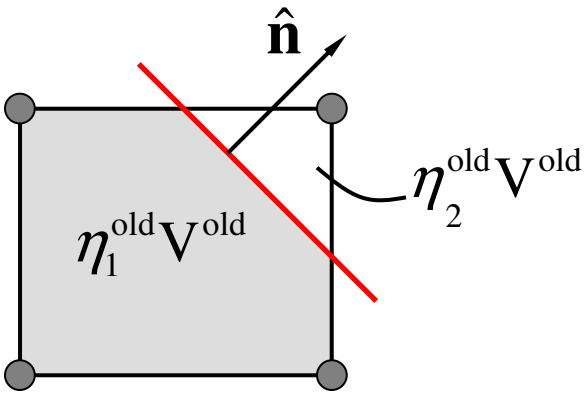
With the interface reconstruction, mixed elements are cut with a plane, separating the location of the different materials. The plane orientation is based on the gradient of the volume fraction field.

example with two materials

element before mesh smoothing



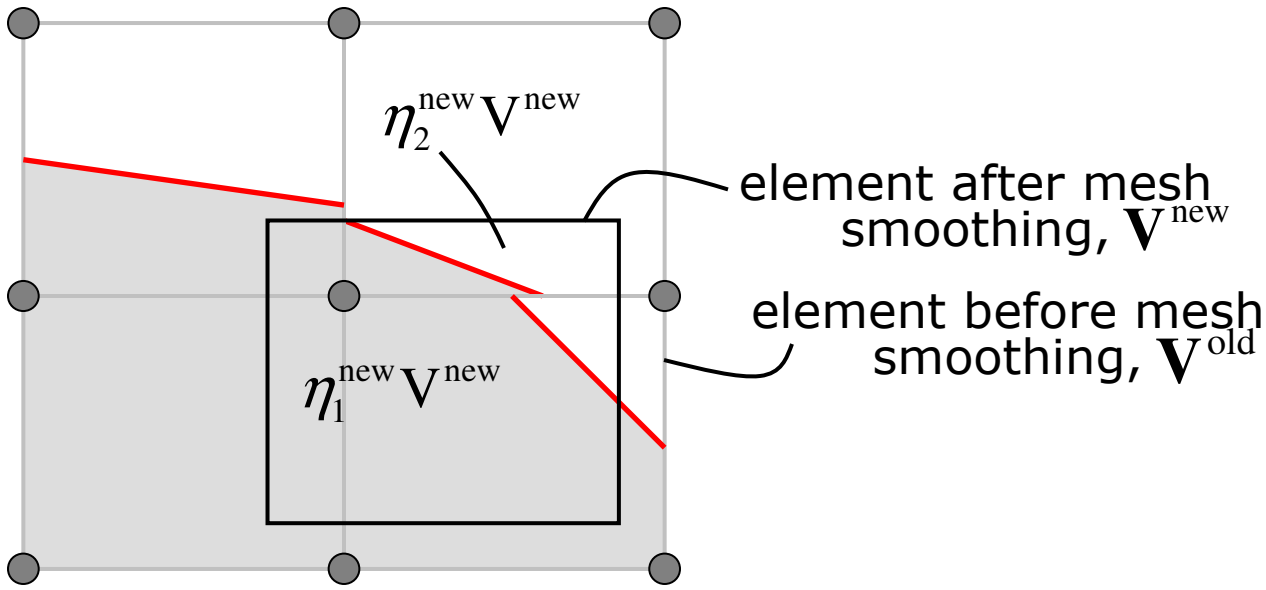
assumed distribution of materials



$$\hat{\mathbf{n}} = \left\| \frac{\partial \overline{\eta}_1}{\partial \mathbf{x}} \right\|^{-1} \cdot \frac{\partial \eta_1}{\partial \mathbf{x}} = \frac{1}{\sqrt{2}} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

Interface reconstruction

The element is moved during mesh smoothing. The volume fractions are updated based on the assumed material distribution.



*CONTROL_ALE

DCT	NADV	METH	AFAC	BFAC	CFAC	DFAC
START	END	AAFAC	VFAC	PRIT	EBC	PREF
			CHKR			

DCT (only active in 2D ALE)

NADV Number of time steps between mesh smoothing and advection

METH Advection method (1=Donor Cell, 2=van Leer)

AFAC, **BFAC**, **CFAC**, **DFAC** Classical mesh smoothing parameters

START Birth time for ALE

END Death time for ALE

AAFAC (obsolete)

VFAC Void factor for element formulation 12

PRIT Pressure equilibrium flag

EBC Flag for automatically applied BC's along the boundary of the ALE domain

PREF Reference pressure (applied to all free boundaries of the ALE mesh)

CHKR Checkerboard pattern prevention parameter

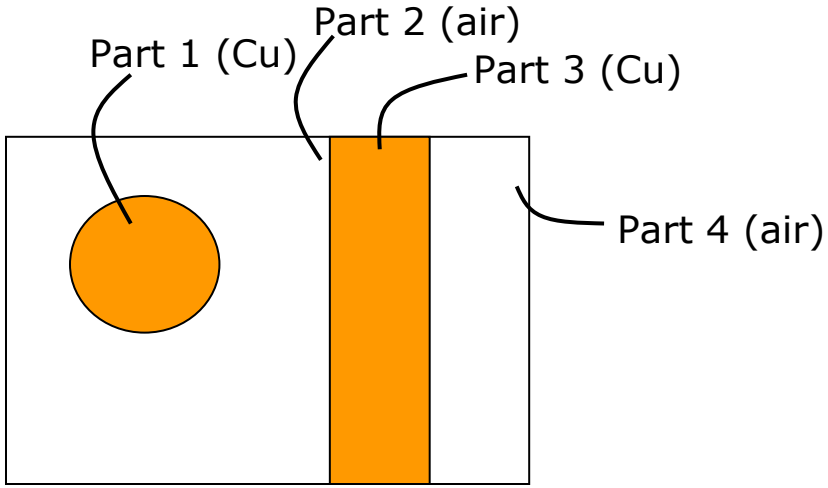
*ALE_MULTI-MATERIAL_GROUP

SID IDTYPE

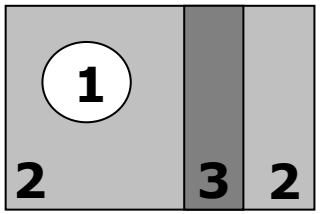
SID Part set ID or part ID

IDTYPE ID type (0=part set, 1=part)

This command groups parts together to define a multi-material group.




```
*ALE_MULTI-MATERIAL_GROUP
         1         1
      1234         0
         3         1
*SET_PART_LIST
      1234
         2         4
```



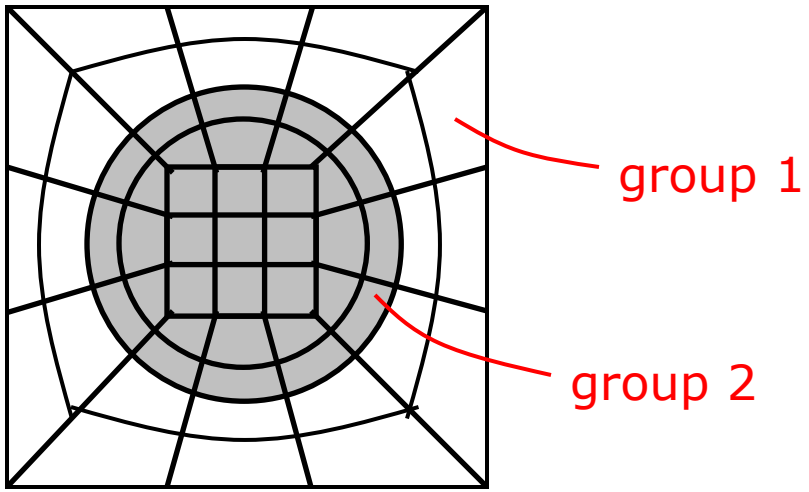
*INITIAL_VOLUME_FRACTION

EID VF1 VF2 VF3 VF4 VF5 VF6 VF7

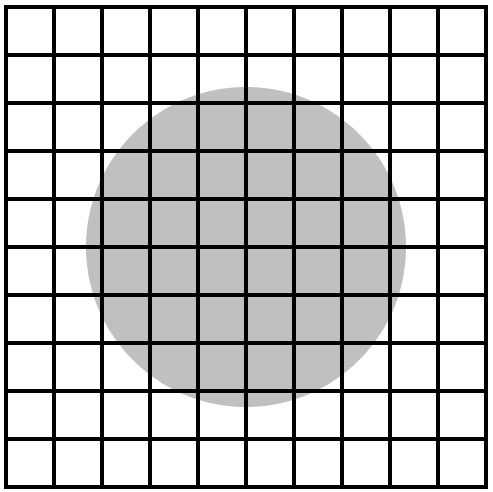
EID	Element ID
VF1	Volume fraction of multi-material group 1
VF2	Volume fraction of multi-material group 2
VF3	Volume fraction of multi-material group 3, etc.

This command is used to allow initially mixed elements. This can help avoiding complex meshing and bad aspect ratios.

conventional meshing



***INITIAL_VOLUME_FRACTION**



*INITIAL_VOLUME_FRACTION_GEOMETRY

SID STYPE DGID
GTYPE INOUT FGID
- geometry description -

SID	Set ID
STYPE	ID type (0=part set, 1=part)
DGID	Default multi-material group ID
GTYPE	Geometry type (1=shell, 2=segment, 3=plane, 4=cylinder, 5=box, 6=sphere)
INOUT	Fill inside or outside geometry (0=inside, 1=outside)
FGID	Multi-material group ID of added material

Each **GTYPE** has its own geometry description, see keyword manual for details.

This command is used to generate initial volume fractions for arbitrary geometries. A body is defined as combinations of simple primitives and shell/segment geometries.

Multi-material ALE formulation

- Motivation and overview
- Keyword commands

  ***ALE_REFERENCE_SYSTEM_GROUP**
  ***ALE_REFERENCE_SYSTEM_NODE**
  ***ALE_REFERENCE_SYSTEM_CURVE**
  ***ALE_REFERENCE_SYSTEM_SWITCH**

- Input deck examples

  dropping box
  bird strike 1
  bird strike 2
  bird strike 3
  detonation in air

Motivation and overview

By moving the fluid mesh, the mass flux between elements can be decreased and numerical dissipation errors can be minimized.

In LS-DYNA there are several ways of moving a mesh that has been set up for a multi-material formulation. Two of the methods have been described previously, in the single material ALE section.

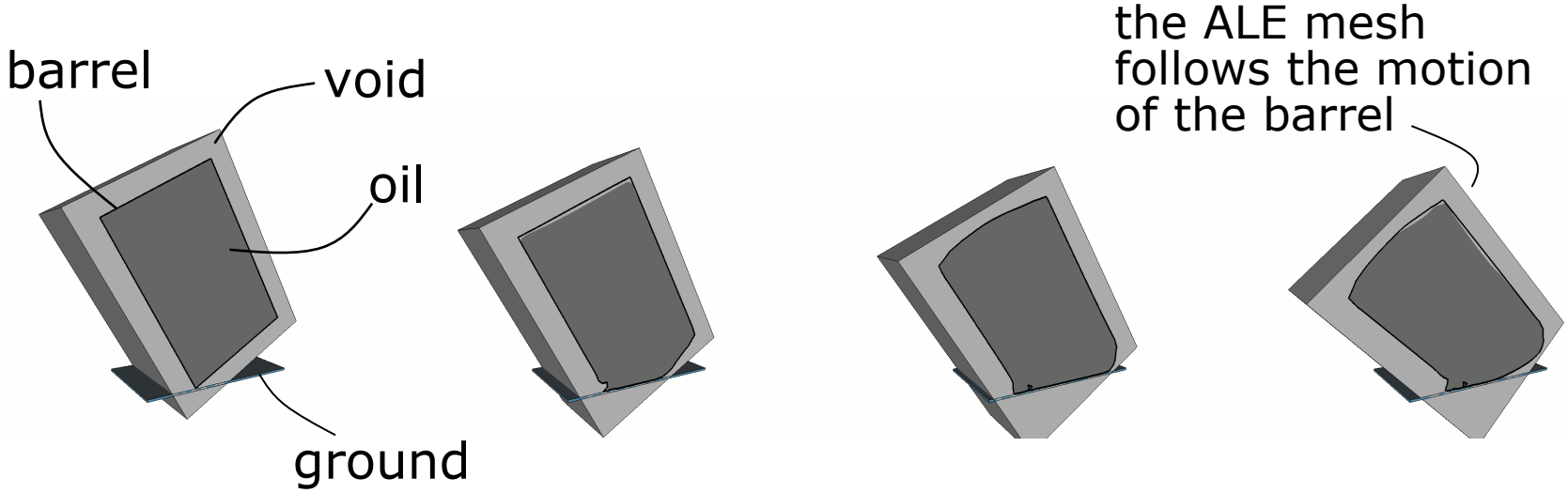
The ALE mesh motion is, to a large extent, controlled with the keyword command ***ALE_REFERENCE_SYSTEM_GROUP**. The structure of the keyword is described in this section.

Motivation and overview

LS-DYNA example

drop test of an oil barrel

complete model



barrel only

*ALE_REFERENCE_SYSTEM_GROUP

SID	**STYPE**	**PRTYPE**	**PRID**	**BCTRAN**	**BCEXP**	**BCROT**	**ICOORD**
XC	**YC**	**ZC**	**EXPLIM**	**EFAC**			

SID Set ID

STYPE Set type

PRTYPE Reference system type

PRID ID of switch list, node group, curve group or part set

BCTRAN Translational constraints

BCEXP Mesh expansion constraints

BCROT Mesh rotation constraints

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/shrinkage

***ALE_REFERENCE_SYSTEM_GROUP**

SID	STYPE	PRTYPE	PRID	BCTRAN	BCEXP	BCROT	ICOORD
XC	YC	ZC	EXPLIM	EFAC			

PRTYPE

0 Eulerian

1 Lagrangian

2 Classical 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 specified nodes (see ***ALE_REFERENCE_SYSTEM_NODE**)

6 Switching in time between different reference system types
(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 Contract the mesh in the vicinity of shock fronts

9 Automatic mesh expansion in order to enclose a set of parts

***ALE_REFERENCE_SYSTEM_CURVE**

```
ID
LC1     LC2     LC3     LC4     LC5     LC6     LC7     LC8
LC9     LC10    LC11    LC12
```

ID Curve set ID

LC1 … **LC12** Load curve ID's

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

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

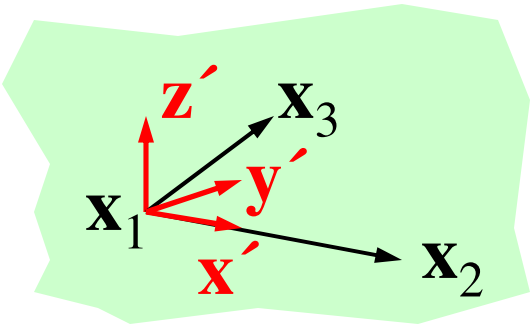
$f_1(t)$ is the value of load curve **LC1** at time t, etc.

*ALE_REFERENCE_SYSTEM_NODE

```
ID
NID1   NID2   NID3   NID4   NID5   NID6   NID7   NID8
NID9   NID10  NID11  NID12
```

ID Node group ID

NID1
:
NID12 } User specified nodes



For **PRTYPE=3** 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, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, 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_SWITCH

```
ID
T1      T2      T3      T4      T5      T6      T7
TYPE1   TYPE2   TYPE3   TYPE4   TYPE5   TYPE6   TYPE7   TYPE8
ID1     ID2     ID3     ID4     ID5     ID6     ID7     ID8
```

ID Switch list ID

T1 … **T7** Times for switching reference system type

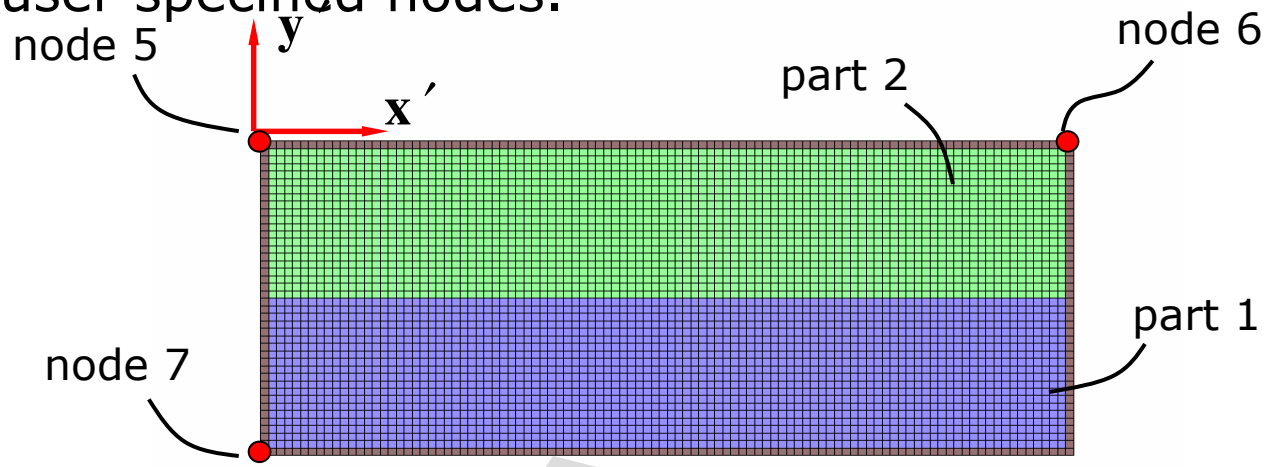
TYPE1 … **TYPE8** Reference system types

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

At time **T2** the reference system type is switched from **TYPE2** to **TYPE3** etc. See ***ALE_REFERENCE_SYSTEM_GROUP** for information about the different reference system types.

LS-DYNA example
dropping box

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




```
*ALE_REFERENCE_SYSTEM_GROUP
         1          0          5          1

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

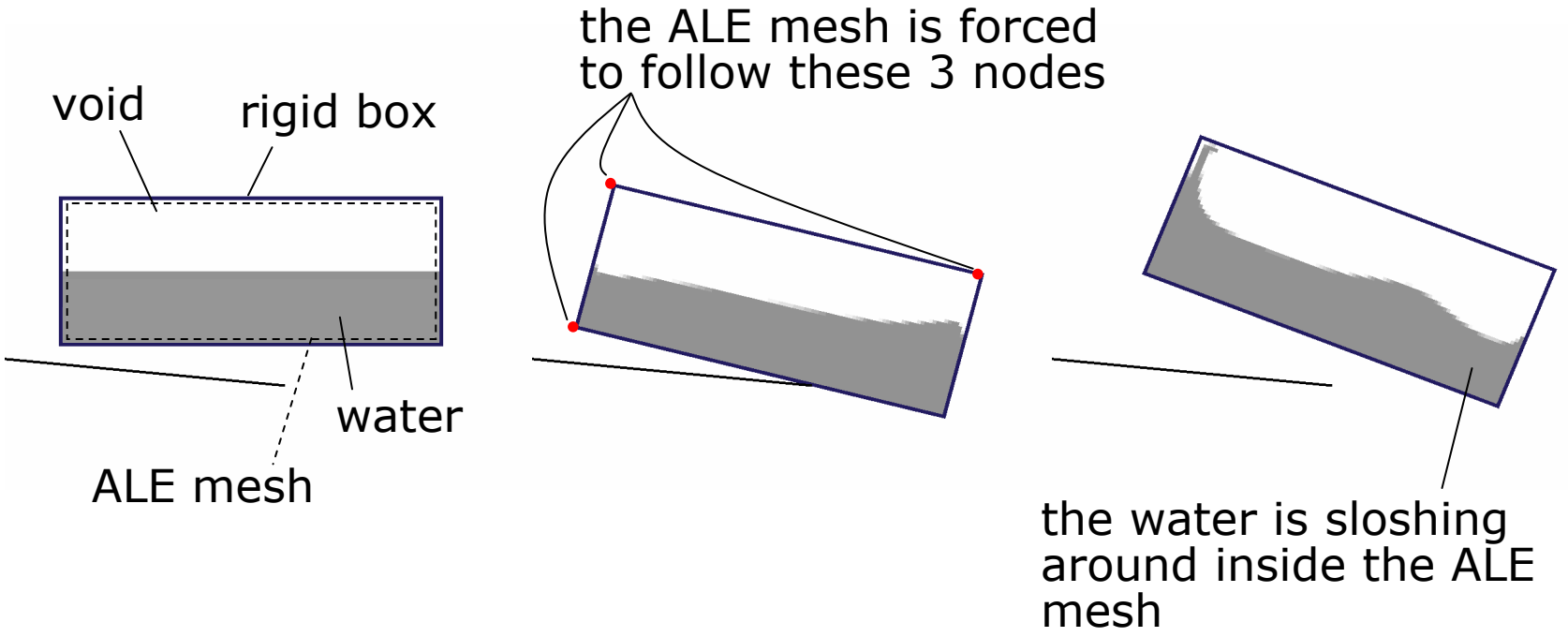
$$\mathbf{x}' = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

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

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

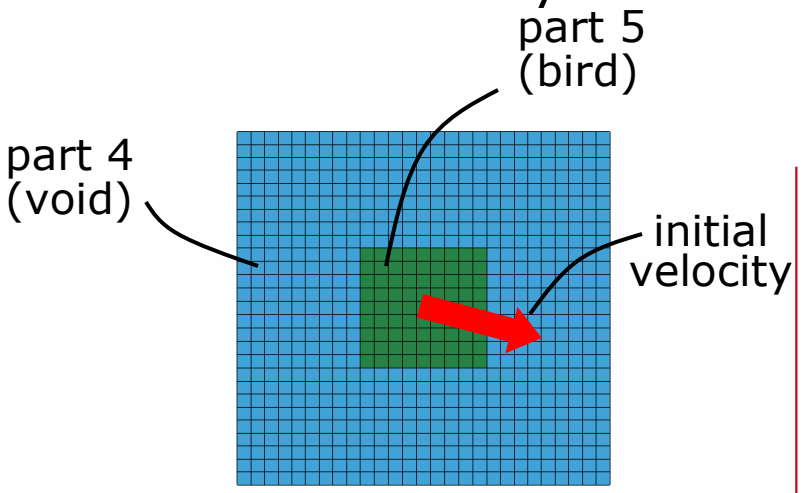
LS-DYNA example
dropping box

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



LS-DYNA example
bird strike 1

Example of spatially constant/linear ALE mesh velocity field, based on the current mass flow velocity distribution.




```
*ALE_REFERENCE_SYSTEM_GROUP
1          0          4          0          0          7          0

*SET_PART_LIST
1
4          5
```

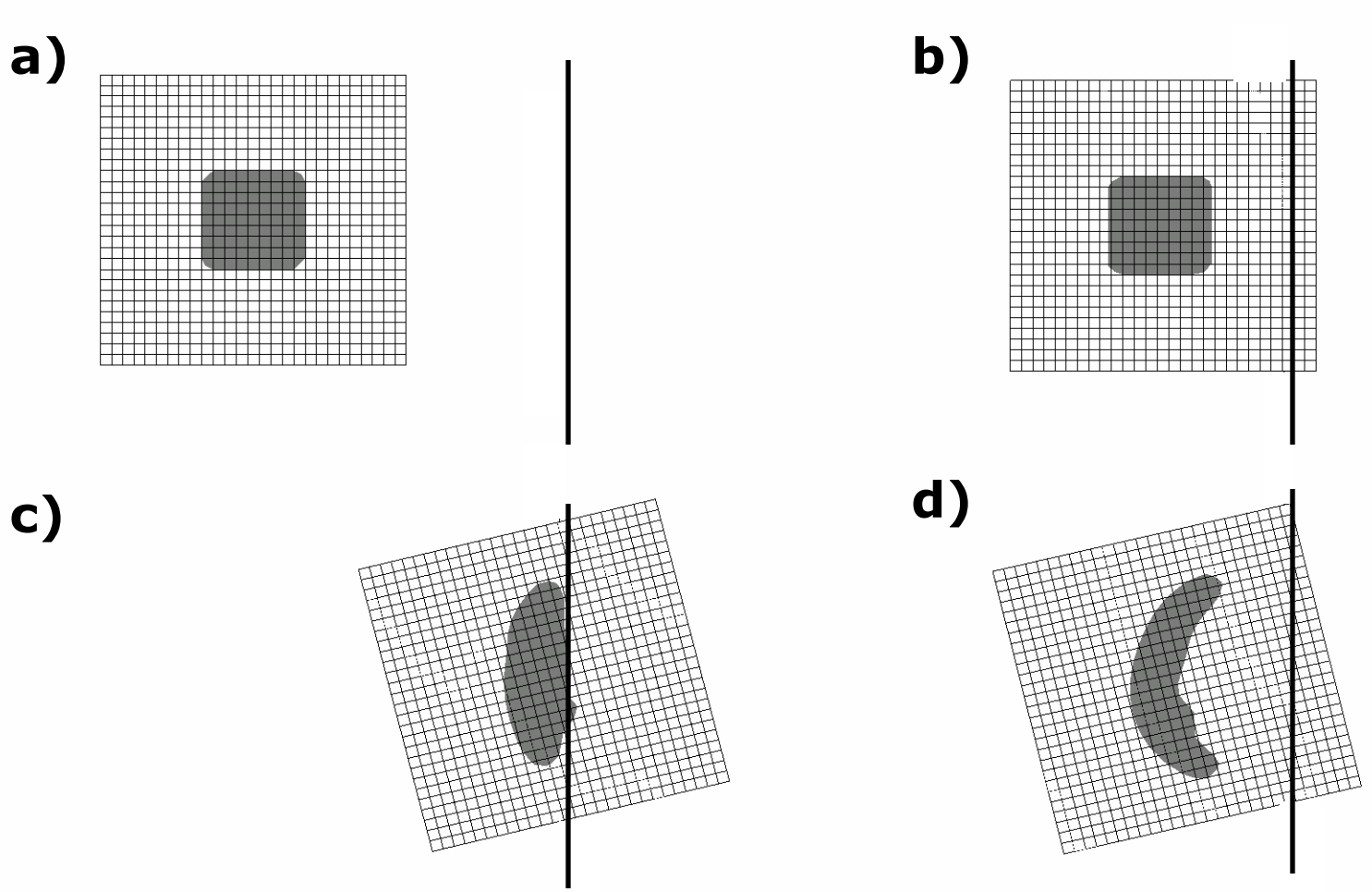
no rotational constraints

expansion not allowed

no translational constraints

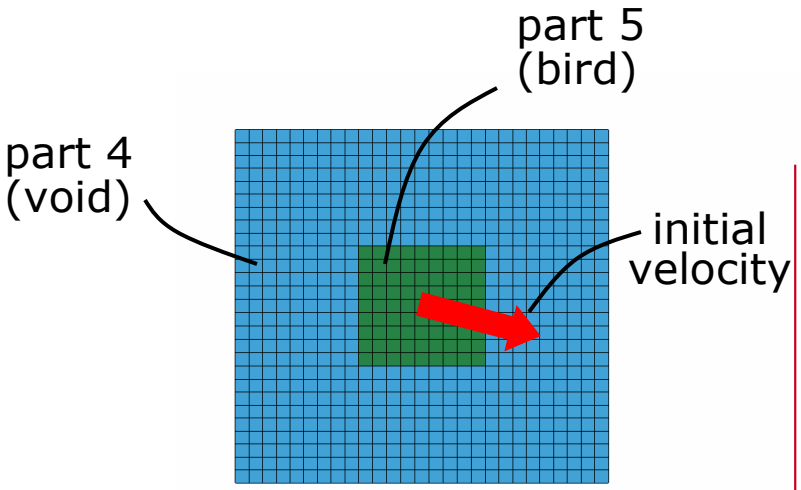
LS-DYNA example

bird strike 1



LS-DYNA example
bird strike 2

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



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

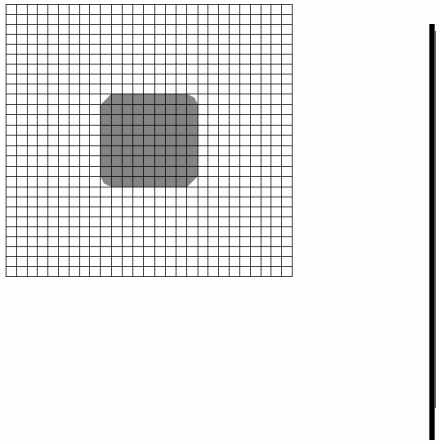
```
*ALE_REFERENCE_SYSTEM_GROUP
1          0          3          1
*ALE_REFERENCE_SYSTEM_CURVE
1
1          0          0          0          2

*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

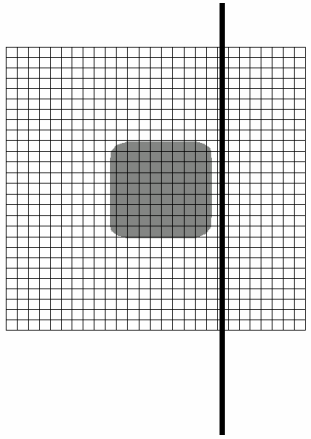
curve ID for f_5

LS-DYNA example
bird strike 2

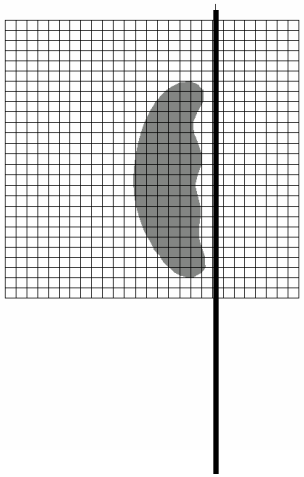
a)



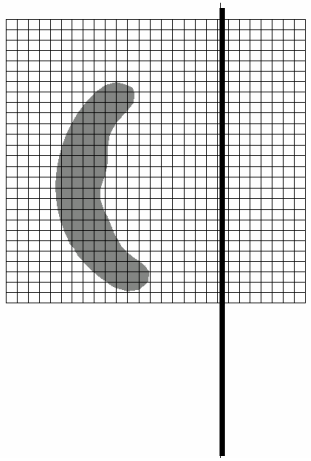
b)



c)

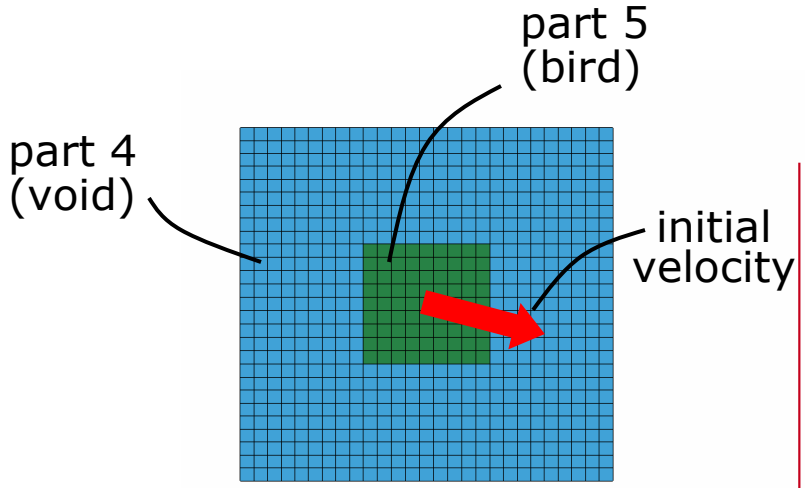


d)



LS-DYNA example
bird strike 3

Example of switching between reference system types.



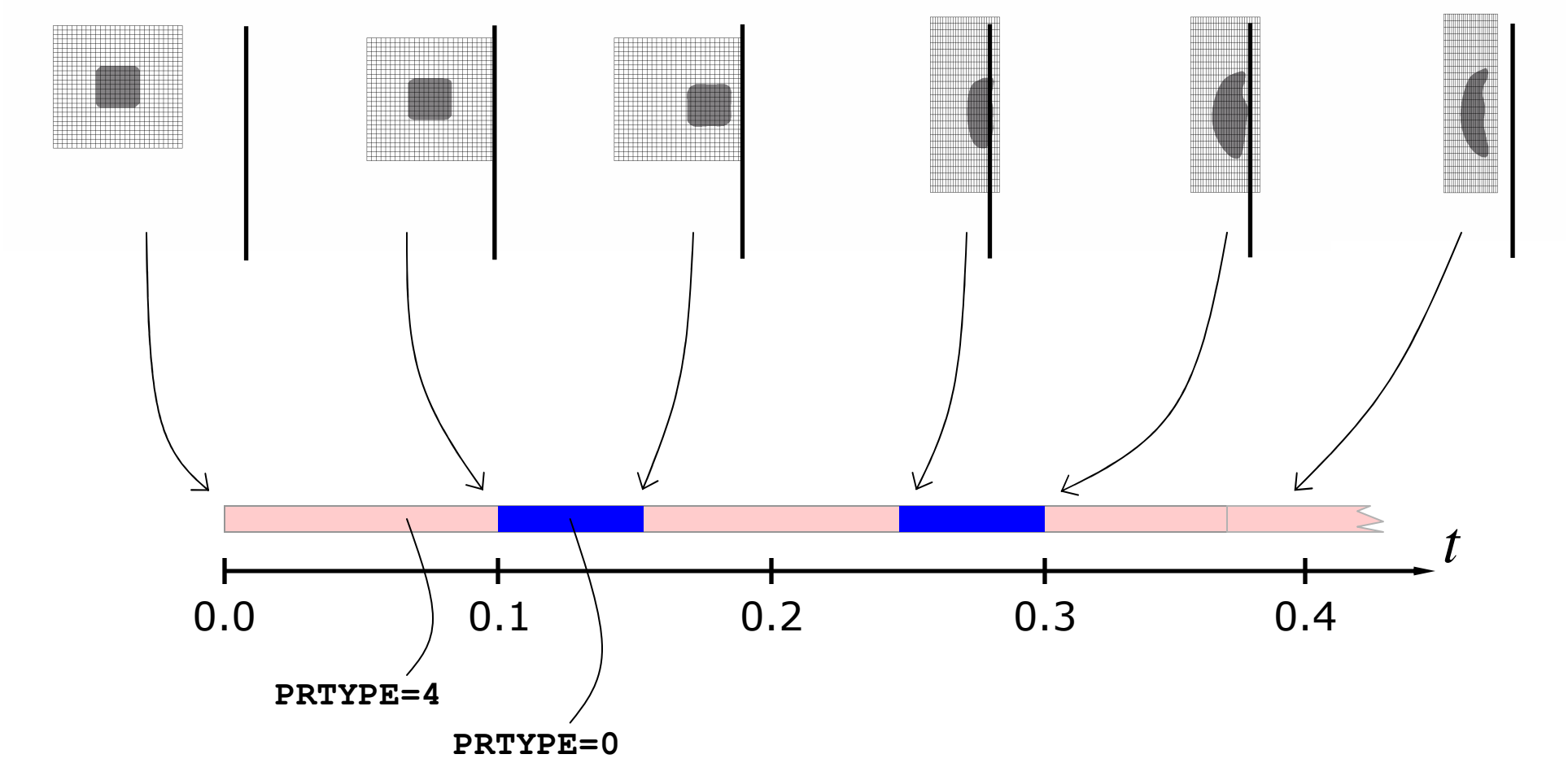

```
*ALE_REFERENCE_SYSTEM_GROUP
1          0          6       1234          0          0          7
*ALE_REFERENCE_SYSTEM_SWITCH
1234
0.1     0.15       0.25       0.30       10.0
4          0          4          0          4

*SET_PART_LIST
1
4          5
```

motion following
mass flow

Eulerian description of motion

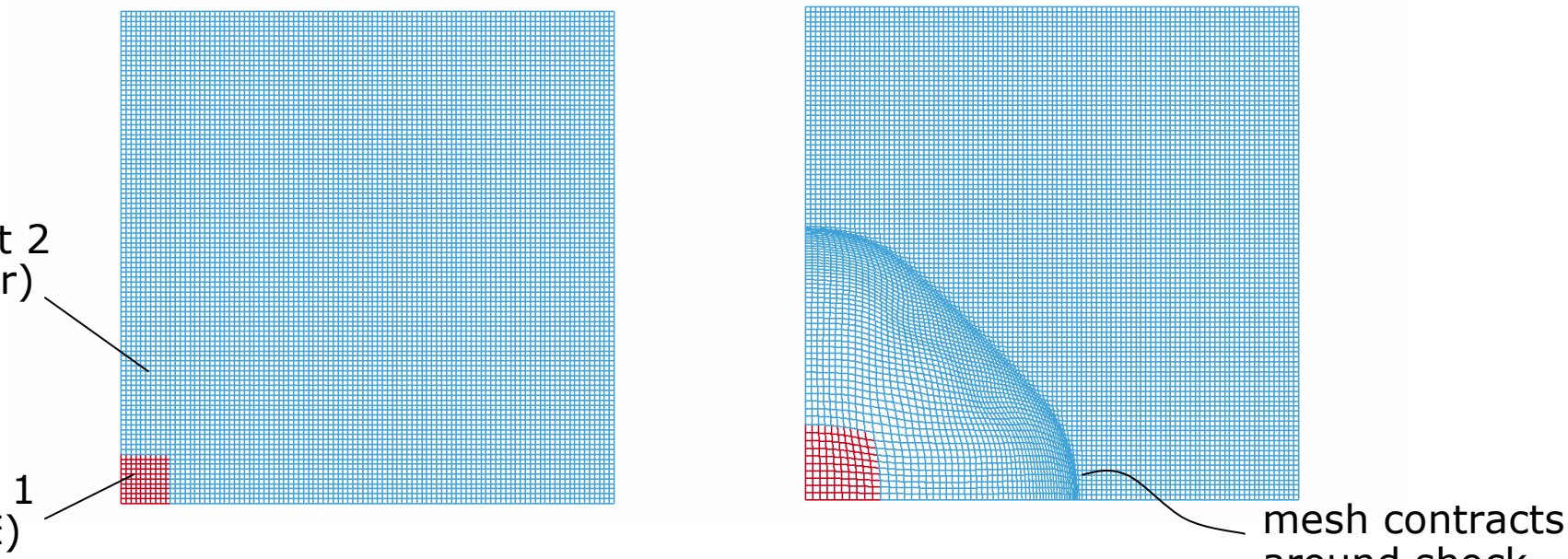
LS-DYNA example
bird strike 3

Example of switching between reference system types.



LS-DYNA example
Detonation in air

Example of delayed mesh relaxation (PRTYPE=8).




```
*ALE_REFERENCE_SYSTEM_GROUP
         1         0         8
         0         0         0         0       0.01
*SET_PART_LIST
         1
         1         2
```

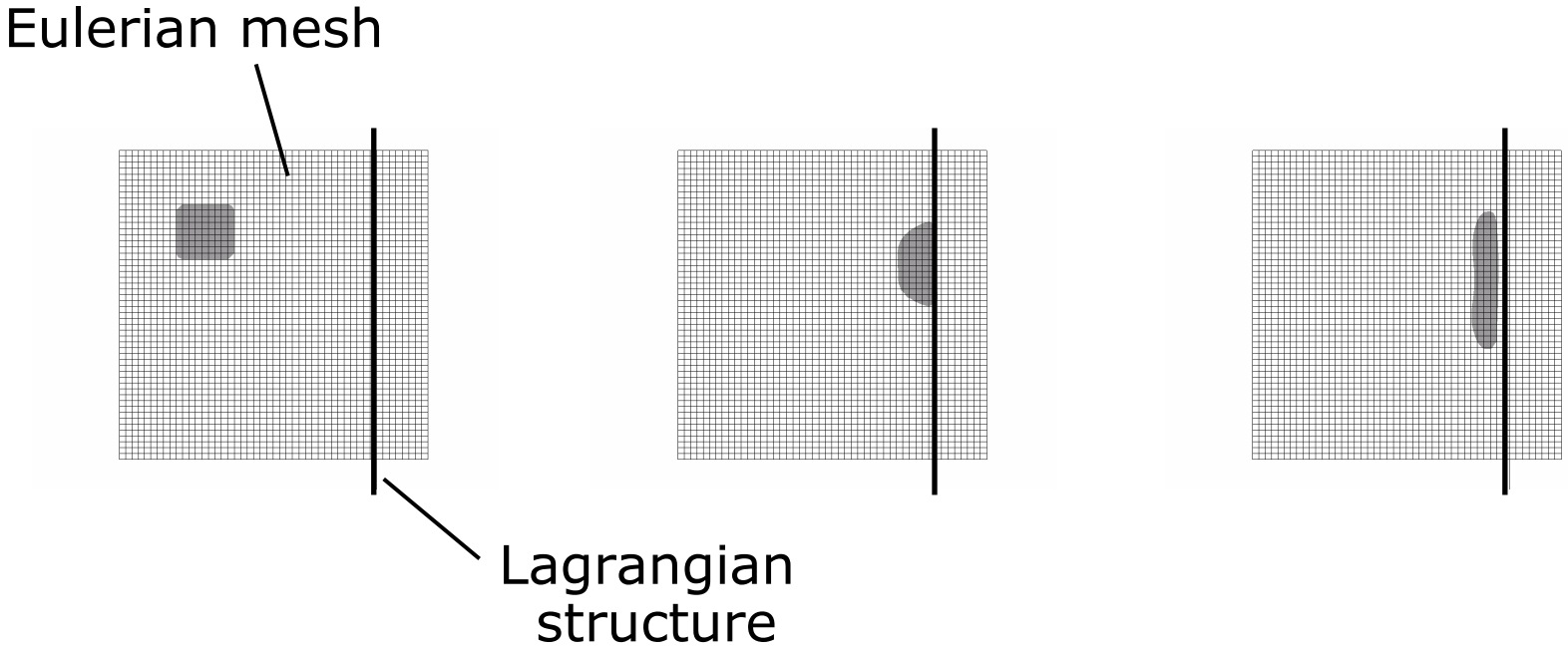
Fluid-structure interaction

- Introduction
- Constraint based method
- ***ALE_FSI_PROJECTION**
- Penalty based method
- ***CONSTRAINED_LAGRANGE_IN_SOLID**
- ***DATABASE_FSI**

Introduction

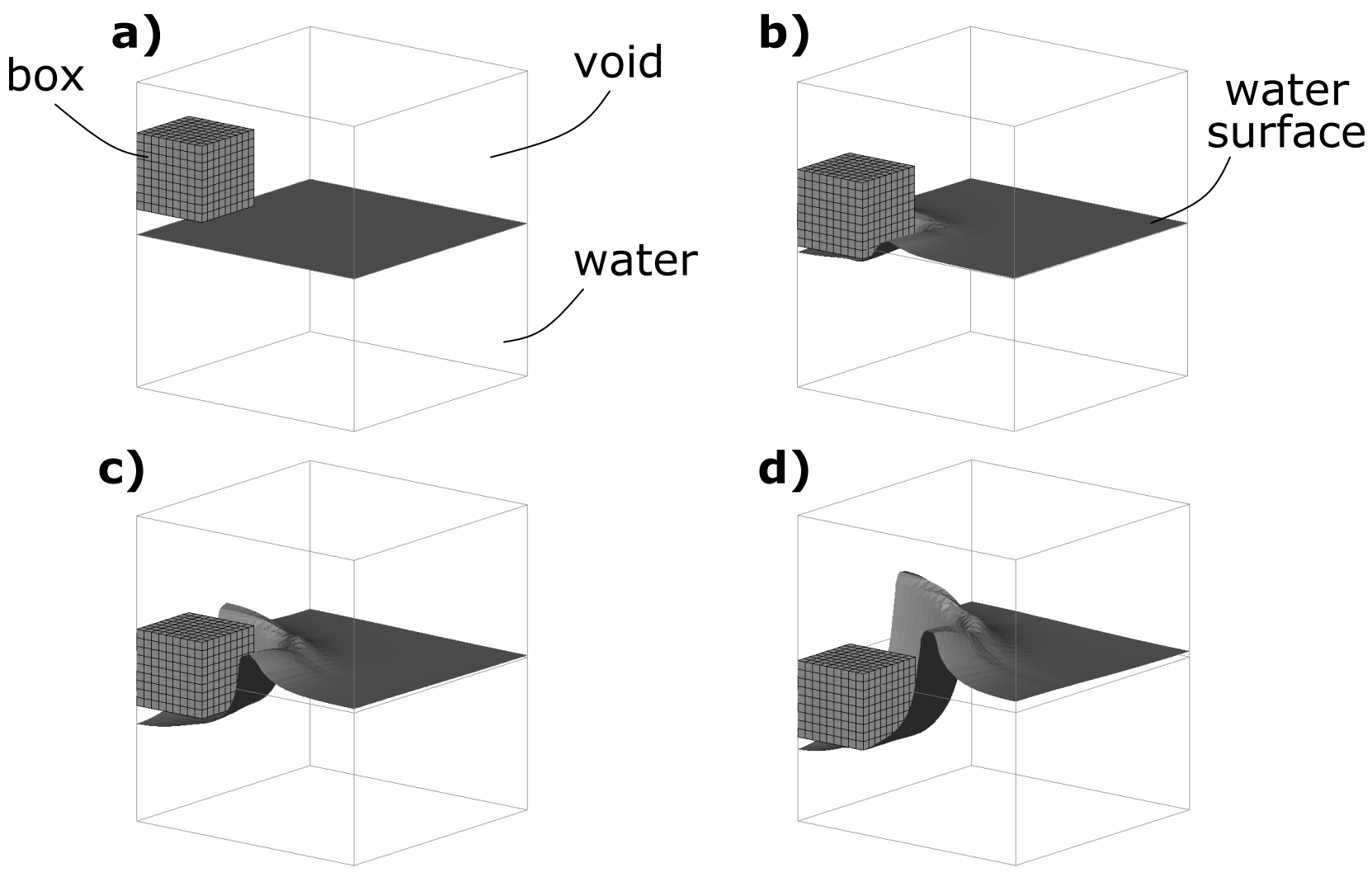
It is quite often suitable to treat parts of a model as Lagrangian and other parts as Eulerian or with an ALE-formulation.

A fluid-structure coupling algorithm is needed for the communication between the different parts.



Introduction

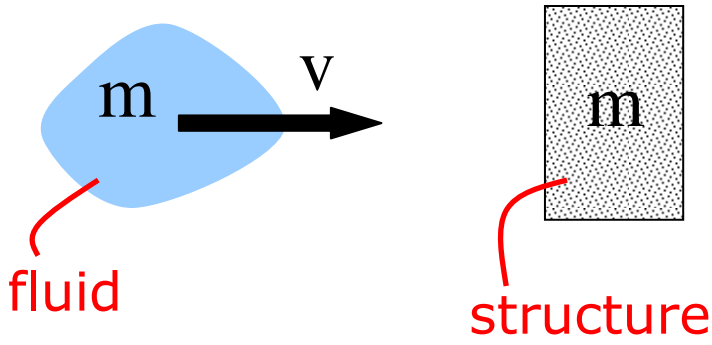
Example of a box hitting a water surface.



Introduction

Constraint based method

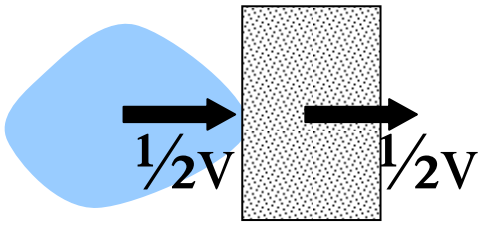
Before impact



$$I_0 = mv$$

$$W_0 = \frac{1}{2} mv^2$$

After impact



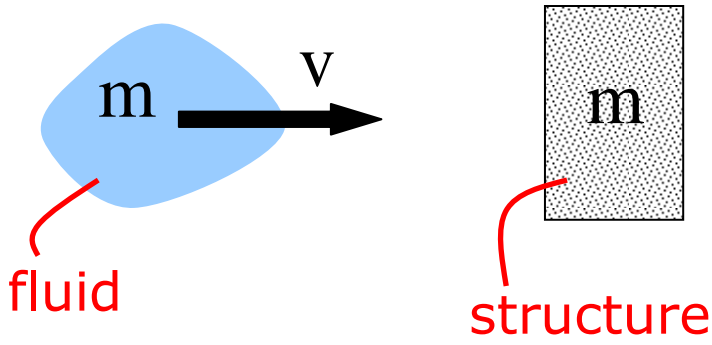
$$I_1 = mv = I_0$$

$$W_1 = \frac{1}{4} mv^2 < W_0$$

Introduction

Penalty based method

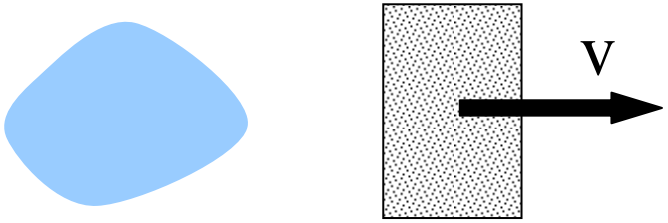
Before impact



$$I_0 = mv$$

$$W_0 = \frac{1}{2}mv^2$$

After impact

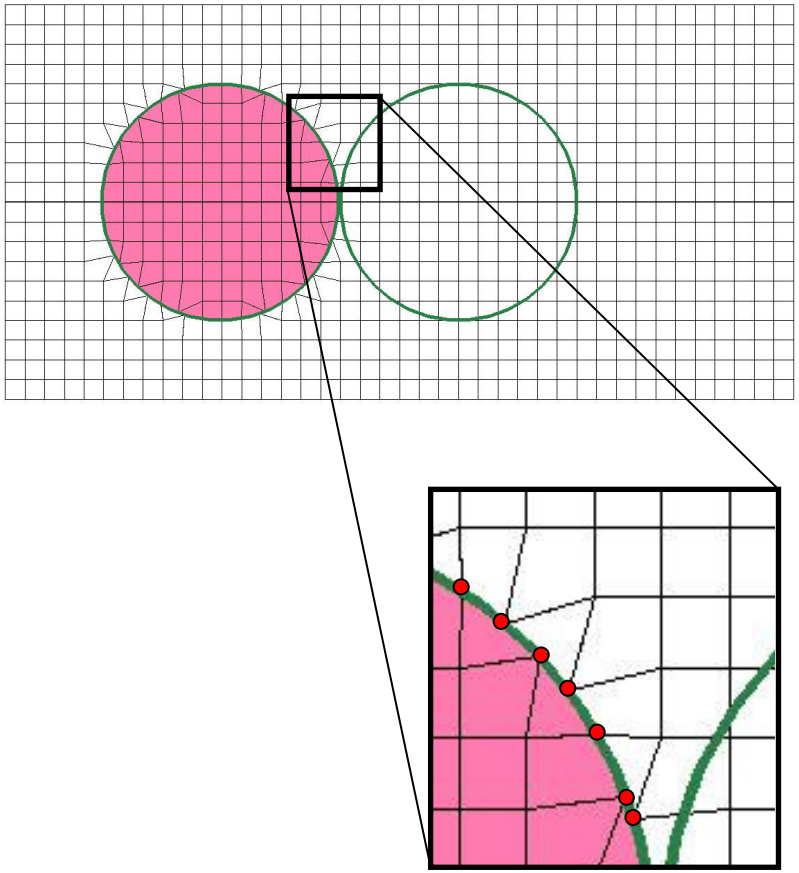


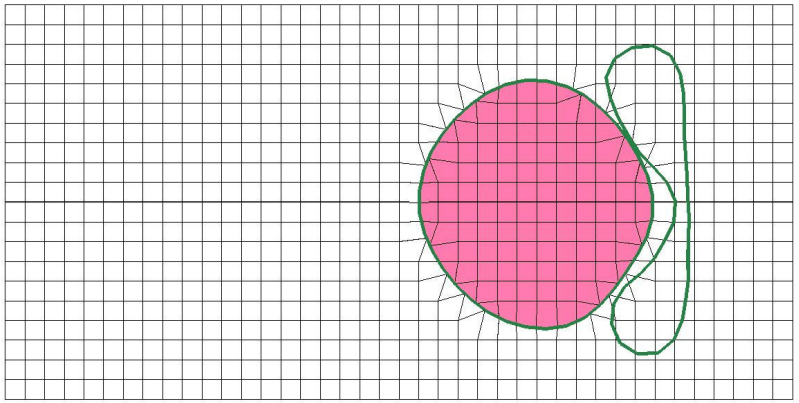
$$I_1 = mv = I_0$$

$$W_1 = \frac{1}{2}mv^2 = W_0$$

Constraint based method

*ALE_FSI_PROJECTION





● =projected fluid nodes

1. Fluid node momentum, mass and force is lumped on the structure.
2. Velocity update (central difference)
3. Projected fluid nodes are forced to follow the motion of the structure.

*ALE_FSI_PROJECTION

SSID FSID SSTYP FSTYP MMSID ICORR DIREC

SSID Structure set ID

FSID Fluid set ID

SSTYP Structure set type

- Eq. 0: part set
- Eq. 1: part

FSTYP Fluid set type

- Eq. 0: part set
- Eq. 1: part

MMSID Multi-material set ID of inside materials

ICORR Advection error correction

- Eq. 1: move mass from wrong side of coupling surface to the correct side
- Eq. 2: delete mass on the wrong side of the coupling surface
- Eq. 3: no correction at all (allow mass on the wrong side)

DIREC Coupling direction

- Eq. 0: all directions
- Eq. 1: normal direction (tension and compression)
- Eq. 2: normal direction (compression only)

Constraint based method

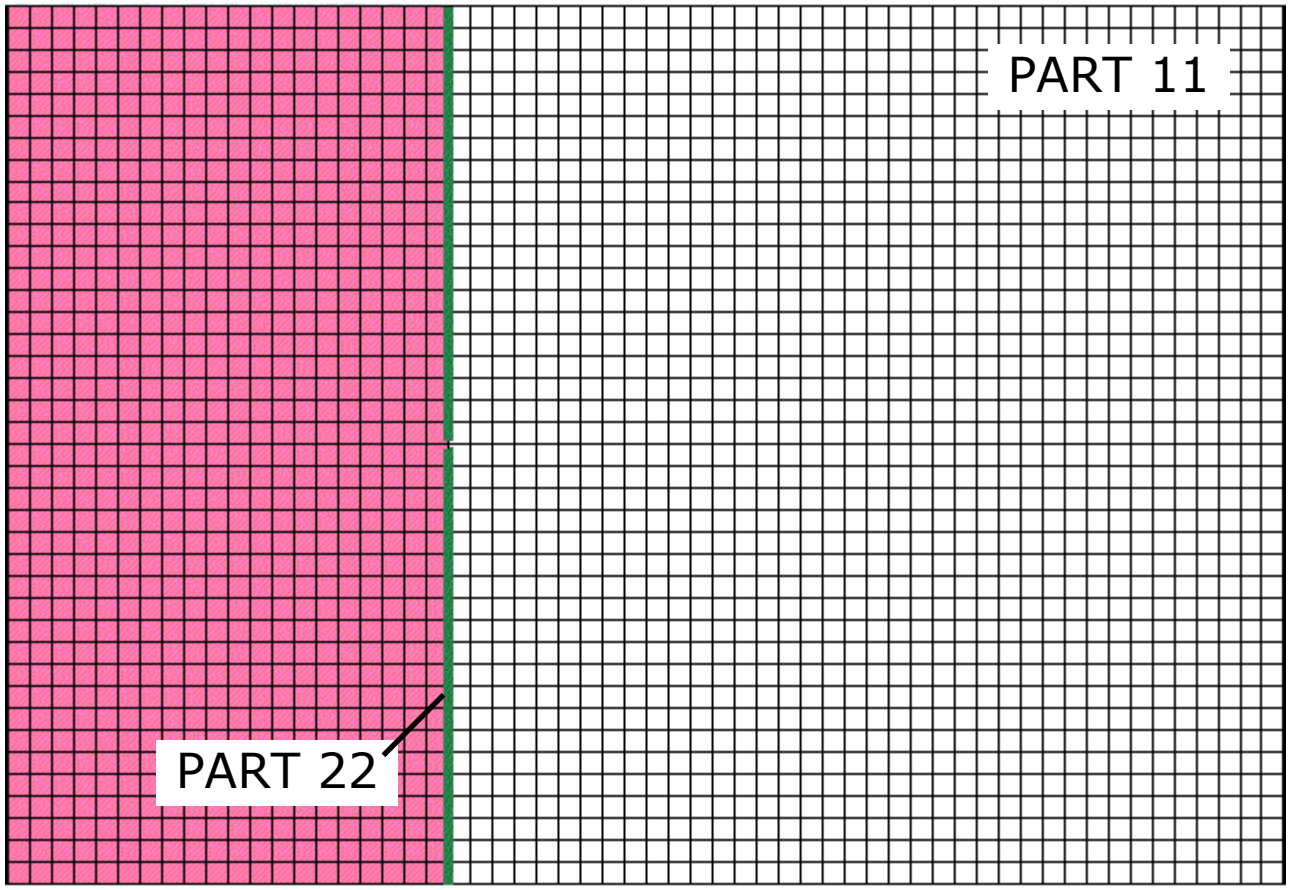
Example

```
*ALE_FSI_PROJECTION
22, 11, 1, 1, 0, 3, 0
```

no advection error correction =>
=> **MMSID** not needed

fluid
part ID

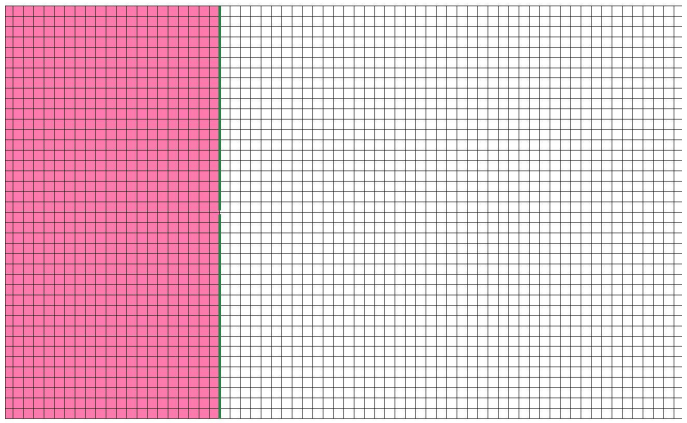
structure
part ID



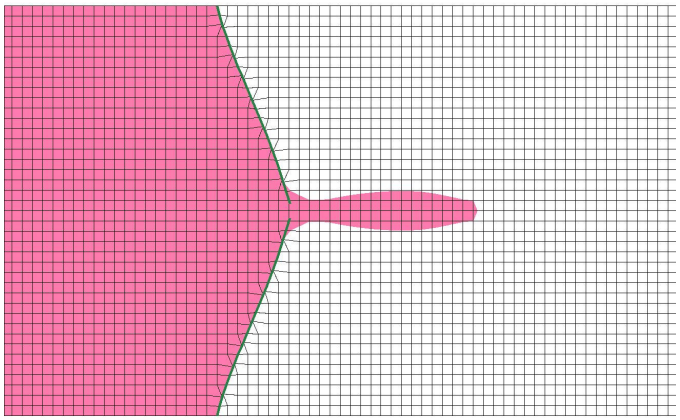
Constraint based method

Example

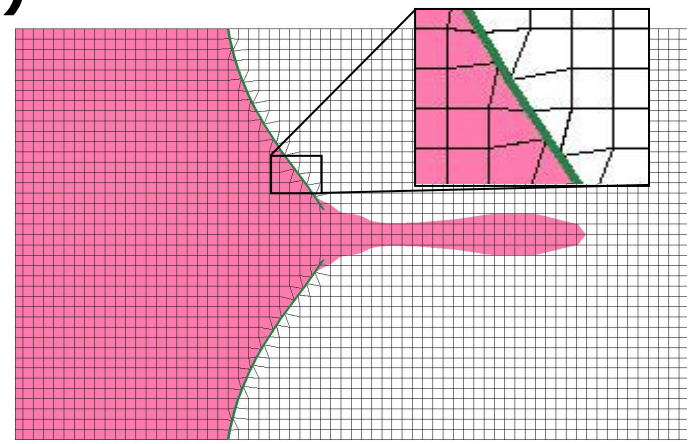
a)



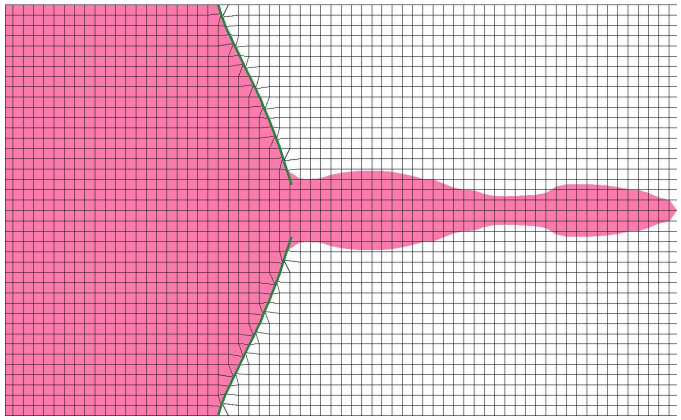
b)



c)



d)

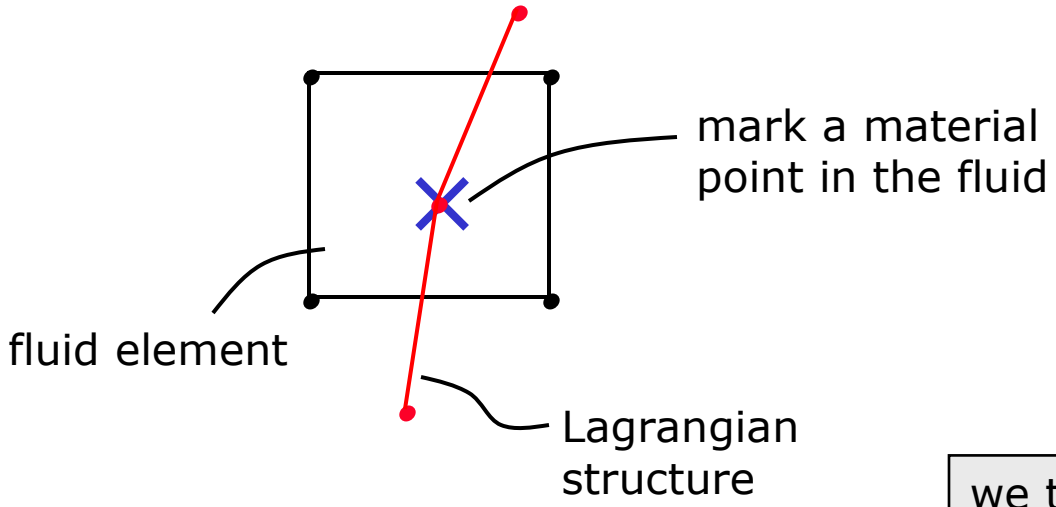


Penalty based method

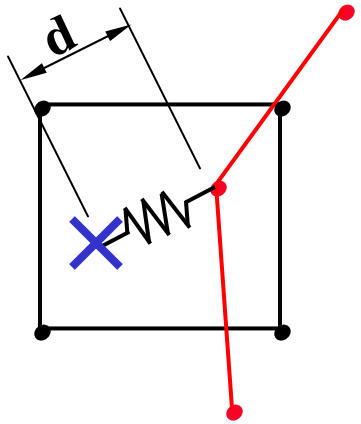
*CONSTRAINED_LAGRANGE_IN_SOLID

The penalty based algorithm tracks the relative displacement between fluid and the structure. Node forces, proportional to the magnitude of the relative displacements, are applied forcing the fluid and structure to follow each other. The method conserves energy but is generally noisier than the constraint based one.

coupling starts



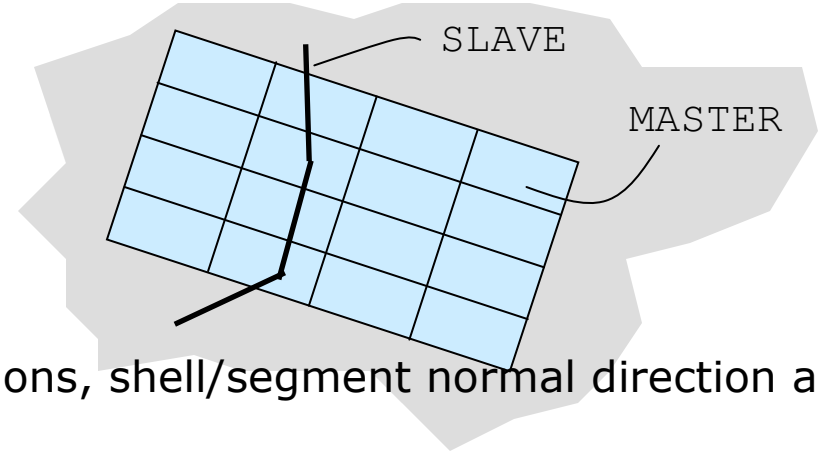
later



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

*CONSTRAINED_LAGRANGE_IN_SOLID

SLAVE	**MASTER**	**SSTYP**	**MSTYP**	**NQUAD**	**CTYPE**	**DIREC**	**MCOUP**
START	**END**	**PFAC**	**FRIC**	**FRCMIN**	**NORM**	**NORMTP**	**CDAMP**
THERM	**HMIN**	**HMAX**	**ILEAK**	**PLEAK**			



SLAVE Slave ID

MASTER Master ID

SSTYP Slave ID type

MSTYP Master ID type

NQUAD Quadrature rule

CTYPE Coupling type

DIREC Switch between coupling in all directions, shell/segment normal direction and in compression only

MCOUP Switch between coupling with all multi-material group and with the highest density one only (or with a set of multi-material groups)

START Birth time for coupling

END Death time for coupling

PFAC Penalty factor

FRIC Coefficient of friction

FRCMIN Minimum volume fraction to activate coupling

NORM Switch between right hand rule and left hand rule for shell/segment normals

NORMTP Node averaged segment normals

CDAMP Coupling damping (fraction of critical damping)

*CONSTRAINED_LAGRANGE_IN_SOLID

SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
START	END	PFAC	FRIC	FRCMIN	NORM	NORMTP	CDAMP
THERM	HMIN	HMAX	ILEAK	PLEAK			

THERM
HMIN
HMAX
} Coefficients for heat transfer in coupling interface

ILEAK Leakage control flag

PLEAK Leakage control factor

Penalty based method

Leakage

In certain situations, or if the coupling cards are not properly specified, there can be a leakage through the coupling interface.

Two common reasons are:

1. The coupling grid is too coarse

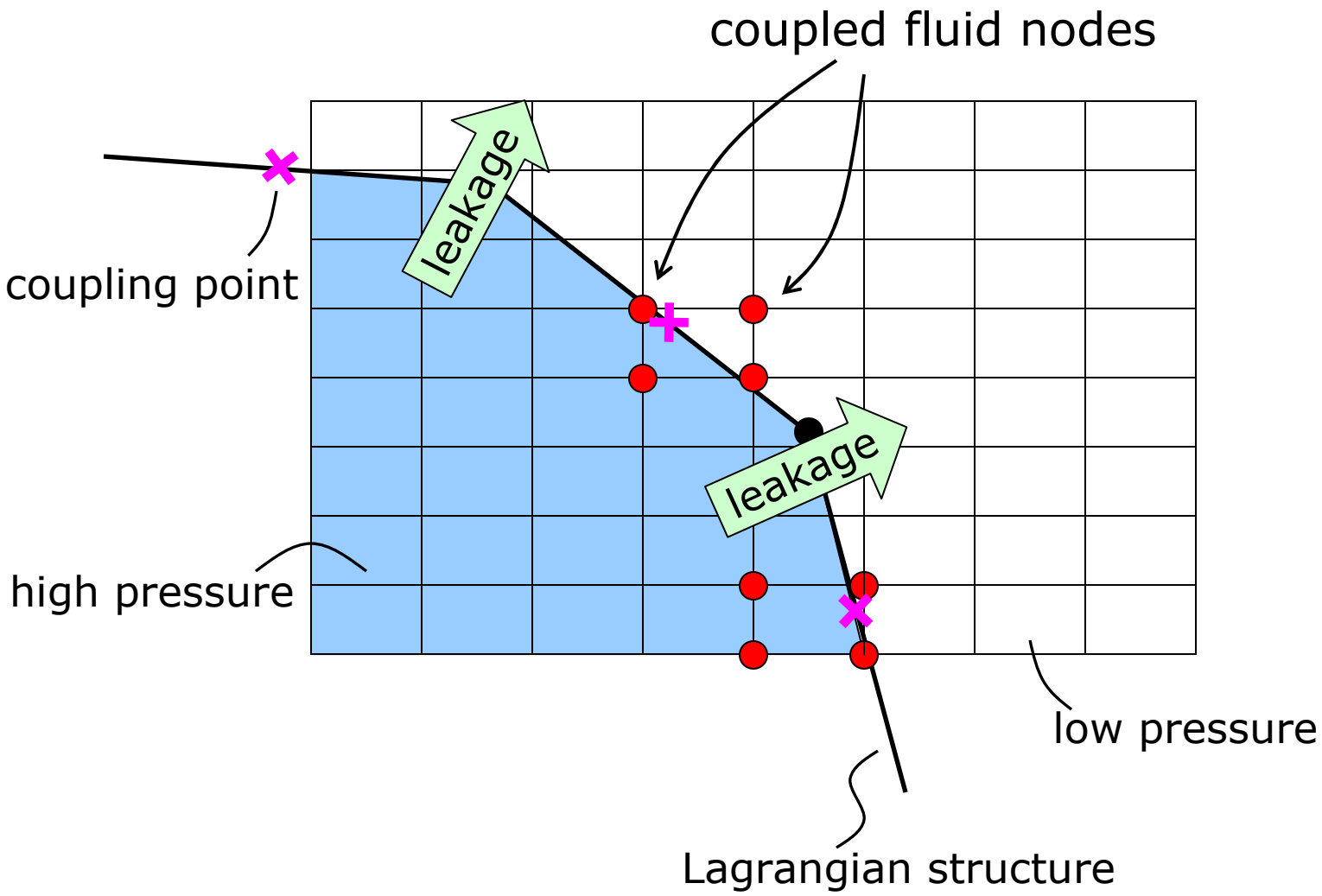
The density of the coupling grid can easily be modified in the input deck, solving the problem

2. Leakage due to numerical errors in the interface reconstruction

Activate the leakage control option (**`ILEAK=1 or 2`**).

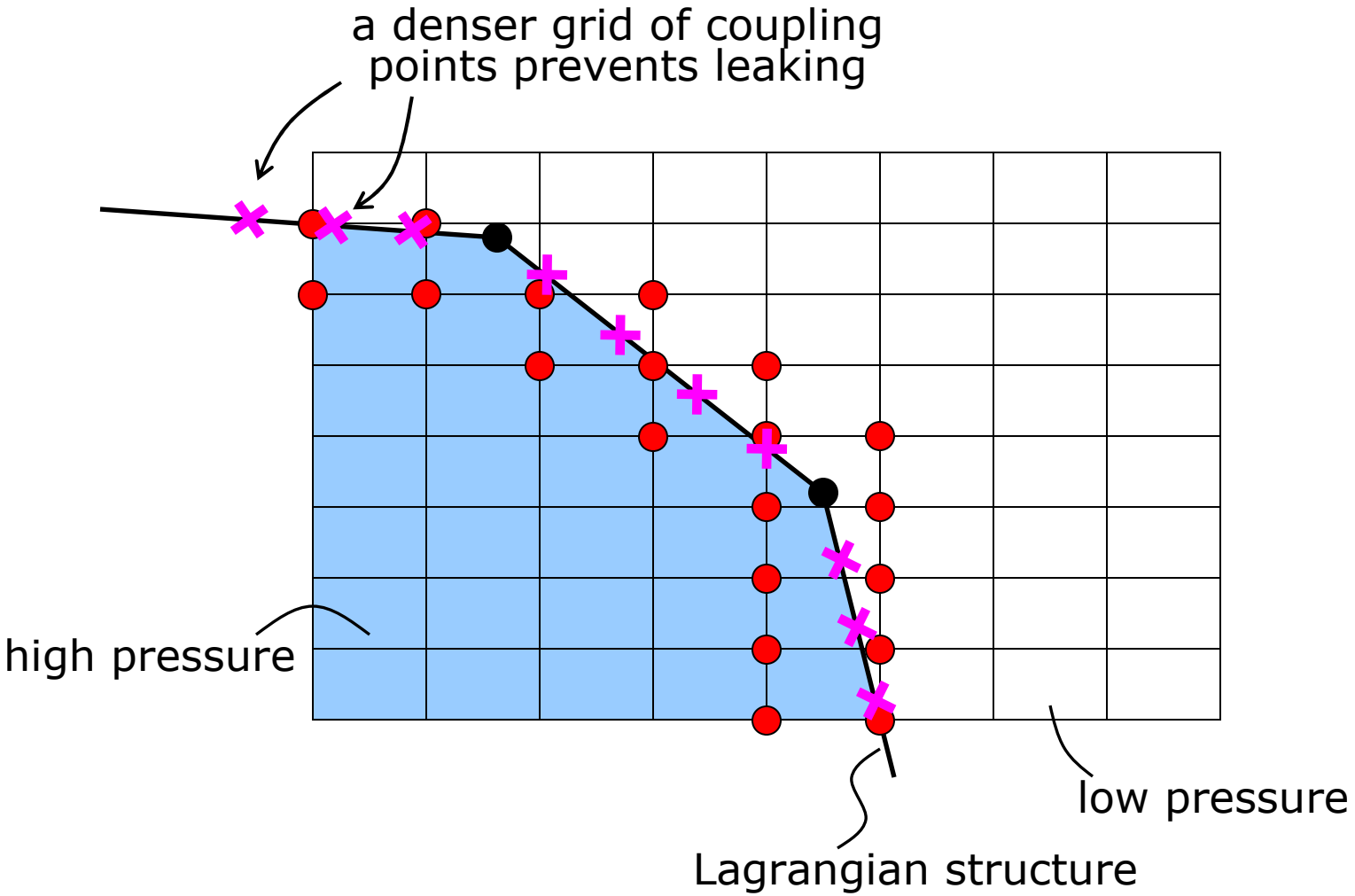
Penalty based method

Leakage, too coarse coupling grid



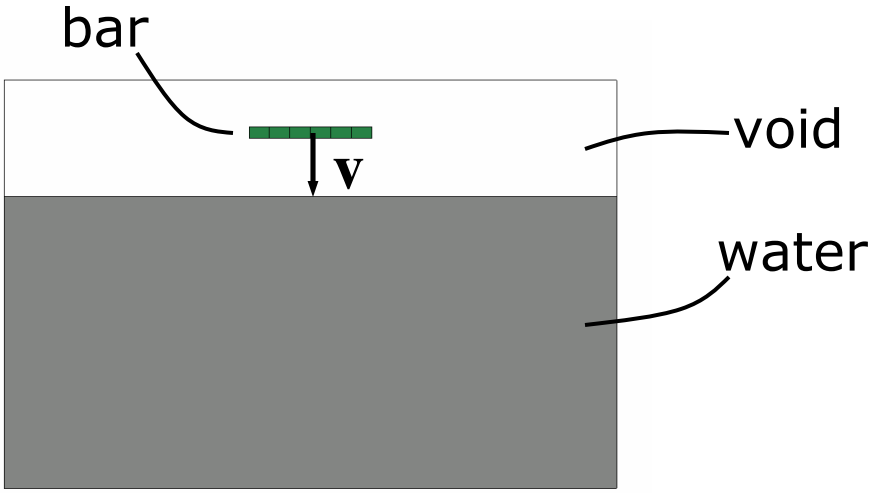
Penalty based method

Leakage, too coarse coupling grid

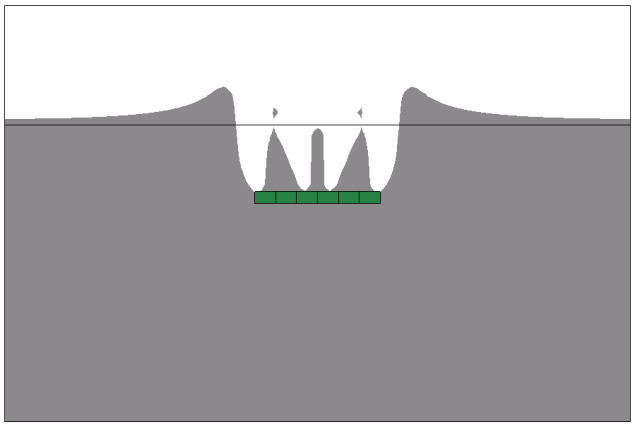


*CONSTRAINED_LAGRANGE_IN_SOLID

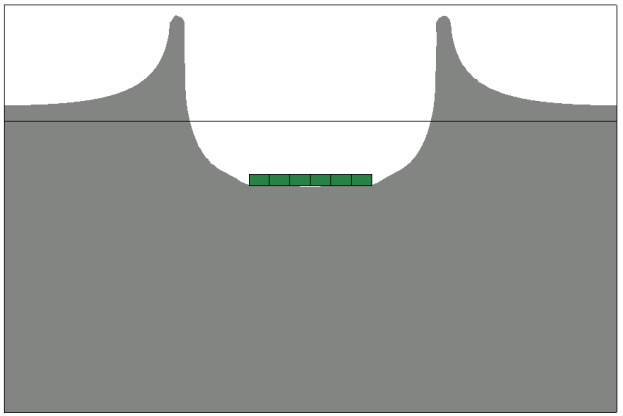
SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
START	END	PFAC	FRIC	FRCMIN	NORM	NORMTP	CDAMP
THERM	HMIN	HMAX	ILEAK	PLEAK			



NQUAD=1



NQUAD=3



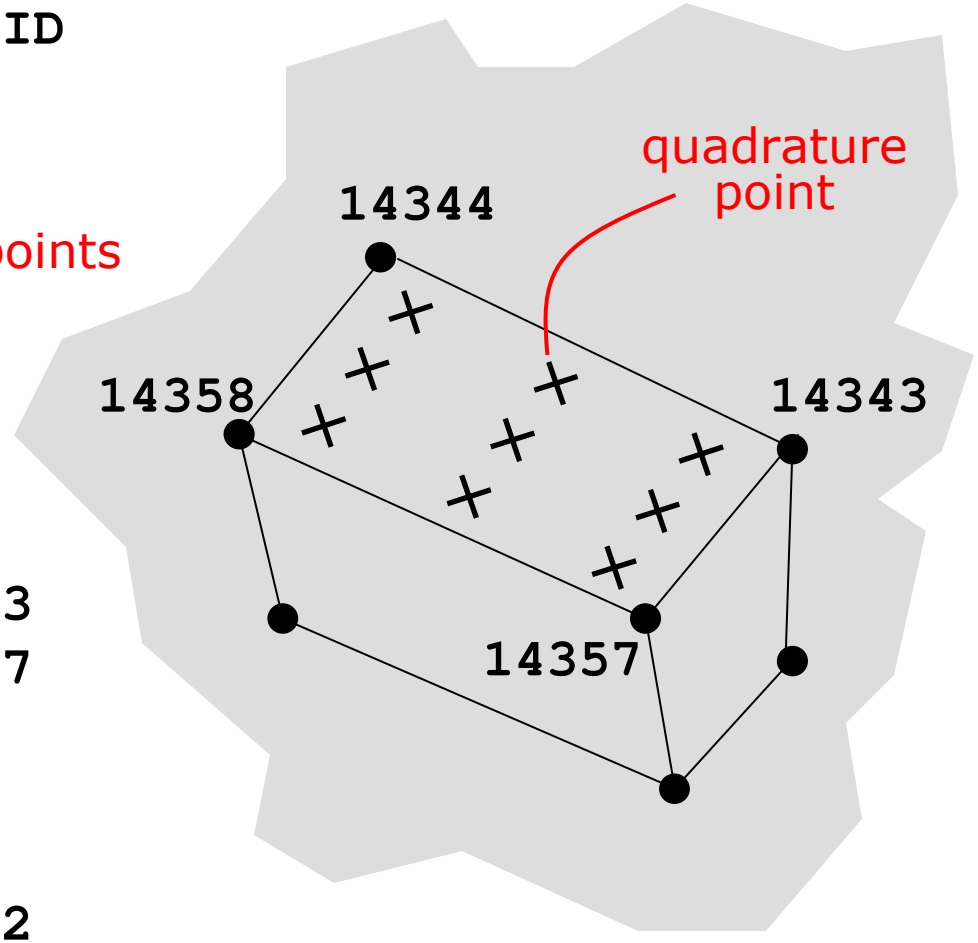
Penalty based method

Leakage, too coarse coupling grid

node coupling

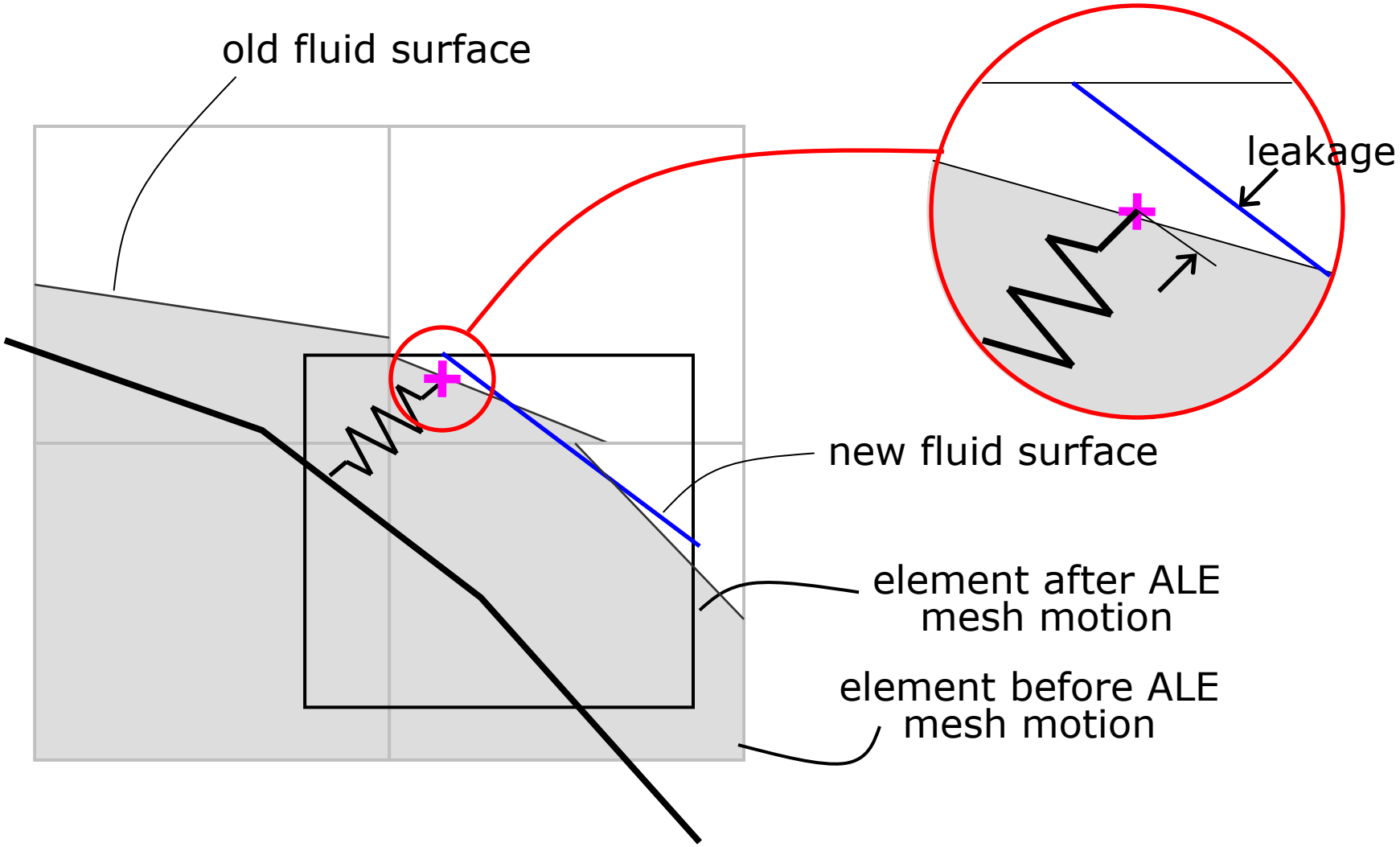
```
*CONSTRAINED_LAGRANGE_IN_SOLID
$1,1,2,0,0,4,2
1,1,2,0,3,4,2

*SET_PART_LIST
1
1,2
*SET_SEGMENT
1
14344,   14358,   14357,   14343
14358,   14372,   14371,   14357
   .
   .
   .
14363,   14377,   14376,   14362
```

3x3 quadrature points



Penalty based method

Leakage, advection errors

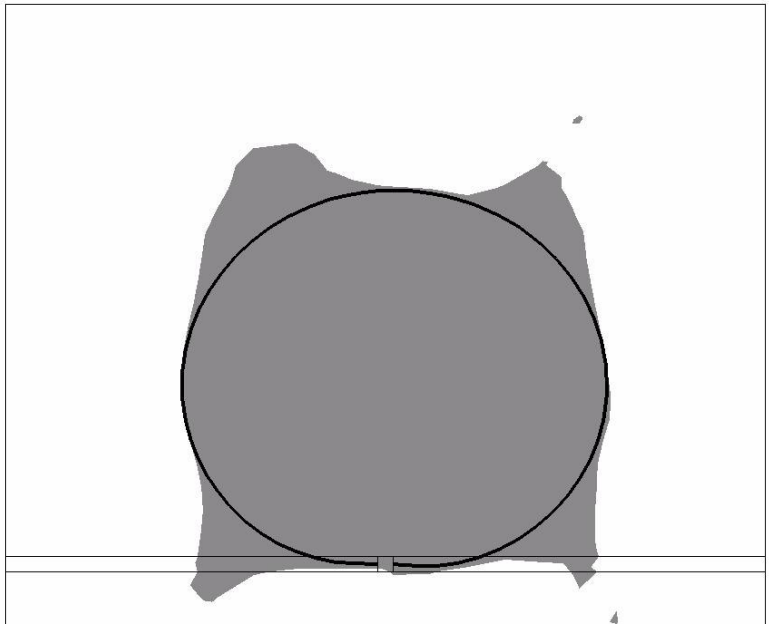


***CONSTRAINED_LAGRANGE_IN_SOLID**

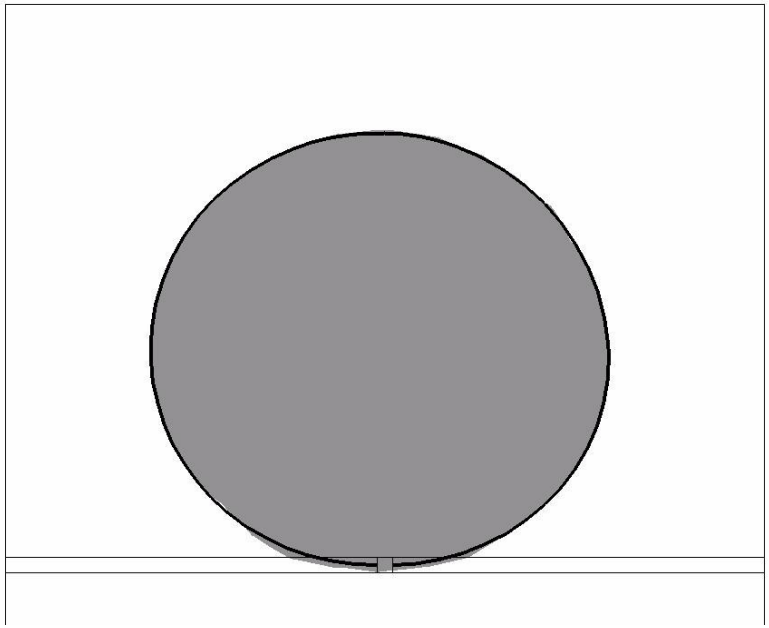
SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
START	END	PFAC	FRIC	FRCMIN	NORM	NORMTP	CDAMP
THERM	HMIN	HMAX	ILEAK	PLEAK			

There is a leakage control algorithm, implemented to deal with this problem. It prevents leaking, but introduces energy to the system.

ILEAK=0



ILEAK=1



Penalty based method

Leakage, energy compensation

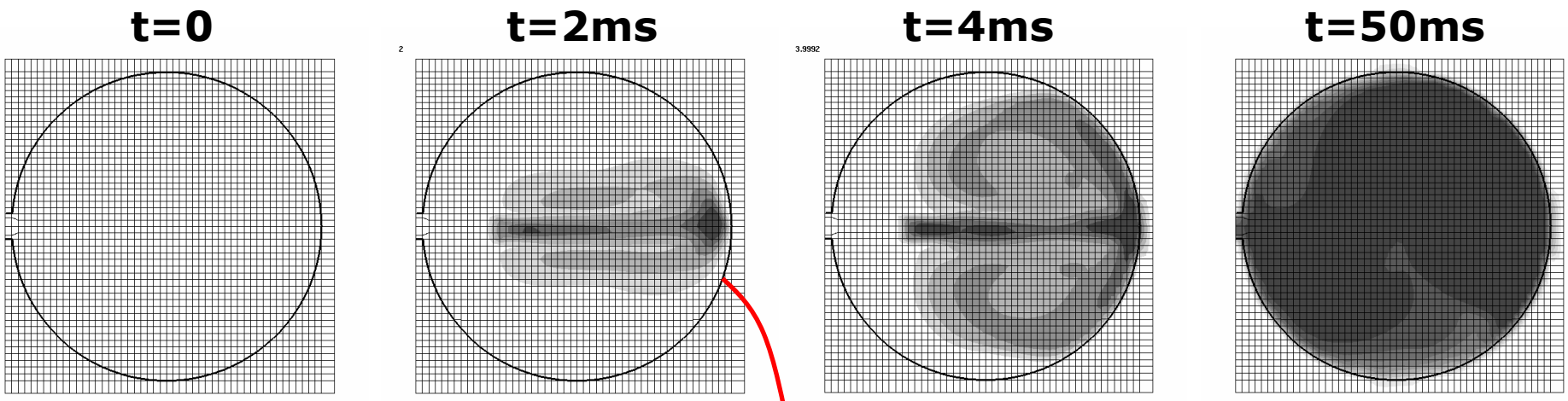
Energy is produced by the leakage control. This energy is eventually transformed into heat and the pressure might rise in a completely non-physical manner.

There is an option that compensates for the added energy by removing kinetic energy from the system. The removal of kinetic energy aims at reducing further leakage. Hence, less leakage control is required. The option is activated by setting **`ILEAK=2`**.

The energy compensation is essentially a set of dampers that work in parallel with the coupling springs. The damping coefficients are dynamically adapted to compensate for non-physically added spring energy.

Leakage
energy compensation

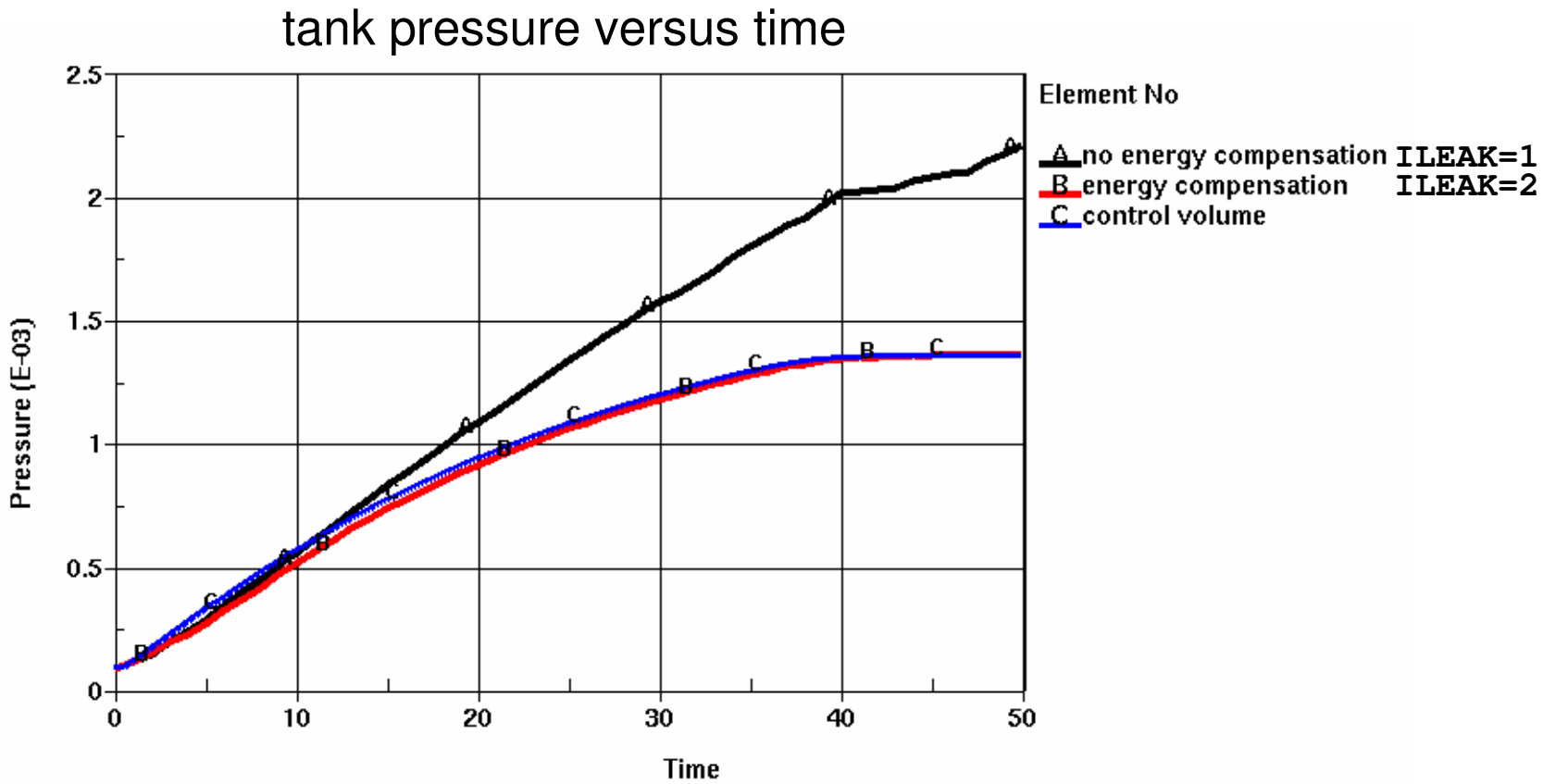
A small two-dimensional tank test. The figures show the density of the inflator gas at different instants in time.



Penalty based method

Leakage, energy compensation

A small two-dimensional tank test. Without energy compensation in the leakage control, the energy grows and the tank pressure ends up far too high.



Penalty based method
User defined penalty stiffness

By default, the coupling interface stiffness is defined automatically by the code. The program looks at the current time step size and tries to estimate a stiffness that is as high as possible, without causing numerical instabilities.

The default method is good enough for many applications. However, there are situations where it leads to instabilities, or to troublesome noise in the contact pressure.

For these situations there is an alternative way of defining the penalty stiffness. The user can provide a load curve that explicitly defines the pressure-penetration relationship. This requires that one, ahead of time, has a rough idea of the expected contact pressure.

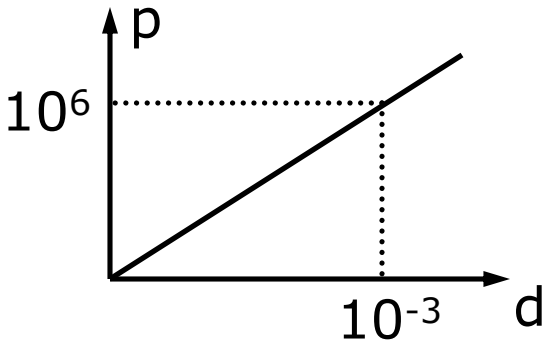
With this method the current time step is automatically reduced, if necessary for stability reasons.

*CONSTRAINED_LAGRANGE_IN_SOLID

```
SLAVE   MASTER SSTYP  MSTYP  NQUAD  CTYPE  DIREC  MCOUP
START   END    PFAC   FRIC   FRCMIN NORM   NORMTP CDAMP
THERM   HMIN   HMAX   ILEAK  PLEAK
```

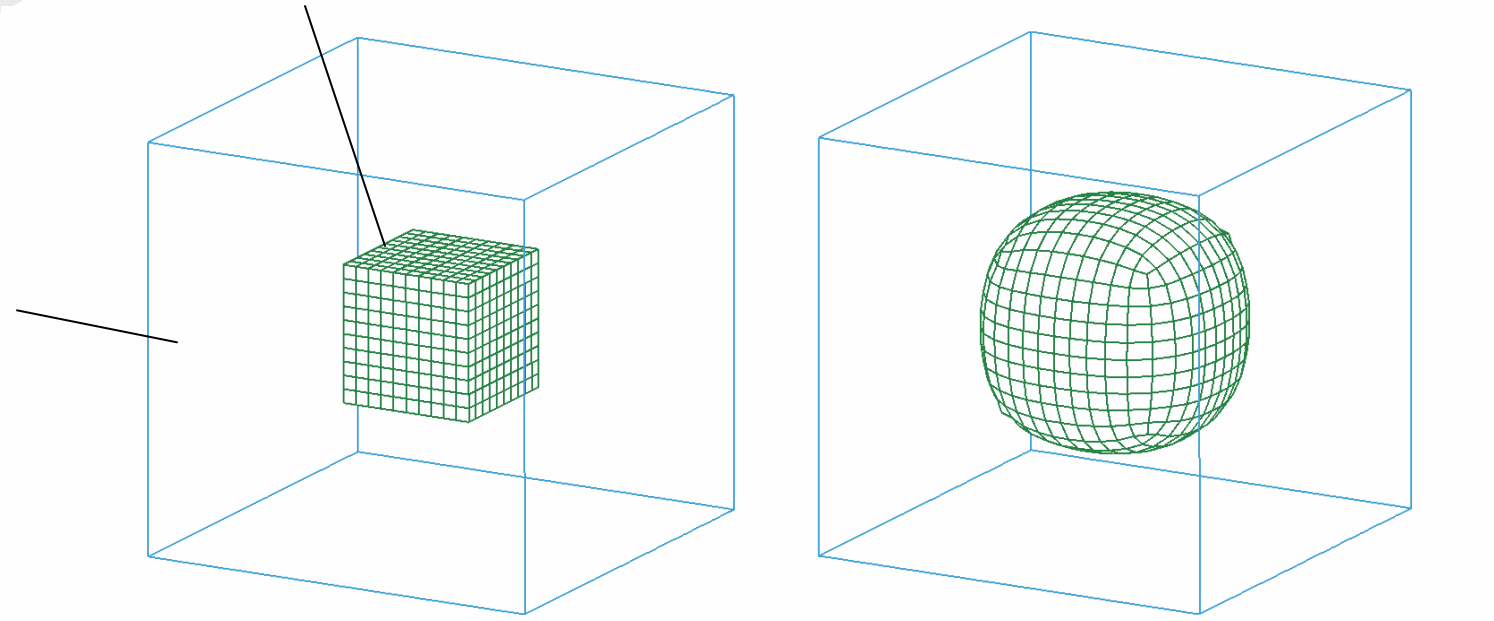
```
*CONSTRAINED_LAGRANGE_IN_SOLID
2, 1, 1, 1, 2, 6, 1
0, 0,-999

*DEFINE_CURVE
999
0.0, 0.0
1.0e-3, 1.0e+6
```

load curve for pressure-penetration relationship



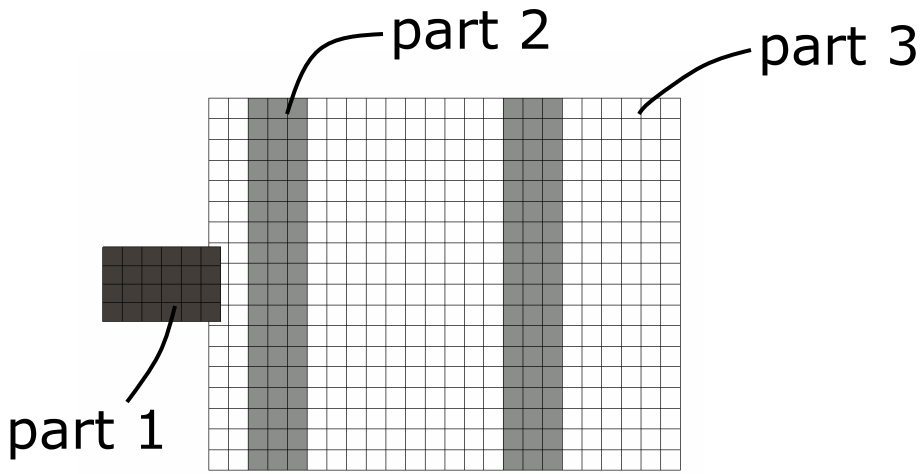
The box (part 2) is filled with a pressurized gas

Eulerian mesh (part 1)



Penalty based method

Eroding coupling



attention!

With Type 5 coupling, the slave side must be a solid part or a set of solid parts

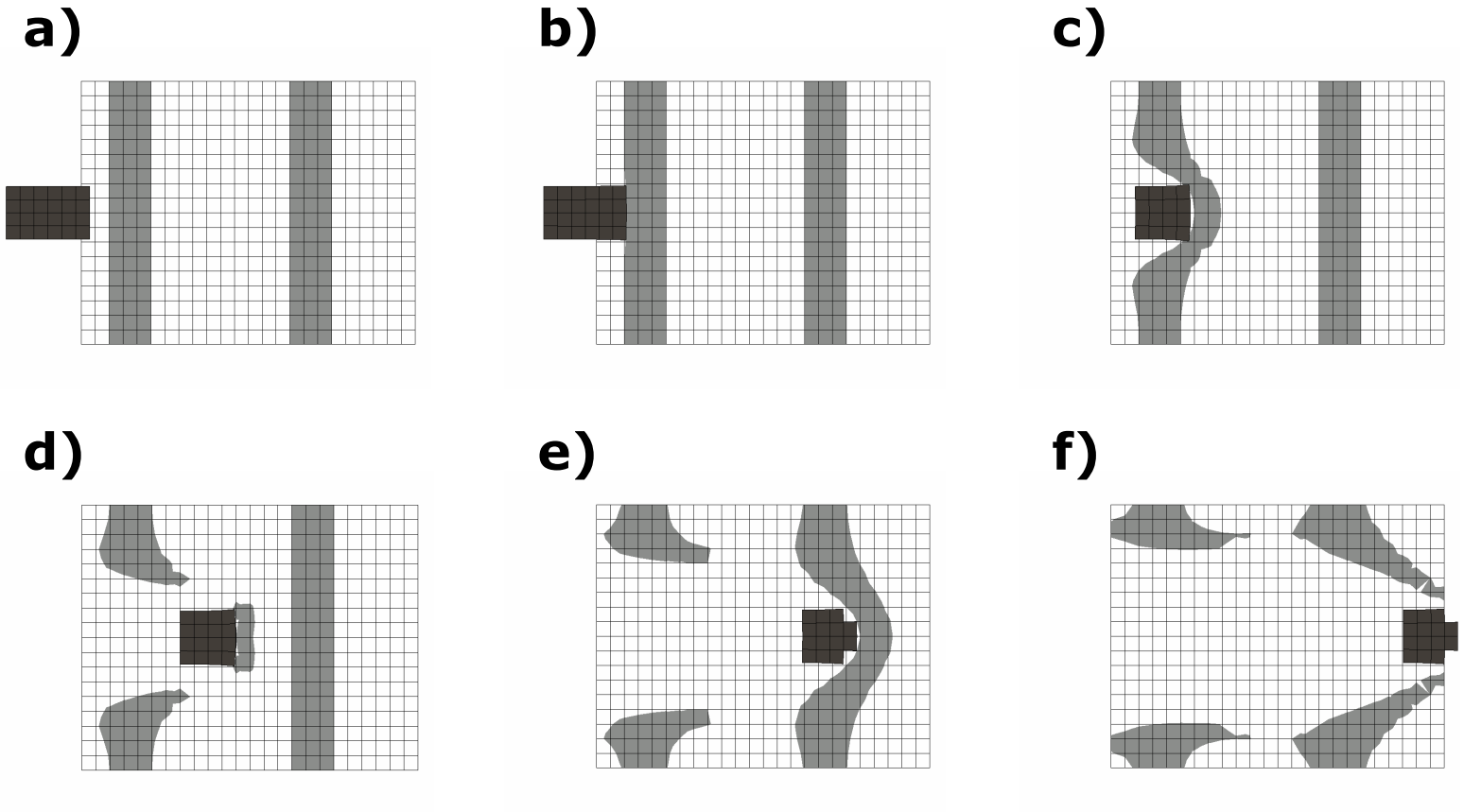
Type 4 coupling should be used when dealing with eroding shell elements

eroding coupling

```
*CONSTRAINED_LAGRANGE_IN_SOLID
1,1,0,1,1,5,2

*SET_PART_LIST
1
2,3
```

Penalty based method

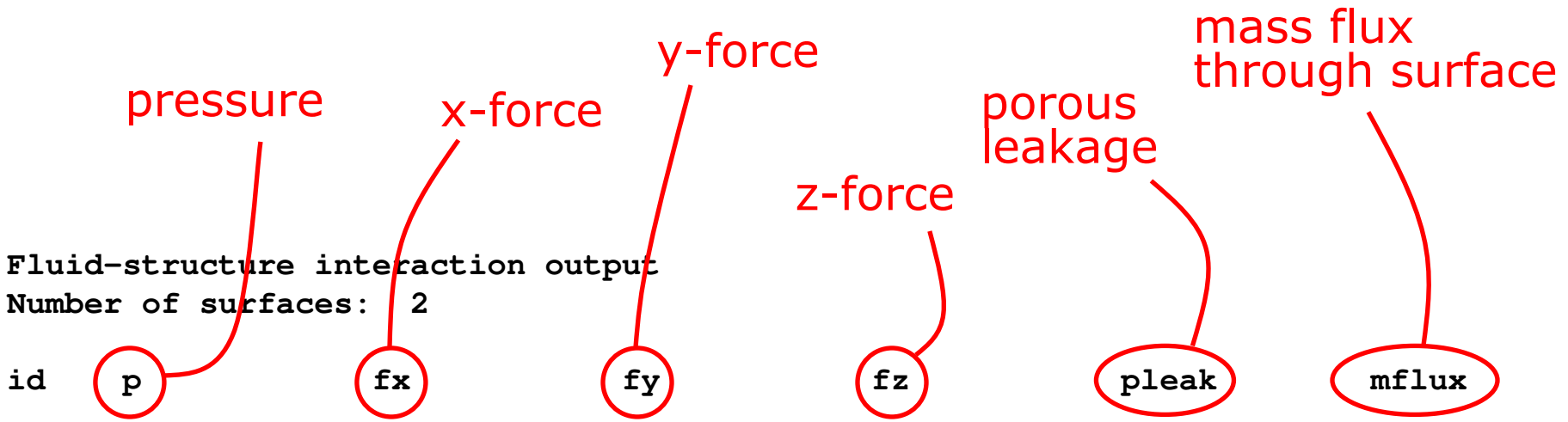
Eroding coupling



*DATABASE_FSI

DT

DBFSI_ID SID SIDTYPE




```
Fluid-structure interaction output
Number of surfaces:  2

id     p          fx          fy          fz          pleak       mflux

   time=  0.00000E+00
 1    0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
 2    0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00

   time=  0.10200E-03
 1    0.1632E+05  0.4091E+01 -0.3215E+01  0.2546E+01  0.0000E+00 -0.1284E-04
 2    0.1947E+05 -0.4878E+01 -0.4174E+01  0.2548E+01  0.0000E+00 -0.7193E-05
```

Some useful materials and EOS

- ***MAT_NULL**
- ***MAT_GAS_MIXTURE**
- ***MAT_VACUUM**
- ***MAT_HIGH_EXPLOSIVE_BURN**
- ***EOS_JWL**
- ***EOS_GRUNEISEN**

Some materials and EOS

*MAT_NULL

A material without shear strength, frequently used to model gases and liquids.

It should always be combined with an EOS that defines the pressure as a function of compression and internal energy.

***MAT_NULL**

MID	DENS	PC	MU

MID	Material ID
DENS	Density
PC	Pressure cut-off
MU	Viscosity

*MAT_GAS_MIXTURE

The model is designed for the treatment of hybrid inflators in coupled ALE-airbag models. ***MAT_GAS_MIXTURE** handles the mixing of up to eight different ideal gases.

Special action is taken to conserve the total energy in the Eulerian advection step. Dissipated kinetic energy is automatically transformed into heat.

up to eight different gases

$$p = \sum_{i=1}^{N} \rho_i (C_{Pi} - C_{Vi}) T$$

density and heat capacities of the different gas species

An additional card, ***INITIAL_GAS_MIXTURE** is required to initialize the state of the mixture.

Some materials and EOS

*MAT_GAS_MIXTURE

*MAT_GAS_MIXTURE

MID							
CV1	CV2	CV3	CV4	CV5	CV6	CV7	CV8
CP1	CP2	CP3	CP4	CP5	CVP	CP7	CP8

MID	Material ID
CV1–CV8	Heat capacities at constant volume
CP1–CP8	Heat capacities at constant pressure

*INITIAL_GAS_MIXTURE

SID	STYPE	MMGID	T0				
RHO1	RHO2	RHO3	RHO4	RHO5	RHO6	RHO7	RHO8

SID	Set ID
STYPE	Set type (0=part set, 1=part)
MMGID	Multi-material group of the mixture that is to be initialized
T0	Initial temperature of mixture
RHO1–RHO8	Initial density of the different gas species

Some materials and EOS

*MAT_VACUUM

Vacuum can be specified to fill up empty regions in a multi-material model.

***MAT_VACUUM**

MID	**RHO**

MID	Material ID
RHO	Density (non-zero value to prevent division by zero)

*MAT_HIGH_EXPLOSIVE_BURN

The model is designed for the modeling of the detonation of a high explosive. It needs to be combined with an EOS, normally ***EOS_JWL** or ***EOS_JWLB**.

The detonation can either be initiated at a specified time and location with ***INITIAL_DETONATION**, or by a volumetric compression (see parameter **BETA**).

Some materials and EOS

*MAT_HIGH_EXPLOSIVE_BURN

*MAT_HIGH_EXPLOSIVE_BURN

MID	RO	D	PCJ	BETA	K	G	SIGY

MID	Material ID
RO	Density
D	Detonation velocity
PCJ	Chapman-Jouget pressure
BETA	Beta burn flag Eq.0: beta + programmed burn Eq.1: beta burn only Eq.2: programmed burn only
K	Bulk modulus
G	Shear modulus
SIGY	Yield stress

Some materials and EOS

*EOS_JWL

***EOS_JWL**

EOSID	A	B	R1	R2	OMEG	E0	V0

EOSID	EOS ID
A	see equation below
B	
R1	
R2	
OMEG	
E0	Internal energy (per unit volume)
V0	Initial relative volume

exponentially decaying pressure in expansion

detonation products behave like an ideal gas at large relative volumes

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

Some materials and EOS

*EOS_GRUNEISEN

The propagation velocity of a shock front depends on the particle velocity. This effect is taken into account by ***EOS_GRUNEISEN**.

***EOS_GRUNEISEN**

EOSID	C	S1	S2	S3	G	A	E0
V0							

EOSID	EOS ID
C	Acoustic speed of sound
S1–S3	Parameters defining shock velocity as a function of particle velocity
G	Gruneisen gamma
A	
E0	Initial internal energy
V0	Initial relative volume

$$p = \frac{\rho_0 C^2 \mu[1 + (1 - \frac{\gamma}{2})\mu - \frac{a}{2}\gamma^2]}{\left[1 - (S_1 - 1)\mu - S_2 \frac{\mu^2}{\mu + 1} - S_3 \frac{\mu^3}{(\mu + 1)^2}\right]^2} + (\gamma + a\mu) E$$

Some materials and EOS

*EOS_GRUNEISEN

Some data for water, steel and cupper from http://de.wikipedia.org (higher order terms omitted)

$$p = \frac{\rho_0 C^2 \mu [1 + (1 - \gamma/2)\mu]}{[1 - (S_1 - 1)\mu]^2} + \gamma E$$

WATER

ρ_0 =1000kg/m³, C=1489m/s, S_1=1.79, γ=1.65

STEEL

ρ_0 =7850kg/m³, C=4500m/s, S_1=1.49, γ =2.17

CUPPER

ρ_0 =8939kg/m³, C=3940m/s, S_1=1.48, γ =1.96