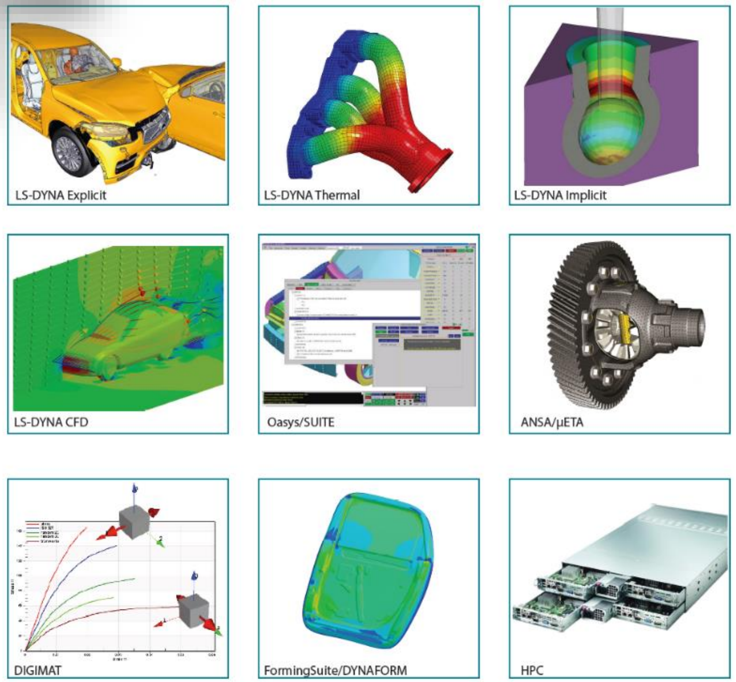
Modelling large deformation and failure using LS-DYNA

Mikael Schill, DYNAmore Nordic

Contact – DYNAmore Nordic

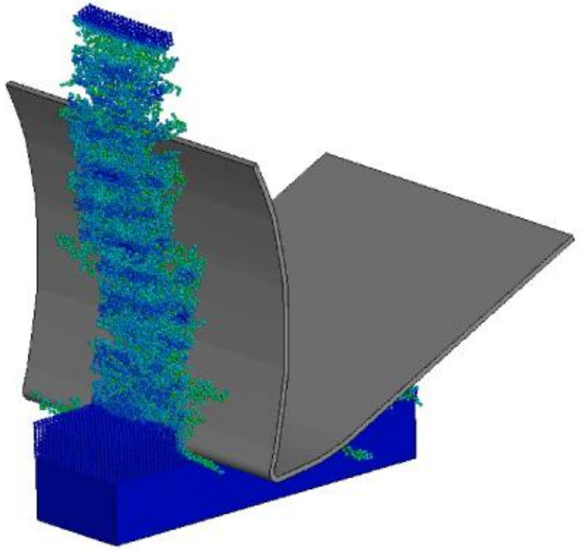
- Support
  - Sales: sales@dynamore.se
  - E-mail: support@dynamore.se
    - Target to answer in 4 hours
    - Call: +46 13 236680
- Training & seminars including on-line & on-site: www.dynamore.se
- Secure file server: files.dynamore.se
  - Software and license download, client area with guidelines and more
- www.dynamore.se - information on LS-DYNA, Seminars, Conferences
- www.dynalook.com – Papers from international LS-DYNA conferenses
- www.dynasupport.com – General support for LS-DYNA
- www.dynaexamples.com – LS-DYNA example models from crash to DEM

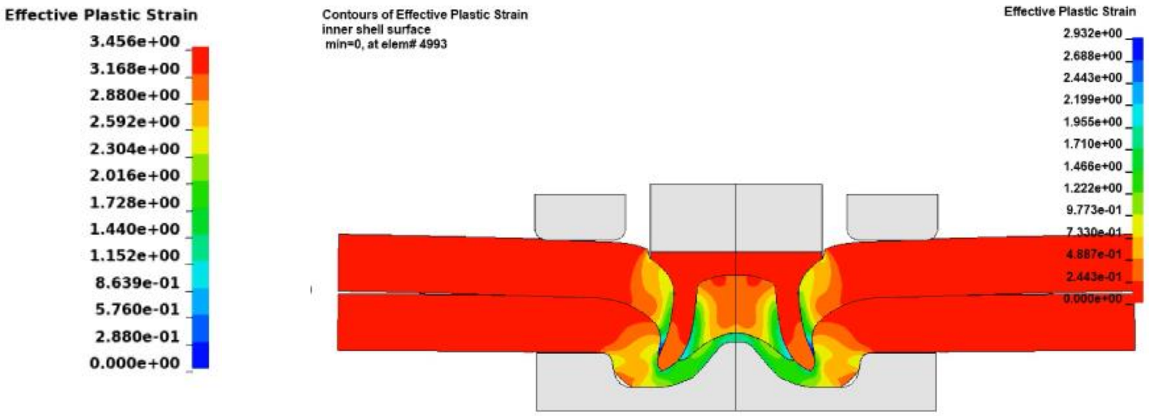


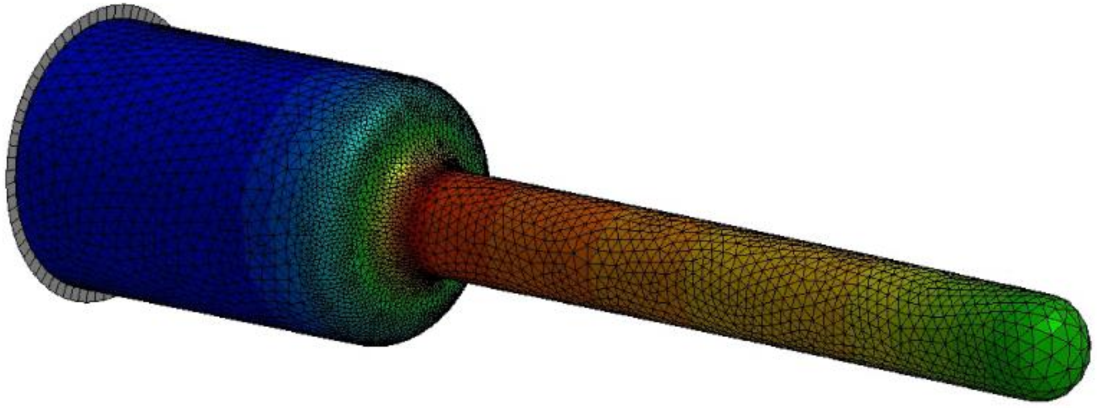
Applications

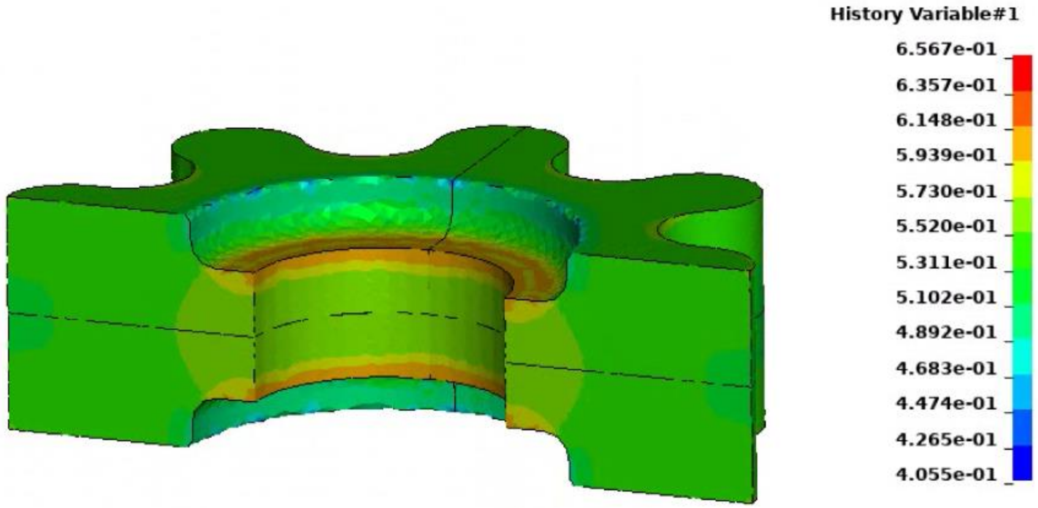
Example applications for simulation of large deformations and failure

- Self Piercing Rivets
- Flow Drill Screws
- Cutting/Milling
- Forging
- Powder Compaction
- Extrusion









Simulation methodology – Large deformations

Overview on available methods in LS-DYNA

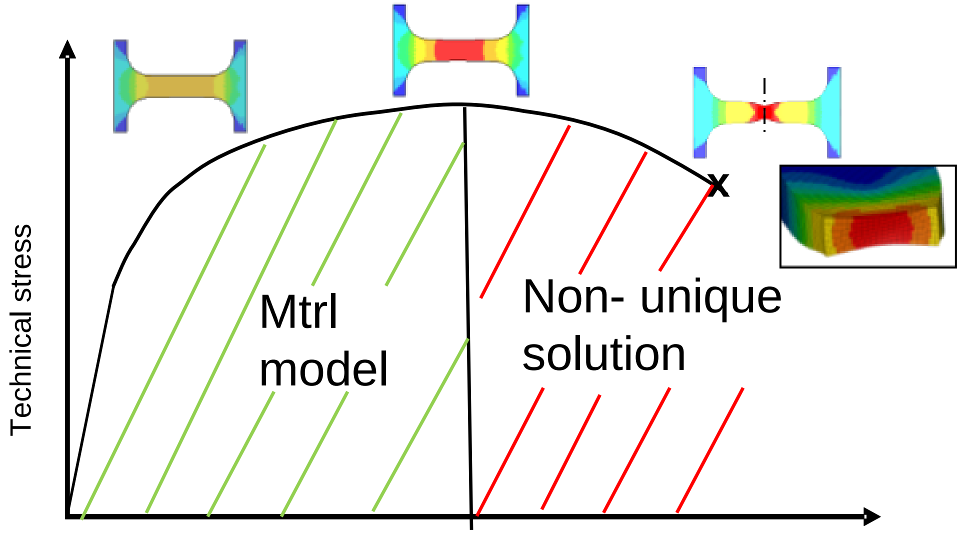
- The applications can show one or several of the following behaviors
  - Large deformations with material movement
  - High contact pressures
  - Material failure
  - Thermo- mechanical process

- There are several methods in LS-DYNA to solve these types of problems
- Time discretization can be either implicit or explicit
- Spatial discretization
  - FEM
  - Remeshing Adaptivity – Both 2D and 3D available
  - ALE/S-ALE – (Structured) Arbitrary Lagrangian Eulerian method.
  - Meshless methods
    - SPH – Smooth Particle Hydrodynamics
    - EFG – Element Free Galerkin
    - SPG – Smooth Particle Galerkin

Failure modeling

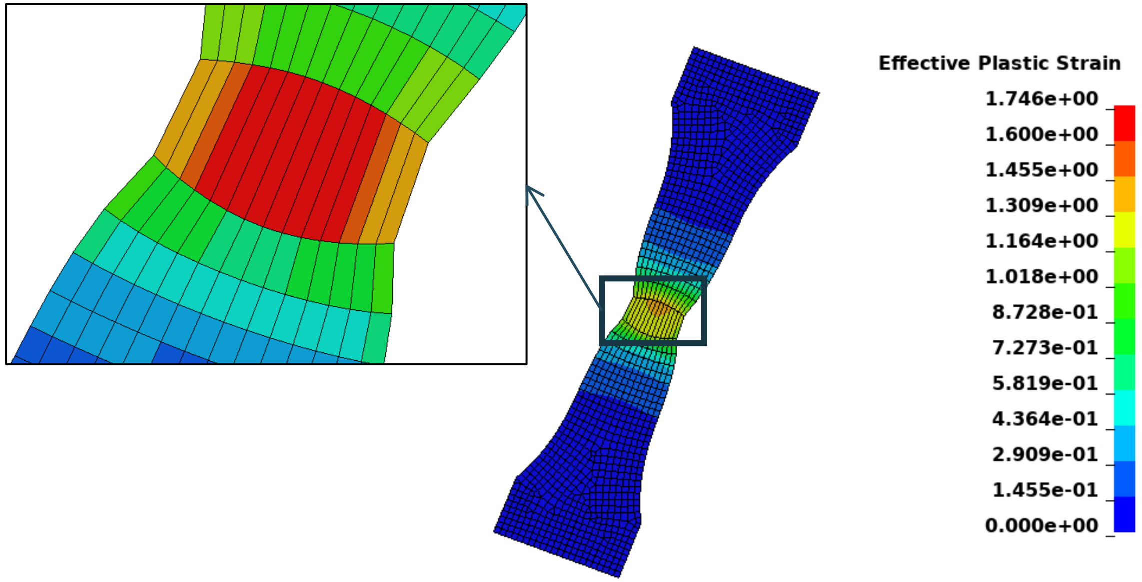
Background

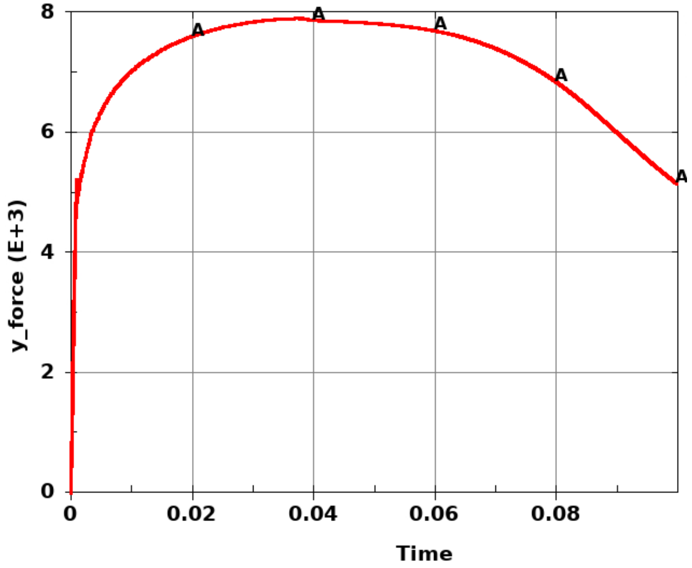
- Failure can be modelled using several approaches in LS-DYNA
- As long as the solution is stable (no material instabilities), the solution is unique and the behavior is determined by the material model
- Beyond the point of instability (e.g. diffuse necking) the solution becomes non- unique.
  - Element type and size
  - Material model
  - Hardening
  - …



Failure modeling - Constitutive model

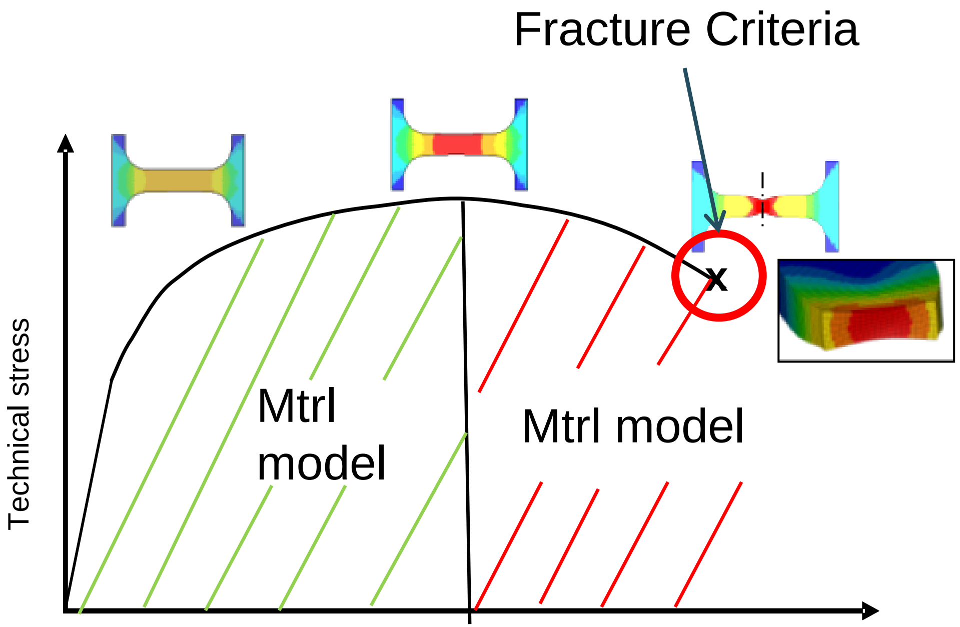
- The constitutive model alone will provide a softening response beyond the instability point.
- For very large deformations e.g. cutting or milling this might be enough since the material data is quite uncertain for high strains, strain rates and temperatures.
- The material “splitting” is then handled by the simulation method (3D adaptivity, ALE, etc)





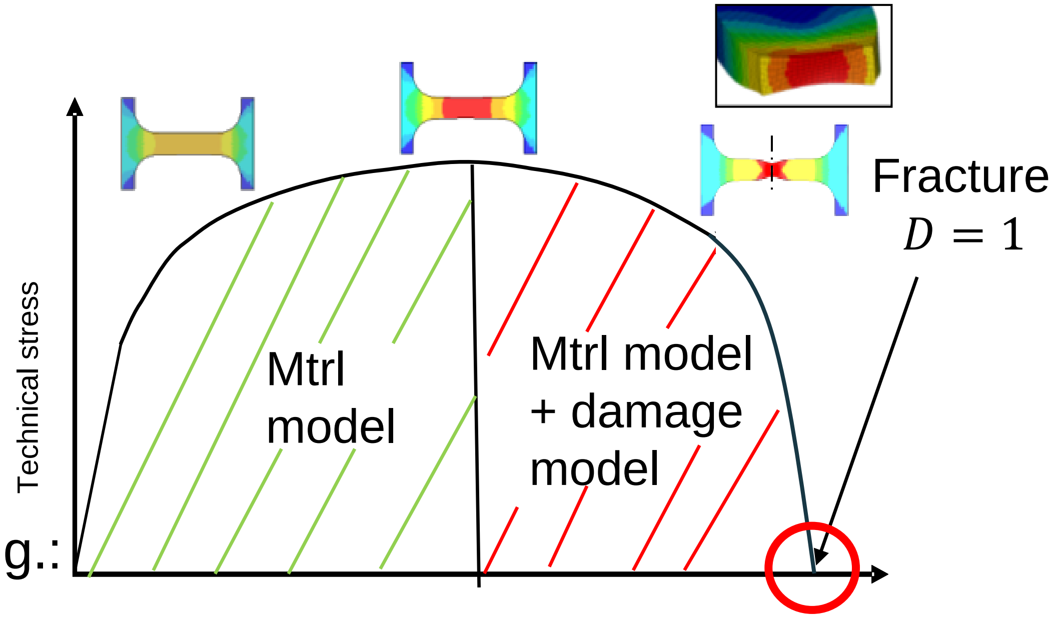
Failure modeling - Criteria

- Fracture Criteria is predicting when a material fails
- It can be based on e.g. effective plastic strain, pressure, maximum principal strain or a failure curve.
- Many material models inside LS-DYNA contains erosion criteria e.g.
  - *MAT_24 : Piecewise Linear Plasticity
  - *MAT_15 : Johnson Cook
  - *MAT_224: Tabulated Johnson Cook
- The user can add erosion criteria to any material model in LS-DYNA using *MAT_ADD_EROSION
- When the failure criteria is reached, the element is deleted from the simulation.



Failure modeling - Damage

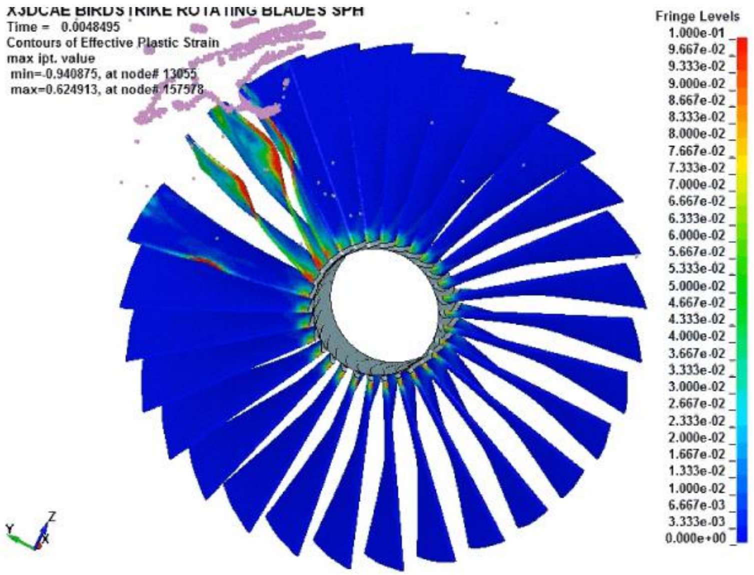
- Damage modeling means predicting how a material fails
- The most common way of doing this is to add damage model to the constitutive model
- The damage model is a scalar (D) that evolves with e.g. plastic deformation and couples to the stress
- As the damage is coupled the material softens
- When D=1, the material carries no load and can be deleted
- Many material models in LS-DYNA accounts for damage, e.g.:
  - MAT_104: Damage
  - MAT_187 : Samp-1
- Damage can be added to “any” material model in LS-DYNA using e.g.:
  - *MAT_ADD_DAMAGE_DIEM
  - *MAT_ADD_DAMAGE_GISSMO
  - *MAT_ADD_GENERALIZED_DAMAGE

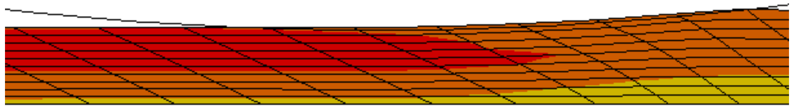


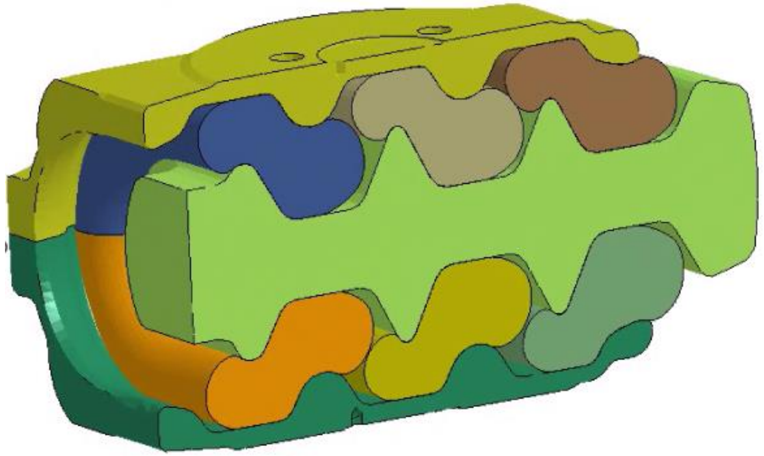
Lagrangian FEM

Modeling Large deformations using FEM

- Handling large deformations basically means dealing with
  - Large transformations and rotations
  - Large material deformations
- LS-DYNA handles large translations and rotations by using objective stress update
  - The stress measure is not dependent on the frame of reference and large rigid body motions does not introduce strains
- In lagrangian FEM the element follows the material deformation which leads to element distorsion.
  - Inverted elements (negative volume)
  - Timestep drops
  - Loss of accuracy
- Lagrangian FEM can handle a lot of abuse, but at some point it is necessary to switch to other formulations

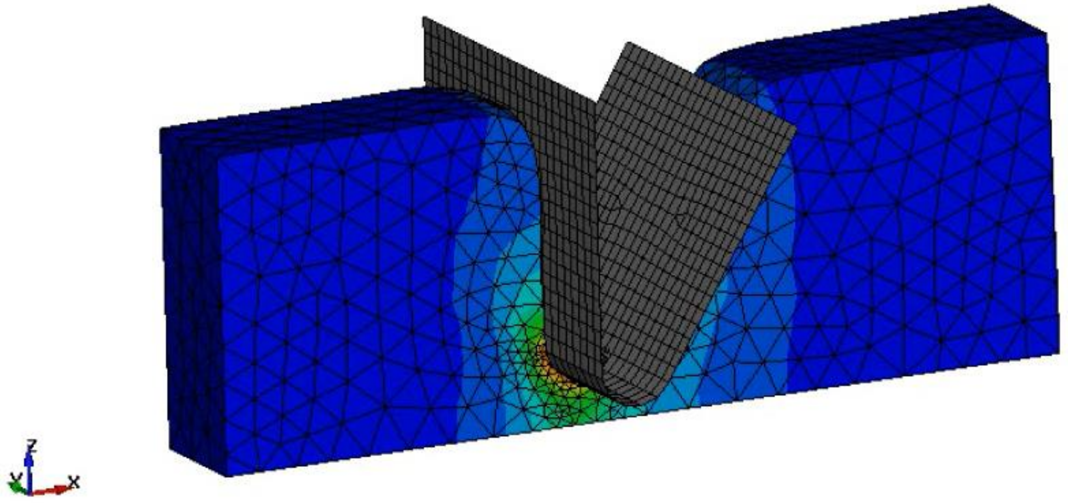






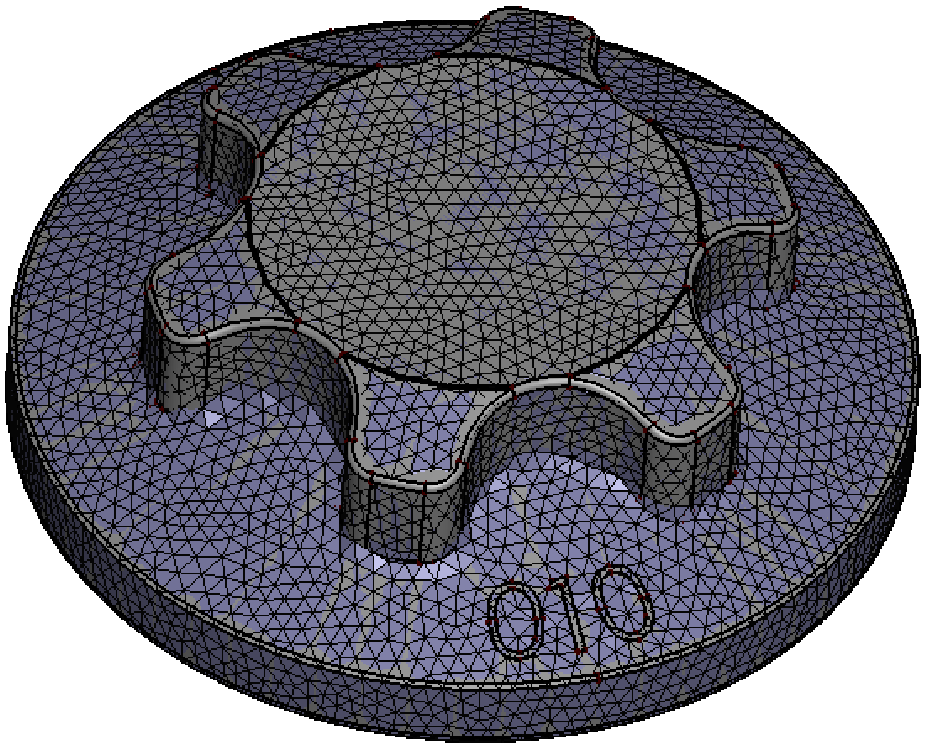
Adaptive remeshing

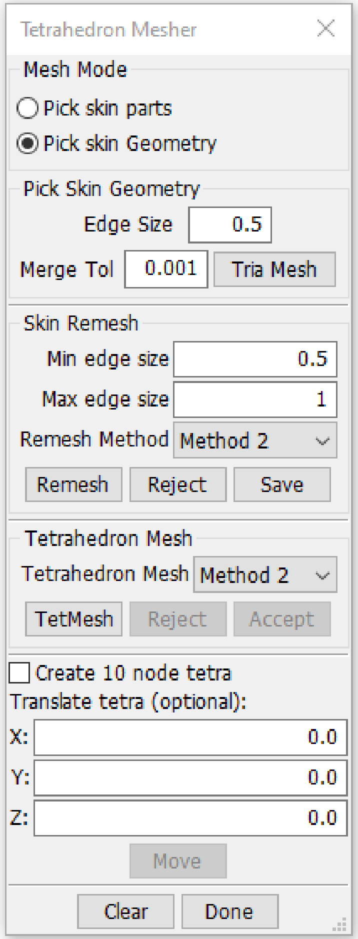
- By using remeshing, it is possible to use lagrangian Finite Elements for simulating arbitrary deformations.
- Supports Thermo- mechanical simulations
- Can be combined with EFG formulation
- By using “Look Ahead Adaptivity”, thus refinement based on contact curvature, it is possible to limit the number of elements.
- No erosion criteria possible. The mesh will conform to the cutting surface
- Drawbacks are:
  - Adaptive part can not separate in 3D
  - Remeshing is time- based – Risk for inverted elements
  - Restart and mapping of results leads to loss of data.
  - Constraints are lost at adaptive refinement

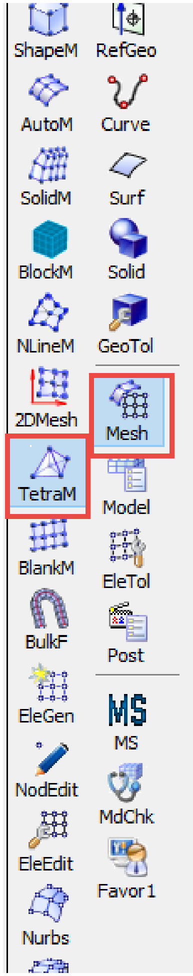


Preprocessing

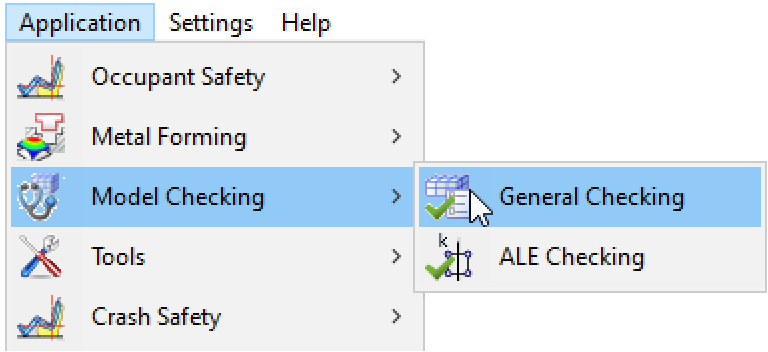
Terahedron mesh in LS-PREPOST







- Tetrahedron mesh needs a triangular skin.
- The skin can be from a meshed part or geometry
- **Remesh**: Remeshes the skin based on the triangular skin. Thus, the geometry information is lost.
- The re-meshing is based on minimum and maximum edge size
- When the surface mesh is OK, do the Tetrahedron mesh
- Element quality can be checked in model checking



LS-DYNA remeshing keywords

```
*CONTROL_ADAPTIVE
$# adpfreq    adptol    adptyp    maxlvl    tbirth    tdeath     lcadp    ioflag
     0.001 1.000E20         7         3       0.01.00000E20         0         0
$# adpsize    adpass    ireflg    adpene     adpth    memory    orient     maxel
       0.0         0         0       1.0       0.0         0         0         0
…
```

- **adpfreq**: Adaptive remesh frequency. Can be controlled by loadcurve **lcapd**
- **adptyp**: Set to 7 for 2D/3D remeshing
- **adpene:** Remesh based on contact search (look ahead adaptivity) and tool curvature.
- **adptol:** For 2D adaptivity this is the target element size

```
*CONTROL_REMESHING
$#    rmin      rmax   vf_loss     mfrac    dt_min     icurv    iadp10    sefang
      1.00      1.50       1.0       0.0       0.0         4         0       0.0
```

- **rmin**: 3D adaptivity minimum edge length
- **rmax**: 3D adaptivity maximum edge length
- **icurve:** Number of elements along look ahead radius

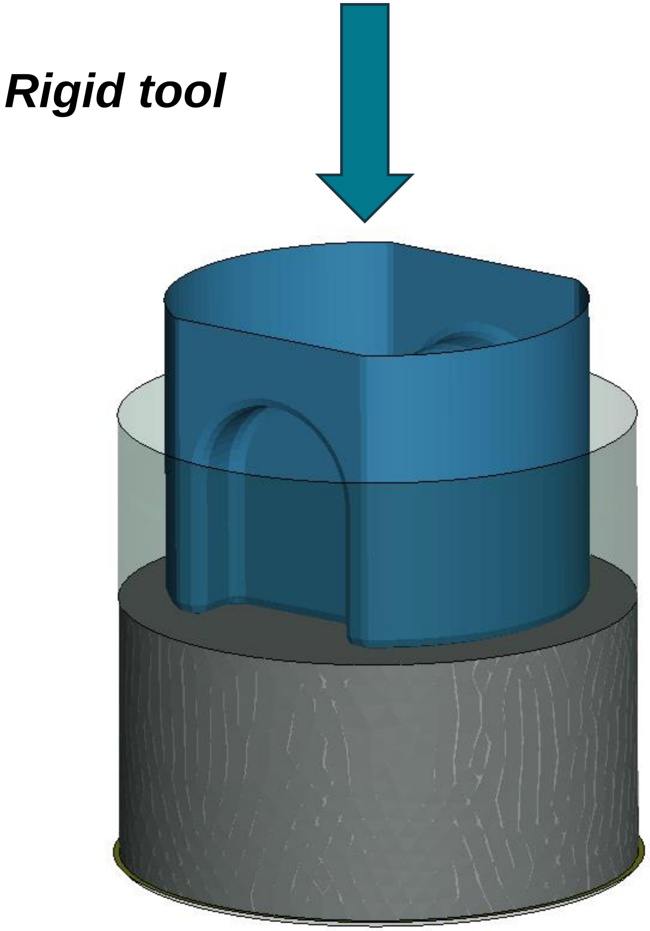
```
*PART
$#                                                                         title
blank
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         1         1         1         0         0         0         2         0
```

- **adpopt**: Set to 2 for remeshing adaptivity (both 2D and 3D)

```
*CONSTRAINED_LOCAL
$#      tc        rc       dir         x         y         z       cid       tol
         1         1         1       0.0      30.0       0.0         1       0.0
```

- `*BOUNDARY_SPC` constraint will not work with adaptivity.
- Use `*CONSTRAINED_LOCAL` instead
- Nodes within a tolerance of a local coordinate plane is constrained.

Forging example



200 adaptive remeshes with look ahead adaptivity

```
*CONTROL_ADAPTIVE
$#  adpfreq    adptol    adptyp    maxlvl    tbirth    tdeath     lcadp    ioflag
        0.1  1.000E20         7         3       0.01.00000E20         0         0
$#  adpsize    adpass    ireflg    adpene     adpth    memory    orient    maxel
        0.0         0         0       4.0       0.0         0         0         0
```

1 point tetrahedron element

```
*SECTION_SOLID
$#    secid    elform       aet
          1        13         0
```

Blank mesh size is less than tool mesh. Rmax/Rmin ≅3

```
*CONTROL_REMESHING
$#     rmin      rmax   vf_loss     mfrac    dt_min     icurv    iadp10    sefang
       1.00      3.00       1.0       0.0       0.0         4         0       0.0
```

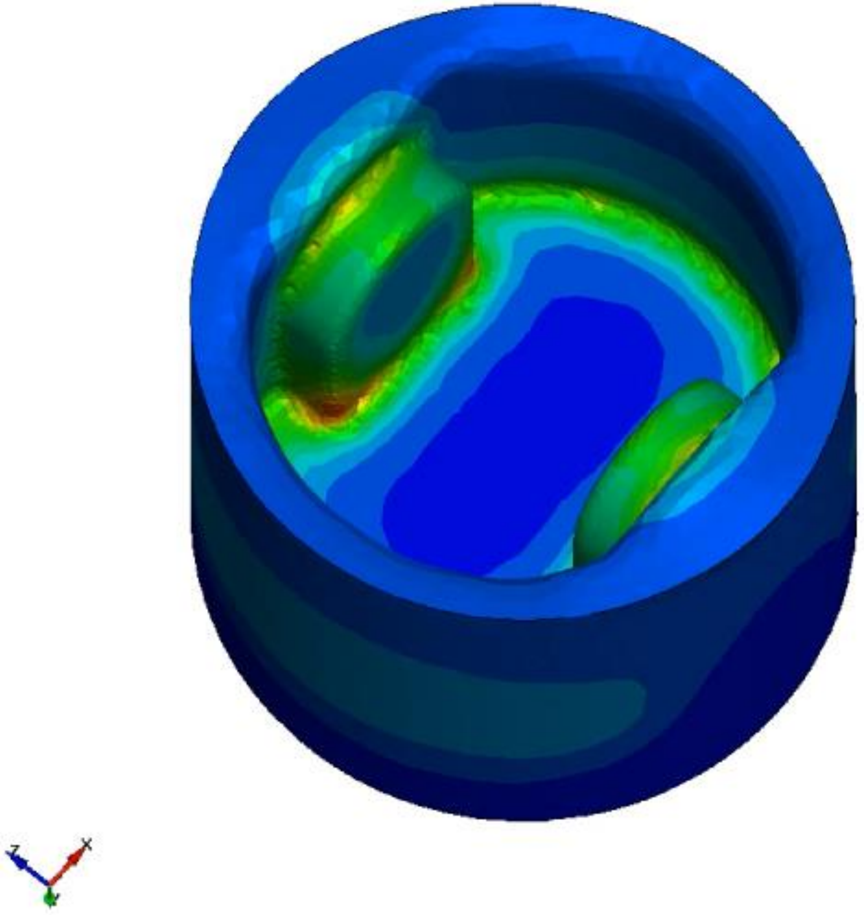
Forming contact allows for "Look ahead" adaptivity

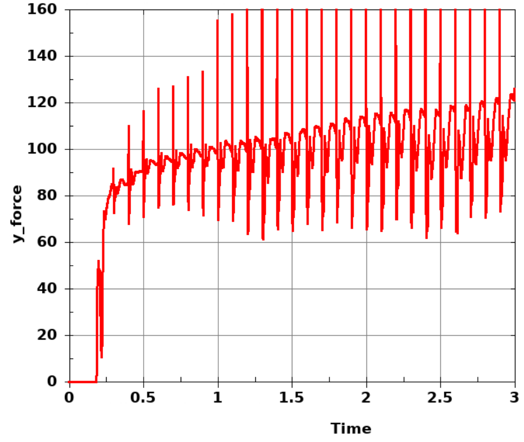
```
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE_ID
$#      cid
          3
$#     ssid      msid     sstyp     mstyp    sboxid    mboxid       spr       mpr
          5         3         3         0         0         0         0         0
```

Forging Example

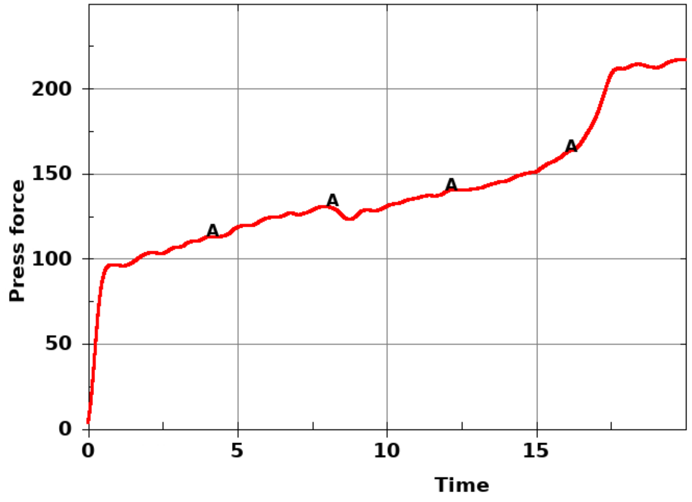
Post processing – LS-PREPOST

"Look ahead" adaptivity





Contact force is noisy from adaptivity



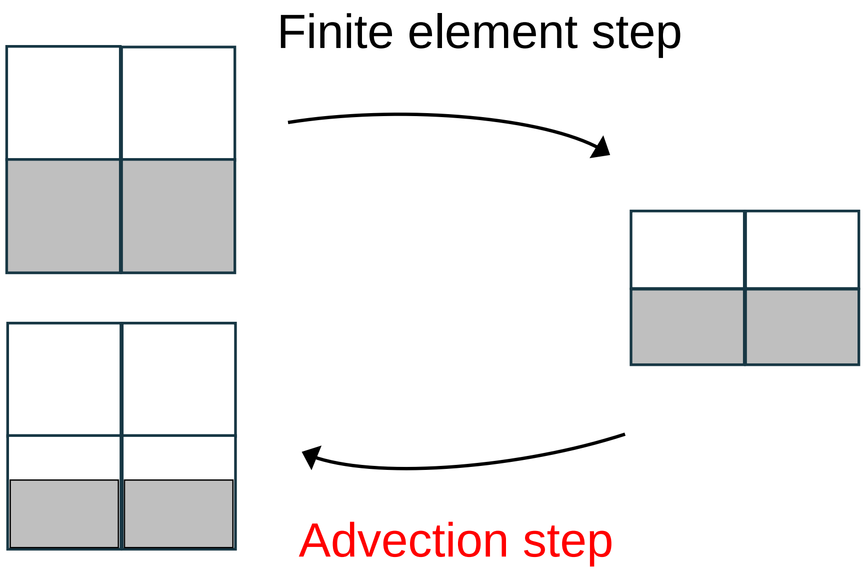
Contact force with filtering

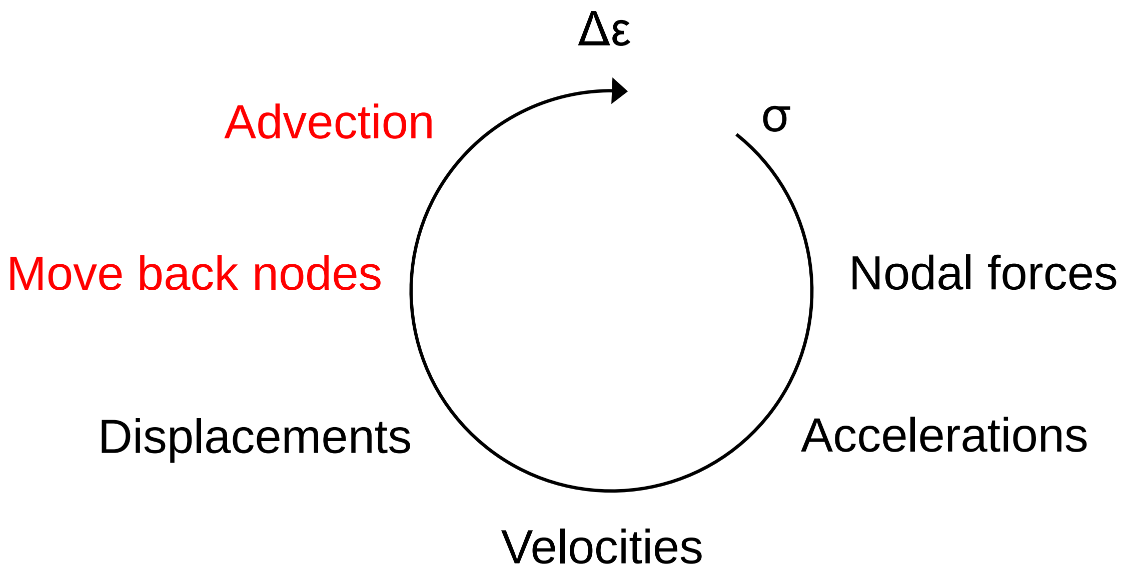
ALE/S-ALE

(Structured) Arbitrary Lagrangian Eulerian

Background ALE - Multi-Material Eulerian formulation

- The steps performed each time step are:
  - Finite element step: Using the explicit central difference method, bulk viscosity, and mixture law. This deforms the mesh. This is a Lagrangian time step.
  - Advection step: The solution on the deformed mesh is mapped back to the original un-deformed mesh including interface reconstruction.





Background ALE

- FEM vs ALE in LS-DYNA
  - Efficiency: ALE is less efficient than FEM in the sense to obtain the same accuracy, for models without excessive strains and deformation, more computational effort and smaller elements are generally needed. One must also mesh the space outside the solids/fluids which can further increase the disadvantage of ALE.
  - Extreme deformations: ALE can handle extreme deformations that FEM cannot.
  - Surface & boundary conditions: In a FEM simulation the boundary of materials/parts are accurately defined making it possible to easily prescribe boundary conditions and contact conditions. This is not the case in ALE. Example if two parts touch in the ALE simulation they will generally stick or weld together.

LS-DYNA ALE keywords

```
*CONTROL_ALE
$#       dct      nadv      meth      afac
           1         1         2      -1.0
```

- **nadv**: Cycles between advection steps
- **meth**: Advection method
- **afac**: -1. Turns smoothing off (Euler method)

```
*CONSTRAINED_LAGRANGE_IN_SOLID
$# lstrsid    alesid  lstrstyp   alestyp     nquad     ctype     direc     mcoup
         2         1         1         0         2         2         1         0
$#   start       end      pfac      fric    frcmin      norm   normtyp      damp
       0.0     1.E10       0.1       0.0       0.5         0         0       0.0
…
```

- **nquad**: Number of coupling points
- **ctype**: Coupling method. Penalty or constraint based.
- **direc**: Tension/compression
- **fric:** Coefficient of friction.
- **pfac:** Penalty factor.

```
*INITIAL_VOID_PART
$#     pid
         1
```

- **pid:** Part can contain material but is initially void. Part material is set to “material”.

```
*SECTION_SOLID
$#   secid    elform       aet
         1        12         0
```

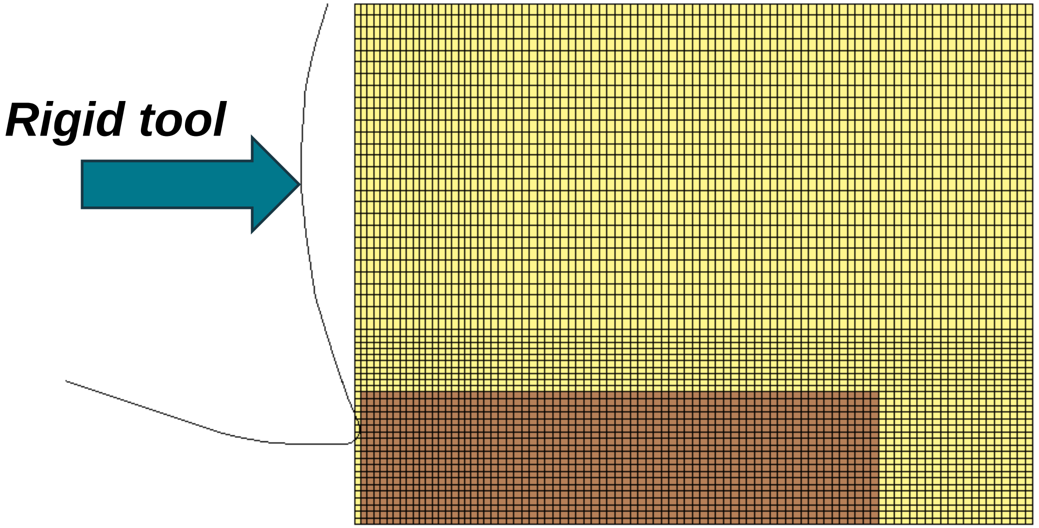
- **elform:**
  - 11 for multimaterial
  - 12 for material and void

Cutting example

Model

```
*INITIAL_VOID_PART
$#       pid
           9
```

Aluminium but initially void



Aluminium

Euler Lagrange coupling output

```
*DATABASE_FSI
$#     dtout     binary
        0.01          1
```

1 point ALE element with material and void

```
*SECTION_SOLID
$#     secid     elform         aet
           1         12           0
```

Full Euler – no smoothing

```
*CONTROL_ALE
$#       dct       nadv        meth        afac
           1          1           2        -1.0
```

Constraint based Euler Lagrange coupling

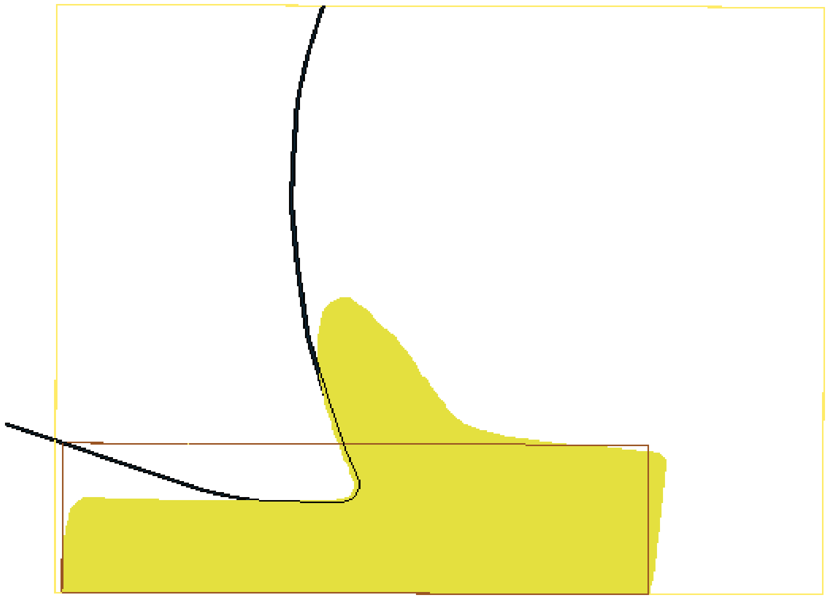
```
*CONSTRAINED_LAGRANGE_IN_SOLID
$# lstrsid    alesid  lstrstyp   alestyp     nquad     ctype     direc     mcoup
         2         1         1         0         4         4         2         0
```

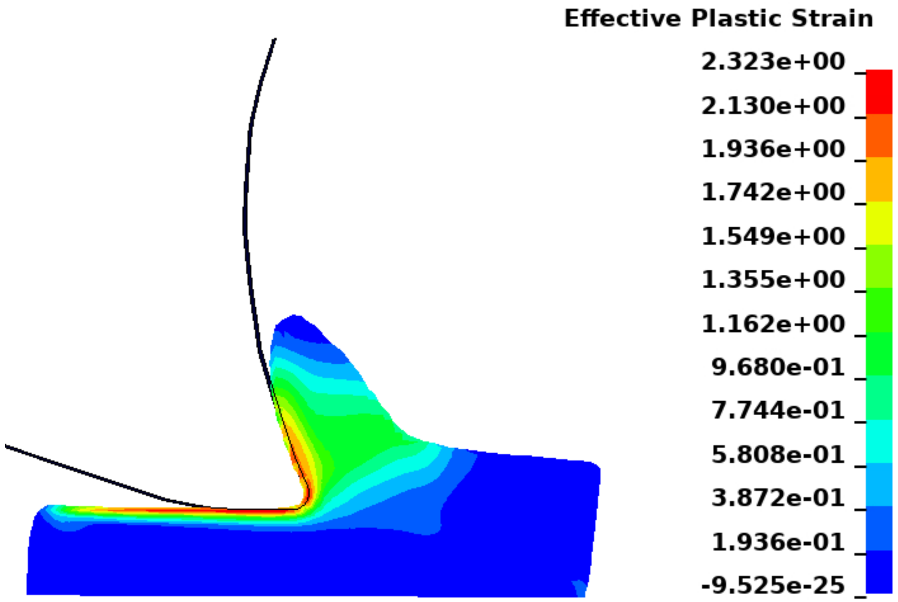
Lagrangian ALE Number coupling points

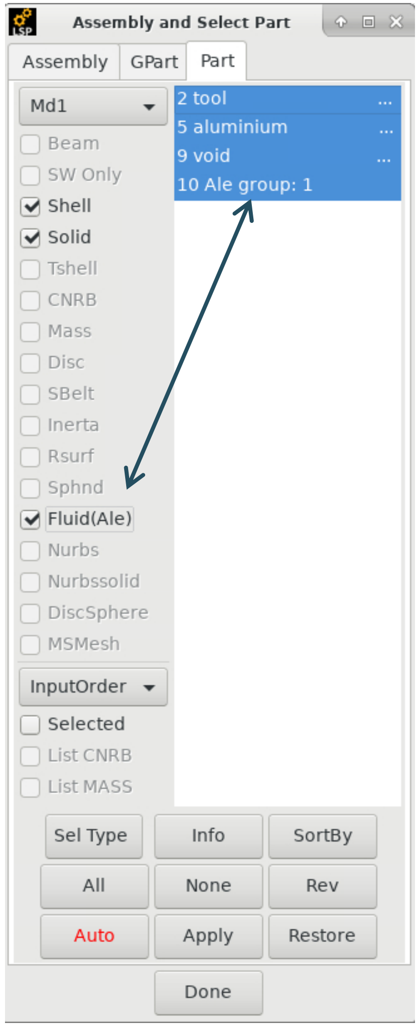
Cutting example

Post processing – LS-PREPOST

- Turn on ALE part in Assembly and Select Part







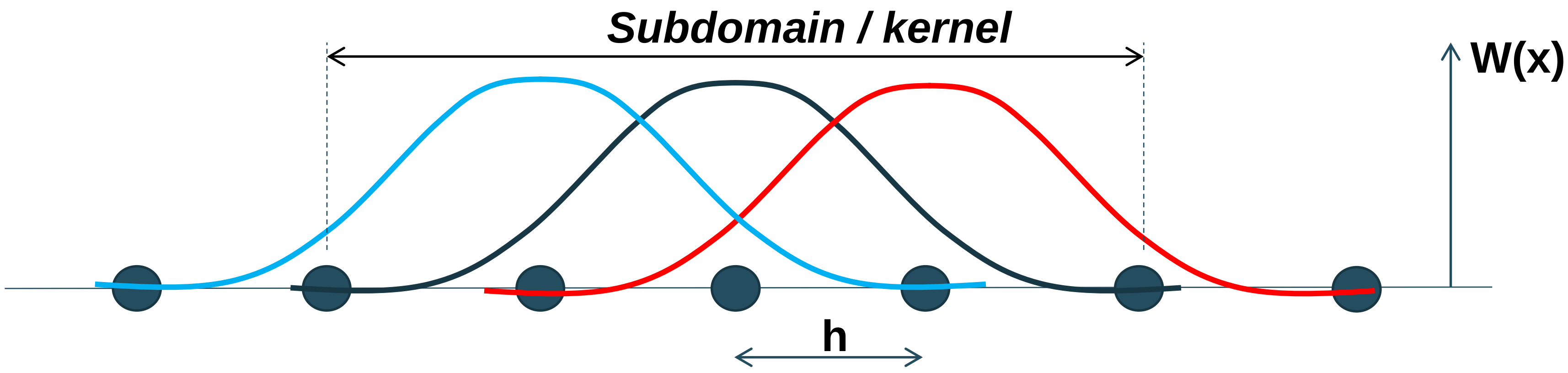
Meshless Methods

SPG/EFG and SPH

Introduction to meshfree methods

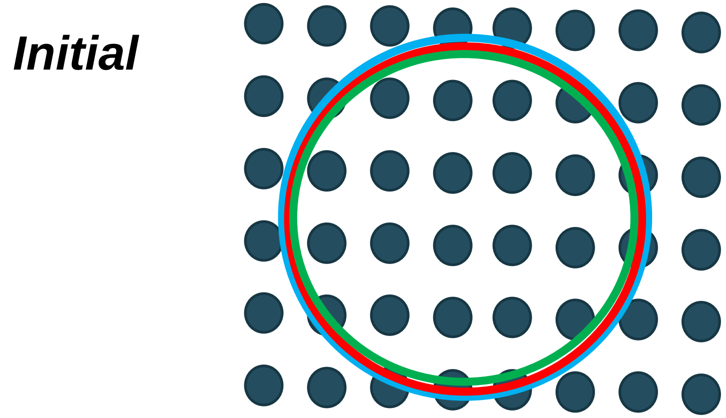
- A meshfree method is not a discrete particle method – It is a continuum based method
  - In meshfree methods the particles serve as computational points for field variables inside a continuum
- The influence between the particles are based on kernel functions
  - The smoothing/interpolating kernel makes the kernel functions continuous and differentiable
- Handles arbitrary large deformations
- Timestep is insensitive to deformation

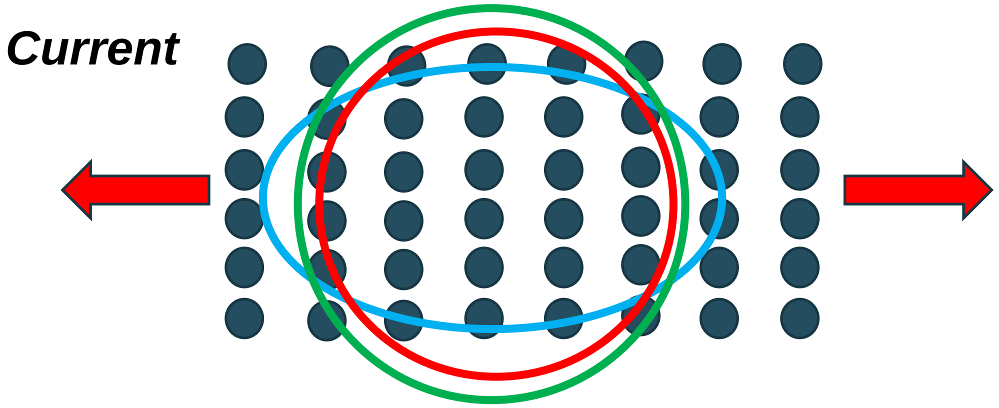
- Drawbacks of the method
  - Treatment of essential boundary conditions is difficult – cannot be described exactly
  - Lack of consistency (shape functions do not add to unity) – renormalization of kernel function necessary
  - Tensile instability – Unphysical/premature failure do to lack of influence between particles



Introduction to meshfree methods - Kernel

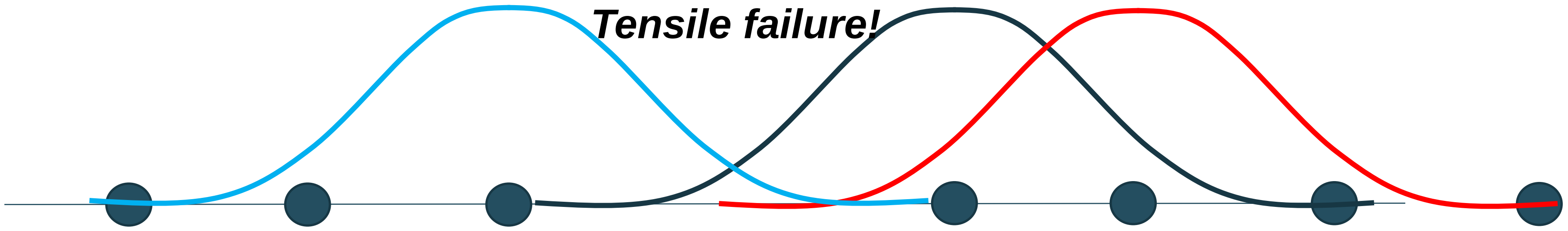
- The type of kernel influences how the deformation of the material changes the structure of the subdomain
- Eulerian: Fixed kernel, the particles are moving in and out of the kernel domain
- Lagrangian: Updated kernel, the particles within the kernel domain stays the same
- Semi Lagrangian: The kernel is defined in the current configuration but the number of particles is the same
- Lagrangian kernel reduces tensile instabilities
- The subdomain size can be scaled
  - Large size is more numerically stable
  - Large size requires more CPU time
  - Ductile material requires a larger size while a semi- brittle can be handled with a smaller size

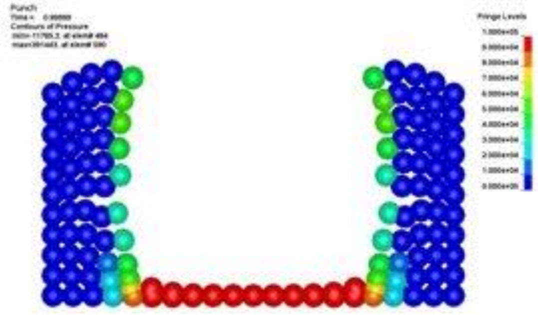




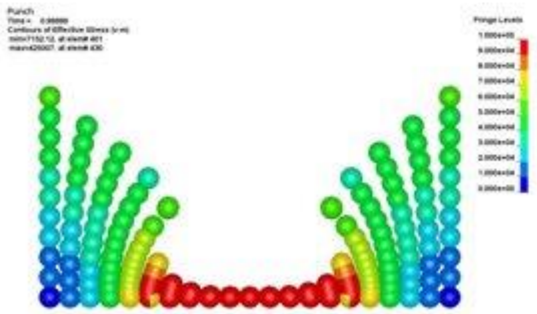
Introduction to meshfree methods – Tensile instabilities

- Tensile instabilitity is a premature failure which occurs when the influence from two nodes becomes too low and the “bond” fails
- The choice of kernel influences the sensitivity for tensile instabilities where in general an Eulerian kernel is more sensitive compared to a Lagrangian kernel





With tensile instability

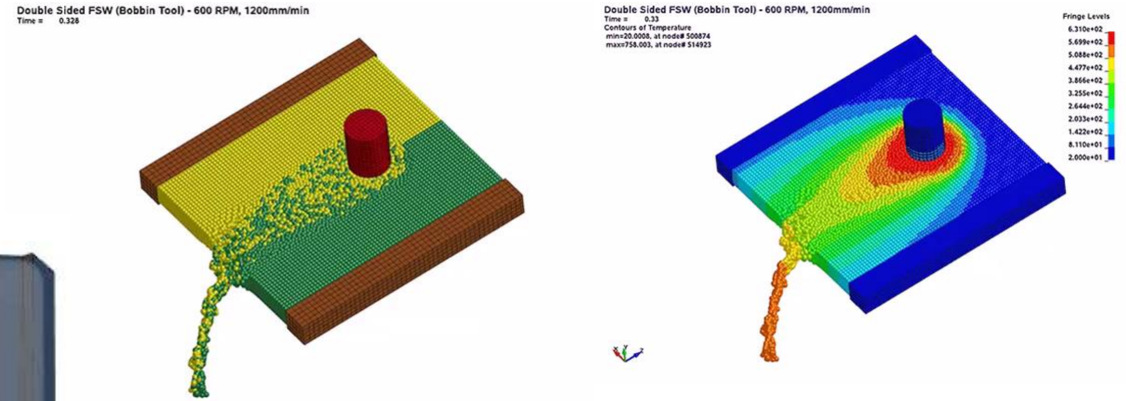


W/O tensile instability

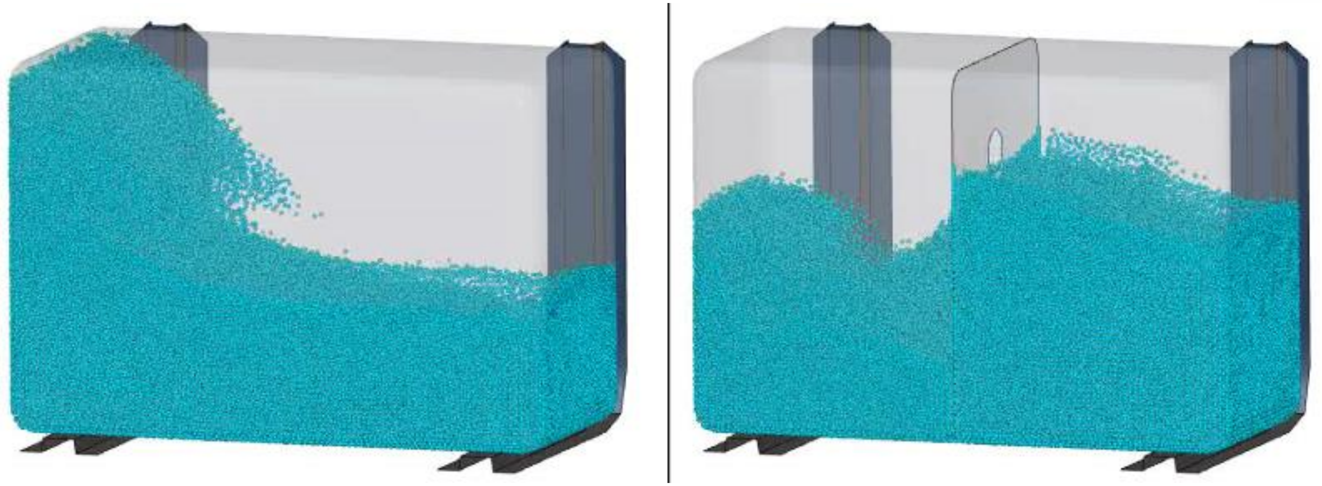
SPH – Smooth Particle Hydrodynamics

- Commonly used for fluids, materials mixing and large deformation simulations
- Uses Collocation method (strong form) to solve partial differentials
- Special feature: FEA to sph conversion

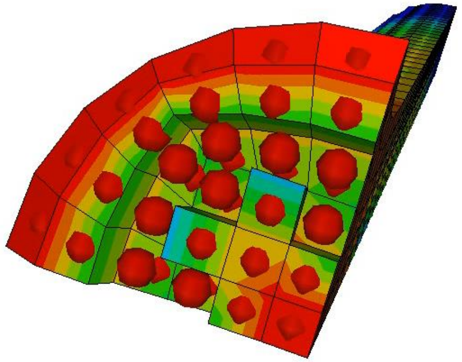
FSW



Sloshing



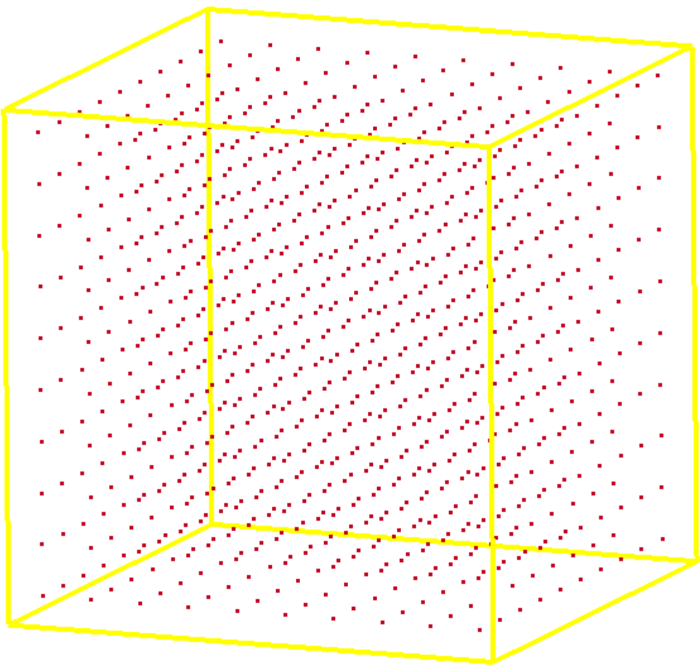
FEM-SPH conversion

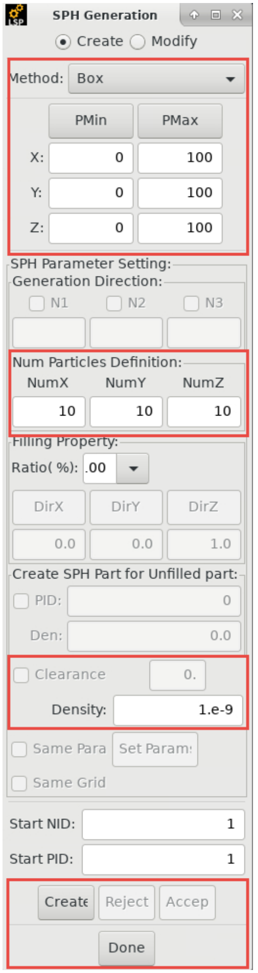


Preprocessing

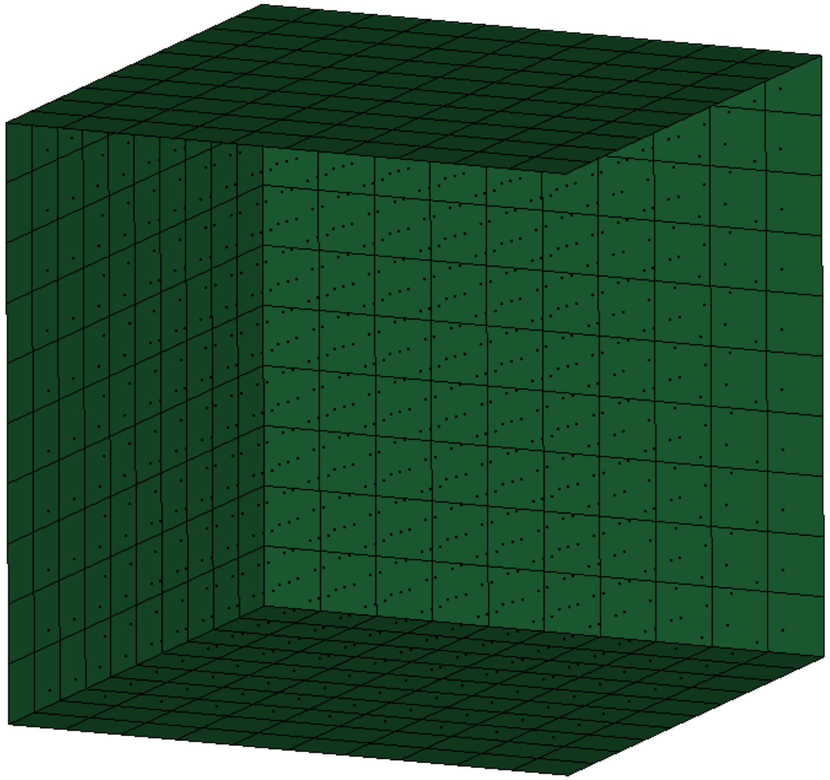
- SPH works best if the spacing between the particles is similar

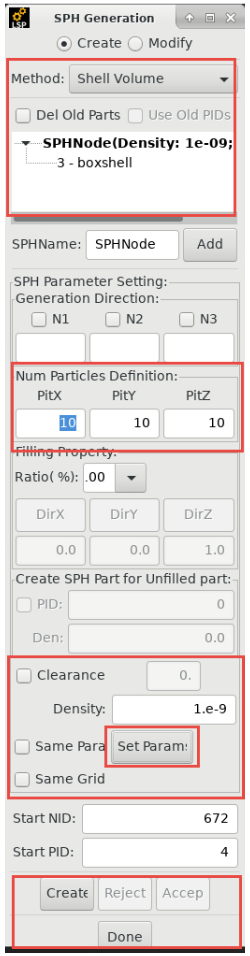
By Geometry (Box, cylinder, cone)





By Shell Volume





LS-DYNA SPH keywords

```
*ELEMENT_SPH
$#    nid      pid              mass
 1005811         1     8.000000e-09
```

- **mass** is calculated by the pre- processor
- The radius of the particle is calculated from the mass and used in the contacts as particle thickness

```
*SECTION_SPH
$#    secid        cslh       hmin       hmax     sphini      death      start    sphkern
          1        1.20        0.2        2.0        0.01.00000E20         0.0          0
```

- Possible scaling of the **initial** smoothing length
- Higher **cslh** increases simulation time
- Smoothing length is variable and **hmin** and **hmax** sets the maximum smoothing lengths
- Default are good values

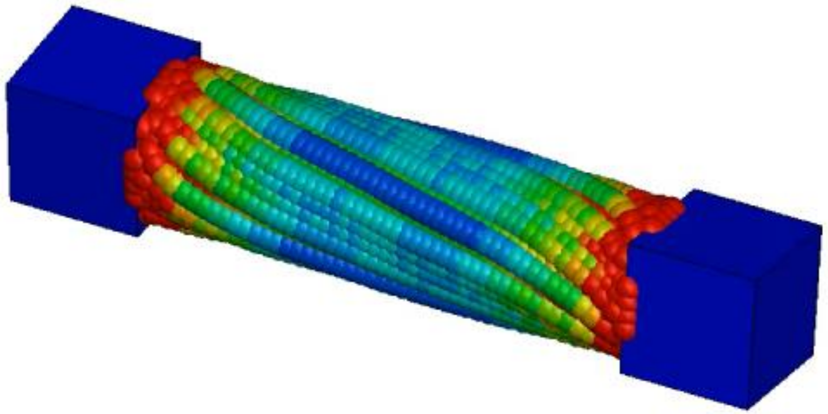
```
*CONTROL_SPH
$#     ncbs     boxid        dt      idim   nmneigh      form     start      maxv
          1         0    1.00E20         3       150         0     0.01.00000E15
$#     cont     deriv       ini     ishow    ierod     icont     iavis     isymp
          0         0         0         0         0         0         0       100
$#     ithk     istab        ql         -   sphsort
          0         0      0.01                   0
```

- **idim**: 2D/3D problem
- **nmneigh**: Might have to be increased depending on model
- **form**: Sets the kernel function type
- Artificial viscosity is always used to dissipate numerical high frequency noise

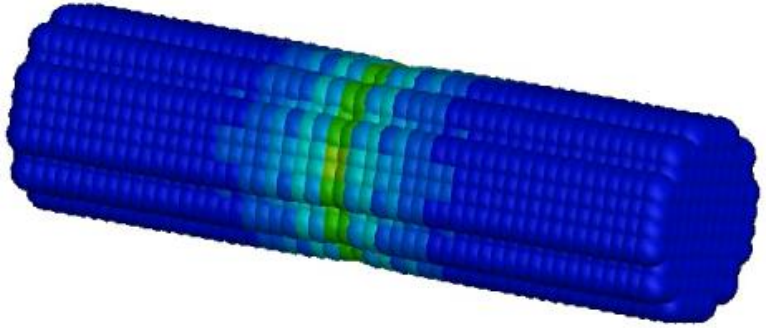
Connect SPH and FEM/Boundary conditions

Building the FE model

- SPH particles can be connected to FEM by `*CONTACT_TIED_NODES_TO_SURFACE_...`



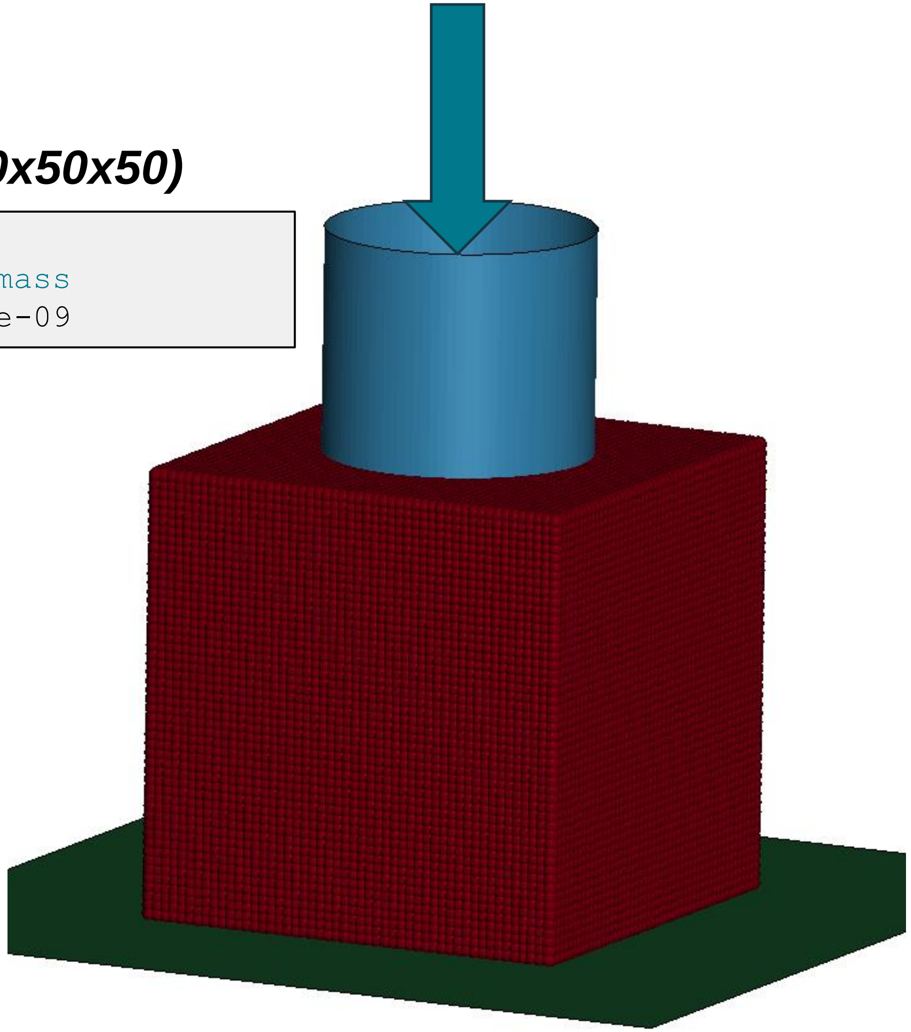
- Boundary conditions (constraints and prescribed motions) using `*BOUNDARY_...`



Punch example

125000 SPH particles (50x50x50)

```
*ELEMENT_SPH
$#    nid       pid                 mass
 1005811         1     8.000000e-09
```



Automatic nodes to surface contacts with set particle radius

```
*CONTACT_AUTOMATIC_NODES_TO_SURFACE
$#     surfa     surfb  surfatyp  surfbtyp
           6         2         4         3
$#        fs        fd        dc        vc
         0.5       0.0       0.0       0.0
$#       sfs       sfm       sst       mst
         1.0       1.0       1.0       0.0
```

Default smoothing lengths

```
*SECTION_SPH
$#     secid      cslh      hmin      hmax
           3      1.20       0.2       2.0
```

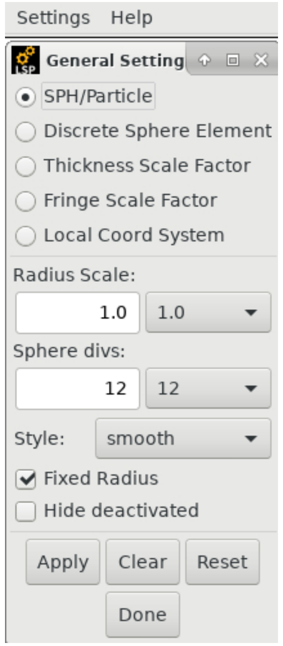
Default Eulerian kernel

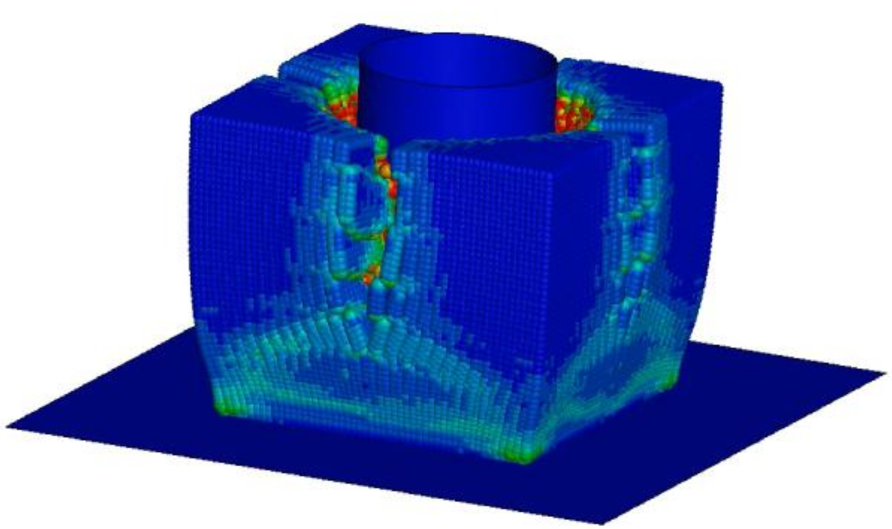
```
*CONTROL_SPH
$#      ncbs     boxid        dt      idim   nmneigh
           1         01.00000E20         3       150
```

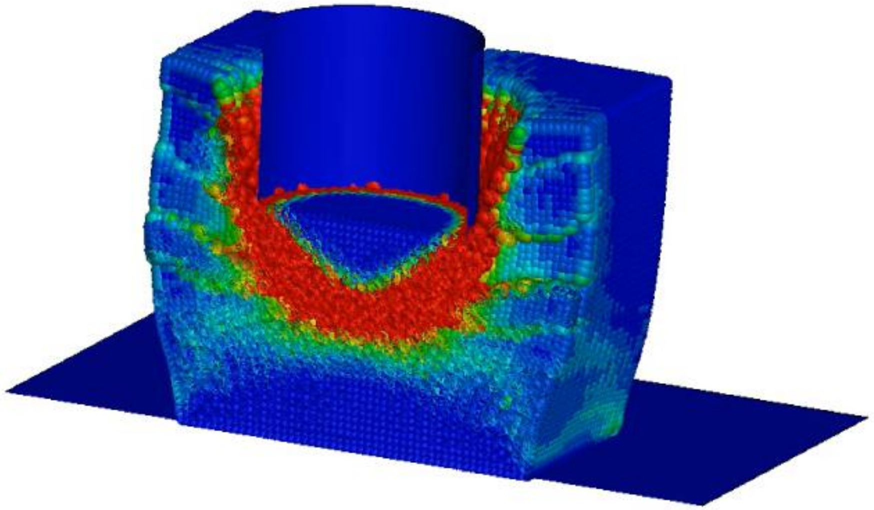
Postprocessing

General Settings

- The postprocessing settings for SPH is found under General Settings
  - Radius scale factor. 1. is actual particle size
  - Sphere divs sets how smooth the visualization is
  - Style sets the particle style
- Fringe plot and cut model as usual



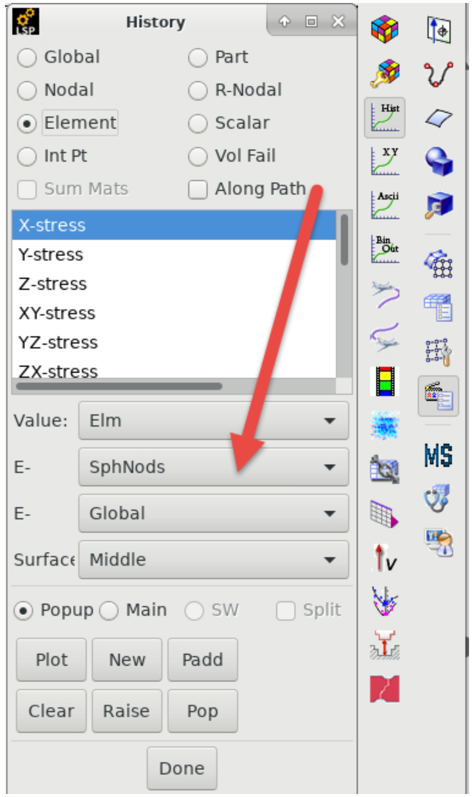


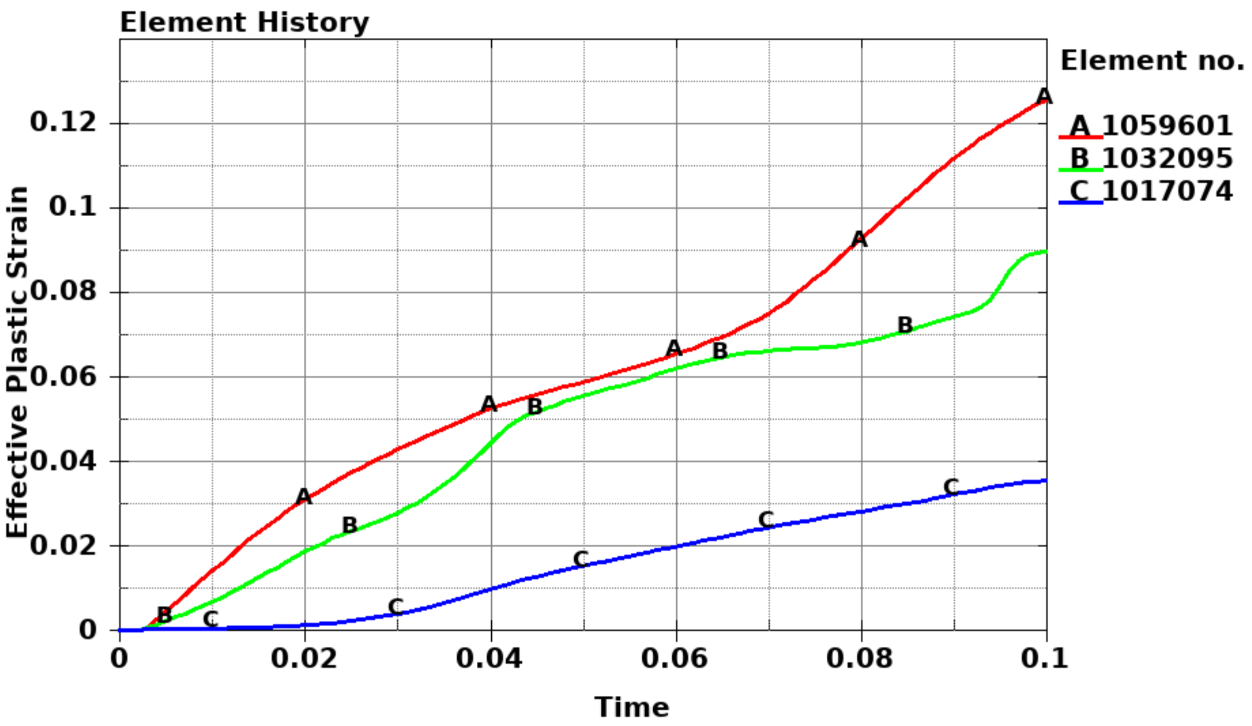


Postprocessing

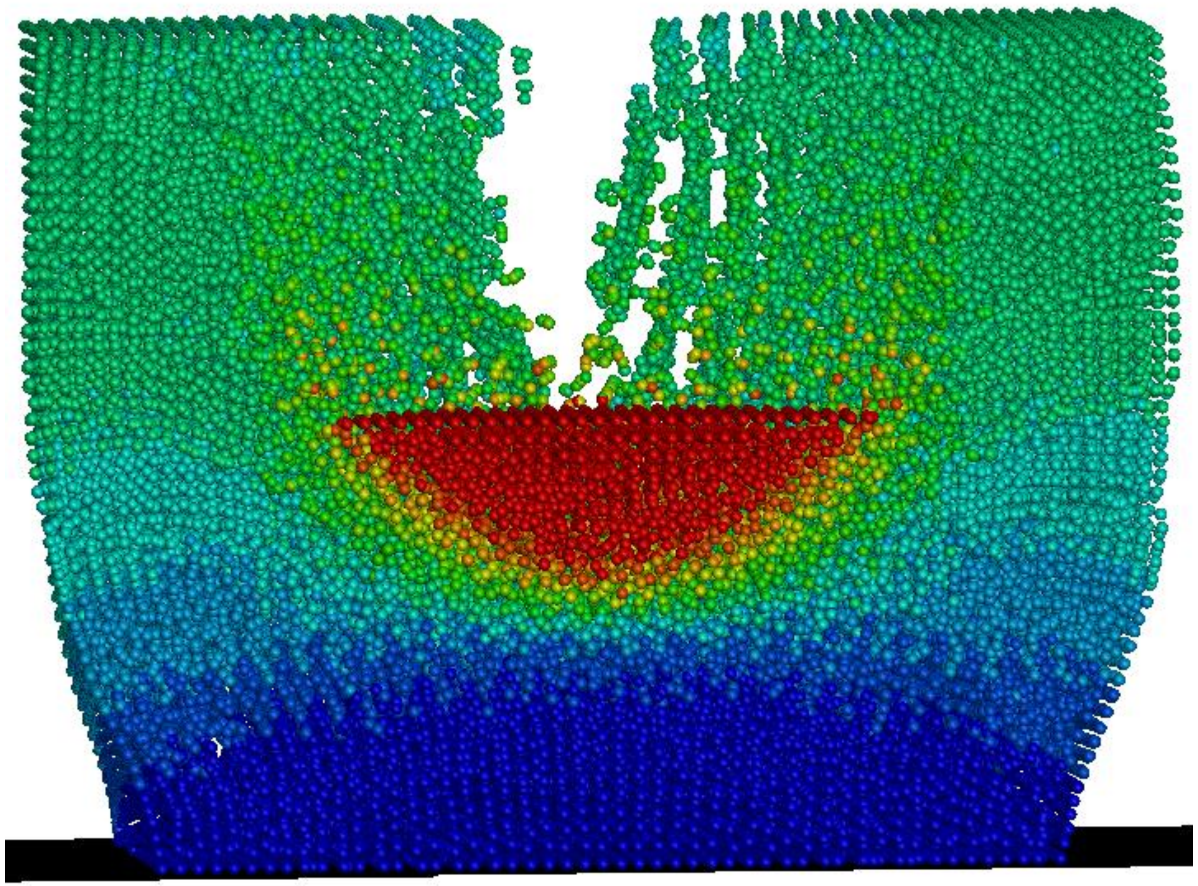
Histories

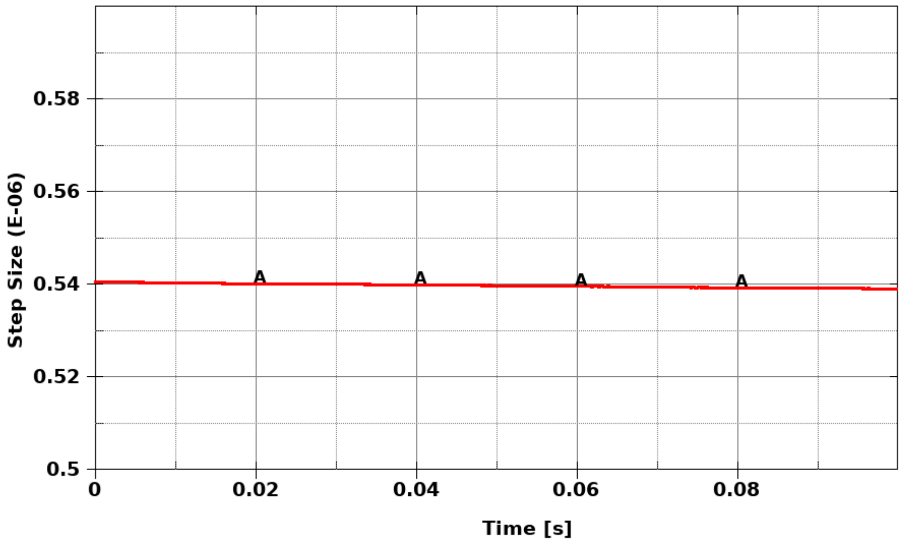
- Particle data can be History plotted as usual
  - Stresses
  - Displacements/Velocities

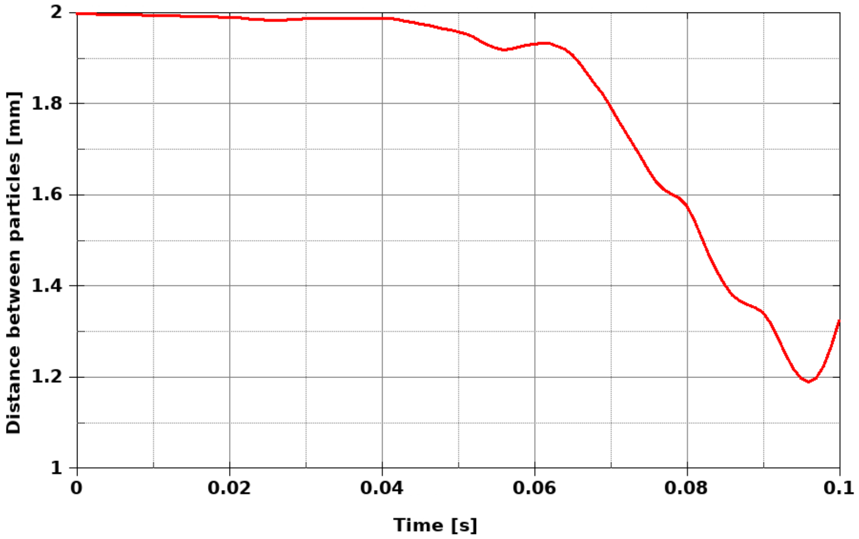




Punch example – Timestep



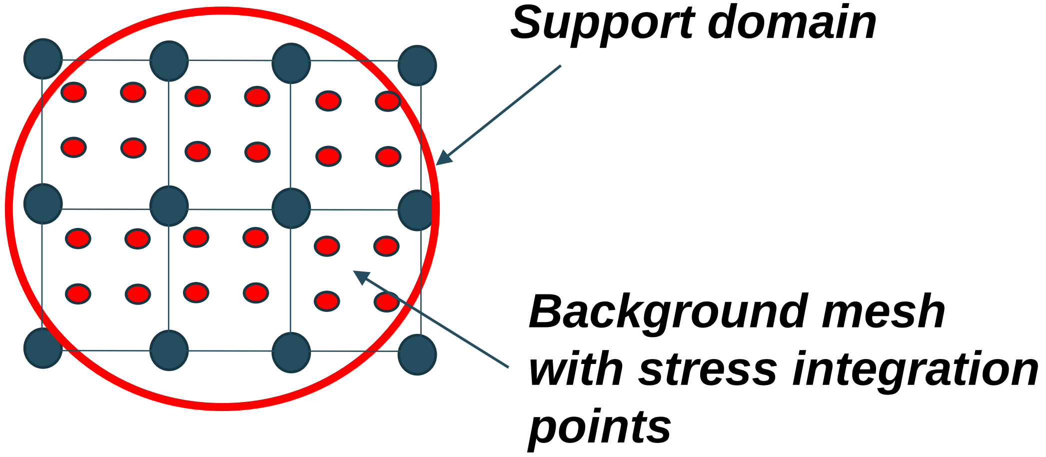




EFG – Element Free Galerkin

- EFG is a mix between FEM and SPH
  - Both EFG and other particle methods has a support domain where the approximated field variables are computed
  - EFG has a background mesh used to integrate the weak forms
- Approximating polynomials are constructed by minimizing the error with respect to the field values through Moving Least Squares (MLS)

- Background mesh makes an easier interpretation of the physical domain
  - Easier for boundary conditions and contacts
- Mesh distorsion issues!
  - Limits are further than for FEM
- Can be combined with 3D adaptivity to avoid mesh distorsion
- Background mesh makes it possible to use for coupled analyses



Preprocessing

- The EFG particles are created automatically from the nodes of a solid element mesh
- Create the EFG background mesh as a usual solid element mesh (tetrahedrons, pentahedrons or hex)
- Change from **`*SECTION_SOLID`** to **`*SECTION_SOLID_EFG`**

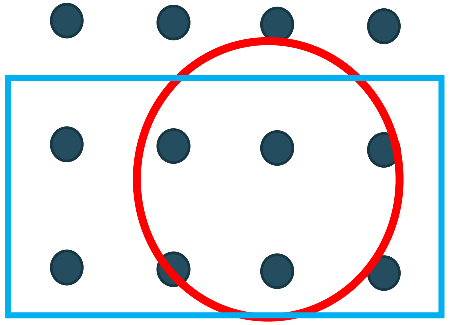
- The same 3D adaptive keywords apply to the EFG remeshing setup

- Since EFG method is formulated with a background mesh, contacts, loads etc is applied as for a regular lagrangian FEM model

LS-DYNA EFG keywords

```
*SECTION_SOLID_EFG
$#    secid    elform       aet    unused    unused    unused    cohoff    unused
          1        42         0                                       0
$#       dx        dy        dz   ispline     idila      iebt      idim    toldef
        1.1       1.1       1.1         0         0         1         2      0.01
$#      ips
          1
```

	regular mesh	irregular mesh
Foam	1.0 ~ 1.2	1.0 ~ 1.2
Metal	1.2 ~ 1.4	1.0 ~ 1.2
Fluid or EOS	1.4 ~ 1.6	1.2 ~ 1.4



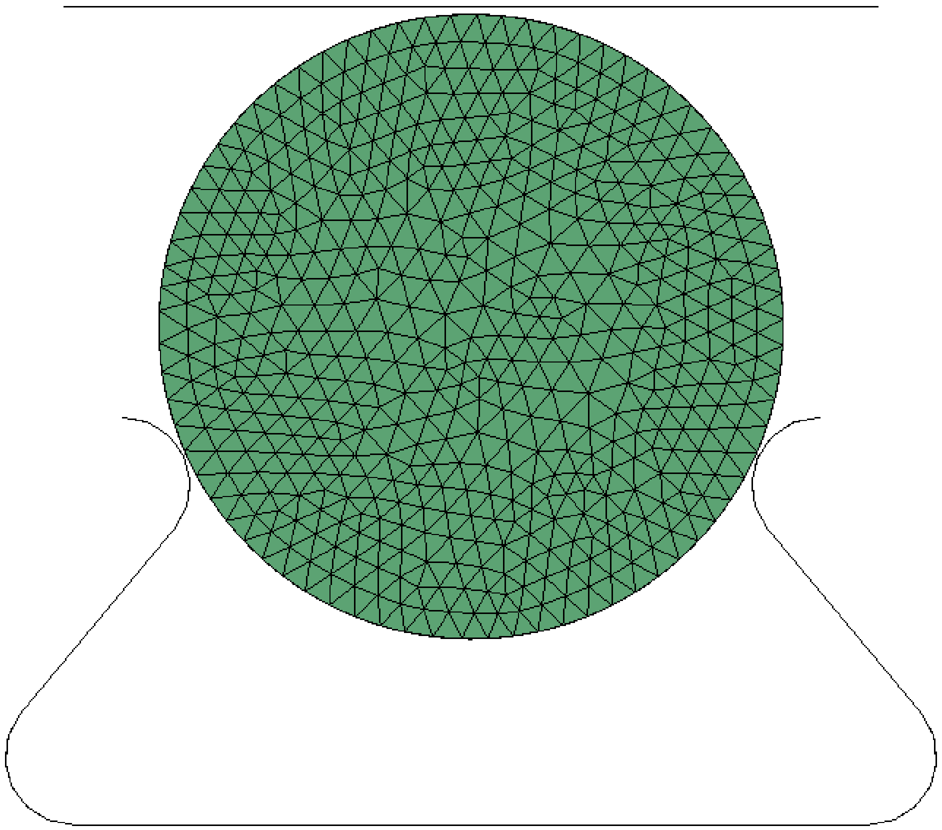
- **elform**: 41 is hex/penta and 42 is tetrahedron
- **dx, dy, dz**: Scales the dilatation parameters
- **ispline**: Choice of kernel function Quadratic/cubi, rectangular/circle support
- **idim**: Weak form integration. Local boundary or Gauss integration.
- **toldef:** Deformation tolerance for the activation of adaptive EFG Semi-Lagrangian and Eulerian kernel
- **ips**: Pressure recovery (ELFORM=42)

```
*CONTROL_REMESHING_EFG
$#     rmin      rmax   vf_loss     mfrac    dt_min     icurv       cid    segang
       0.01       0.1
$#      ivt       iat      iaat       ier        mm

$#     iat1      iat2      iat3
```

- **rmin, rmax**: Minimum and maximum edge length
- **iat:** Flag for interactive adaptivity
- **iat1, iat2, iat3**: Tolerance for interactive criteria: shear strain, element edge ratio and volume change

Rubber compression example



Automatic surface to surface contact

```
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE
$#     surfa     surfb  surfatyp  surfbtyp
           6         2         4         3
$#        fs        fd        dc        vc
        0.05       0.0       0.0       0.0
```

Default kernel

```
*SECTION_SOLID_EFG
$#     secid    elform       aet
           1        42         0                                       0
$#        dx        dy        dz   ispline     idila      iebt      idim
         1.1       1.1       1.1         0         0         1         2
```

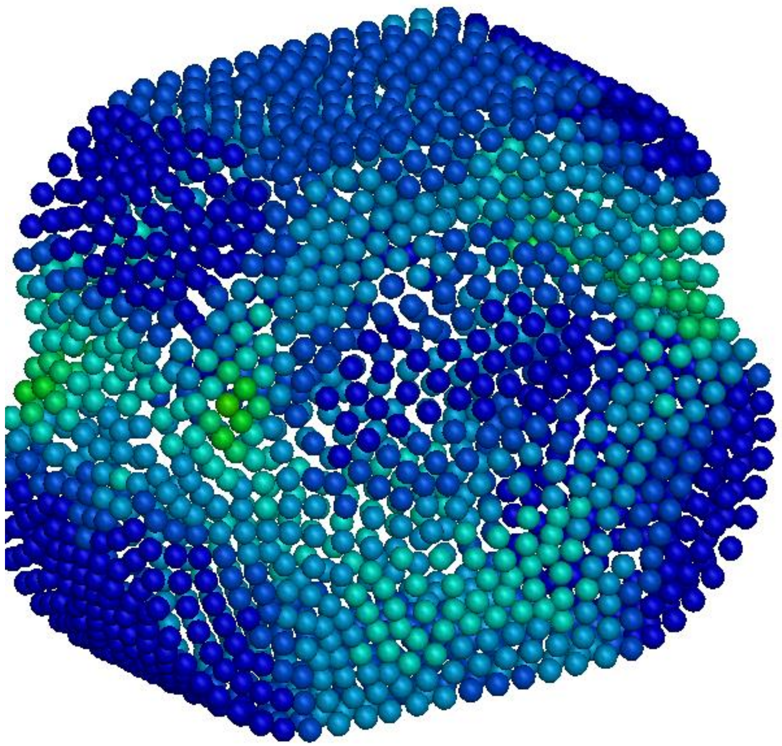
Constant adaptive frequency

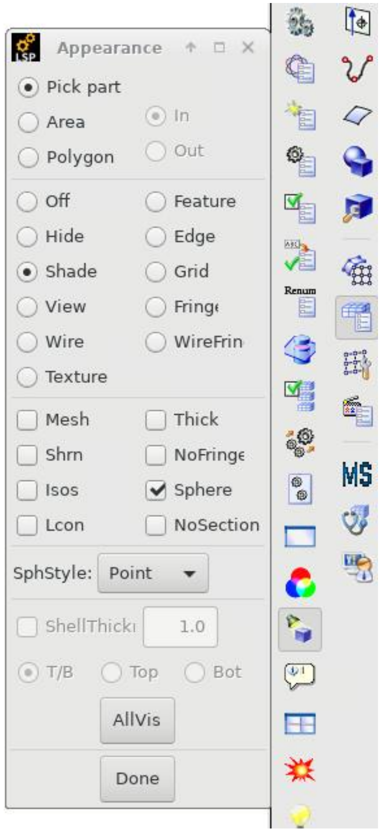
```
*CONTROL_ADAPTIVE
$# adpfreq    adptol    adptyp
      0.011.00000E20         7
```

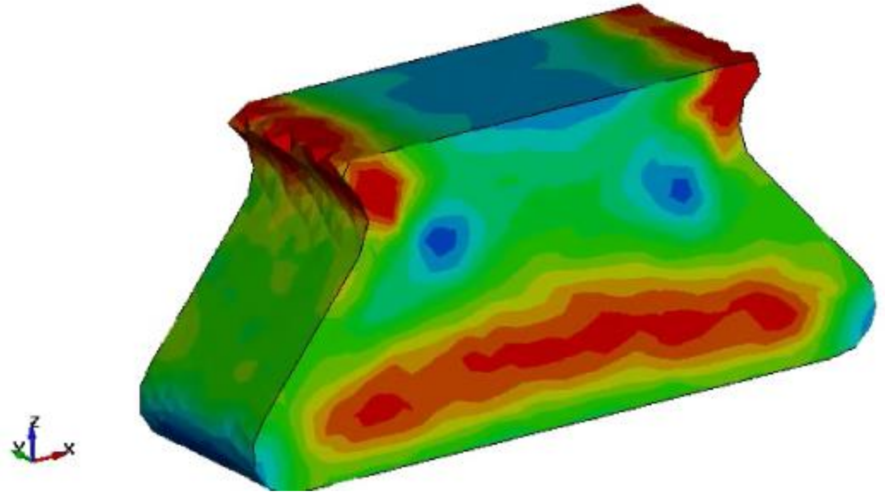
Element length based remeshing

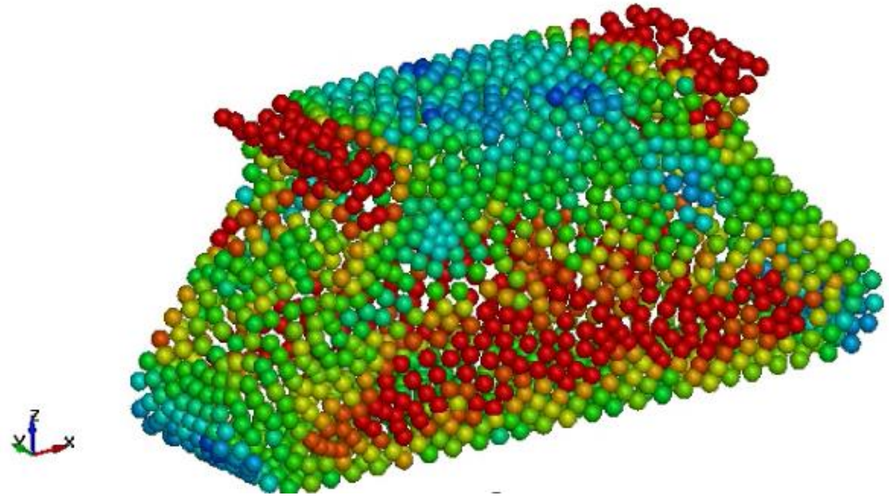
```
*CONTROL_REMESHING
$#      rmin      rmax   vf_loss
         1.5       2.5       1.0
```

Postprocessing









Spherical nodal representation

SPG – Smooth Particle Galerkin

- SPG is a truly mesh free method similar to SPH
- SPG has a Galerkin weak formulation
- A displacement smoothing technique is applied to stabilize low energy modes occurring from the direct nodal integration (DNI)
  - The displacement smoothing function adds a stabilizing term to the equilibrium equation
- Bond based failure is possible by removing the failed bond in the neighbour sorting algorithm
- Bond based failure criteria in **`*SECTION`** card
- Damage model by **`*MAT_ADD_DAMAGE_GISSMO`**
- Timestep is not influenced by deformation

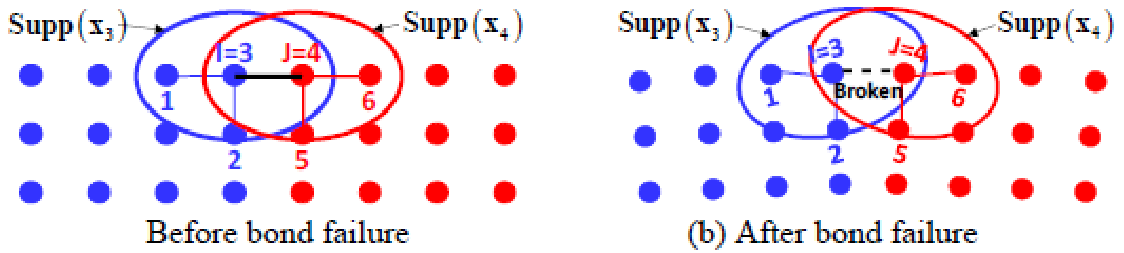


Figure 1. Illustration of SPG bond failure mechanism.

Preprocessing

- The SPG particles are created automatically from the nodes of a solid element mesh
- Change from **`*SECTION_SOLID`** to **`*SECTION_SOLID_SPG`**
- LS-DYNA automatically changes from FEM to SPG, no SPG particle definition needed
- SPG particles can be connected to FEM by node merge or tied contacts.
- Use node to surface contacts for SPG to FEM contact
- Multiple stage analysis is possible. **`*INTERFACE_SPG_1`** outputs a result file and **`*INTERFACE_SPG_2`** indicates that the spg parts should be initialized from the result file.

LS-DYNA SPG keywords

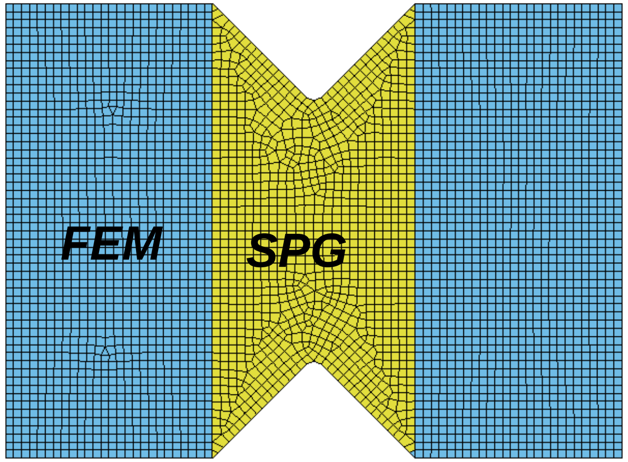
```
*SECTION_SOLID_SPG
$#   secid    elform       aet    unused    unused    unused    cohoff    unused
         3        47         0                                       0
$#      dx        dy        dz   ispline    kernel              smstep       msc
       1.5       1.5       1.5         0         1       0.0         0       0.0
$#    idam        fs   stretch       itb    unused    unused    unused     pdamp
         1       0.4       0.0         3                                  -0.001
```

- **elform**: 47 is SPG element
- **dx, dy, dz**: Scales the dilatation parameters. 1.4 to 1.8 are recommended
- **ispline**: Choice of kernel function Quadratic/cubic
- **kernel**: Choice of kernel
- **idam**: Bond failure mechanism (e.g. effective plastic strain, shear strain..)
- **fs**: Critical bond fail value
- **itb**: Stabilization option
- **pdamp**: Particle to particle damping (itb=3)

```
*CONTACT_SPG
$#    pid1      pid2      pid3      pid4      pid5      pid6      pid7      pid8
         1         2
$#  iself1    iself2    iself3    iself4    iself5    iself6    iself7    iself8
         1
$#   pfac1     pfac2     pfac3     pfac4     pfac5     pfac6     pfac7     pfac8
       0.1       0.1
$#      fs        fd        dc     nfreq
      0.12
```

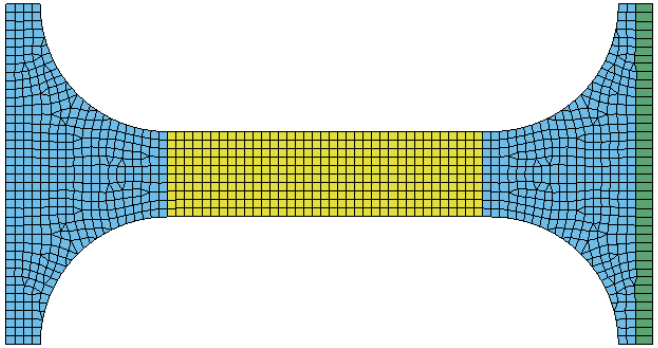
- **pid#**: SPG part in contact
- **iself#**: Self contact flag
- **pfac#**: Penalty factor
- **fs**: Friction coefficient
- **nfreq**: Contact update frequency

SPG and GISSMO damage model example



Eulerian kernel with MCSPG stabilization

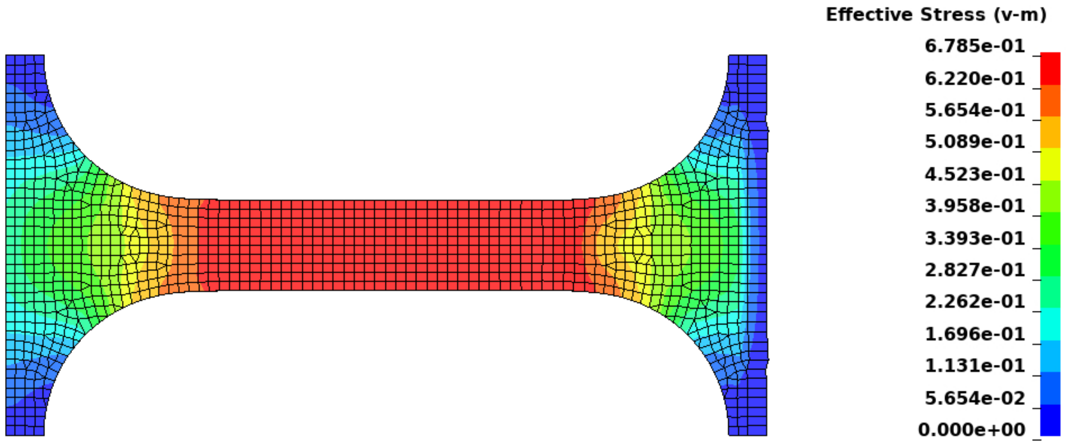
```
*SECTION_SOLID_SPG
$#   secid    elform       aet    unused    unused    unused    cohoff    unused
         2        47         0                                       0         0
$#      dx        dy        dz   ispline    kernel    lscale    smstep    swtime
       1.8       1.8       1.8         0         1       0.0         0       0.0
$#    idam        fs   stretch       itb    unused    unused    unused    unused
         01.00000E28       0.0         3                                  -0.005
```

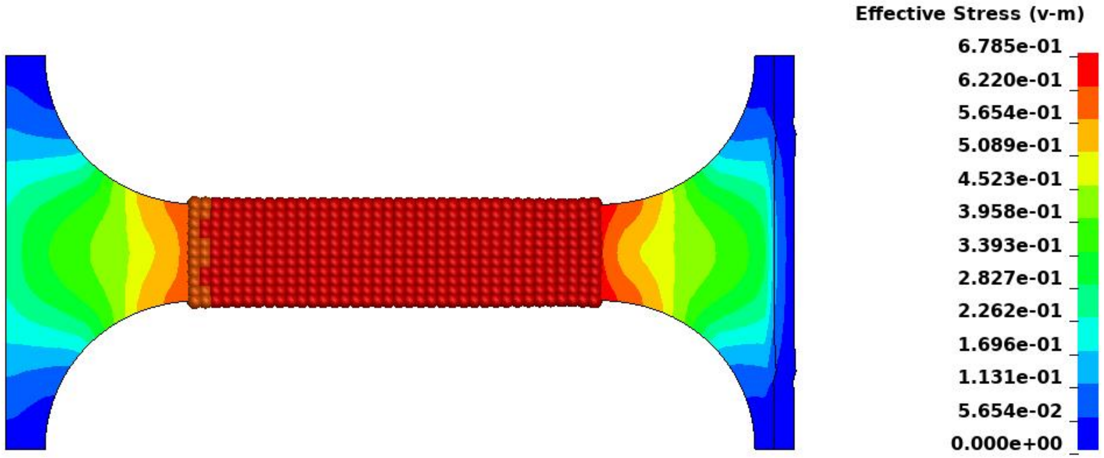


Add on GISSMO failure model

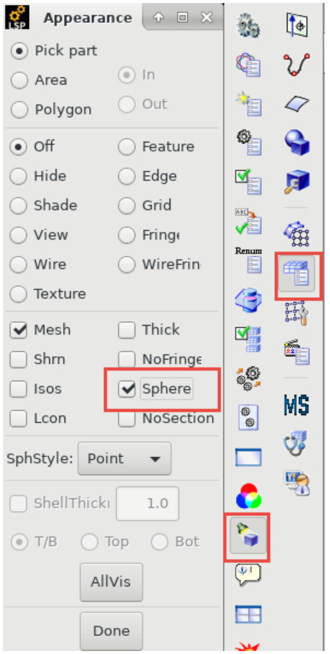
```
*MAT_ADD_DAMAGE_GISSMO
$#     mid         -      dtyp     refsz    numfip
         5               1.0       0.0       1.0
$#   lcsdg     ecrit    dmgexp     dcrit    fadexp    lcregd
         5       0.1       1.0       0.0       1.0         4
$#   lcsrs      shrf     biaxf    lcdlim   midfail     hisvn      soft     lp2bi
         0       0.0       0.0         0       0.0       0.0       0.0       0.0
```

Postprocessing SPG

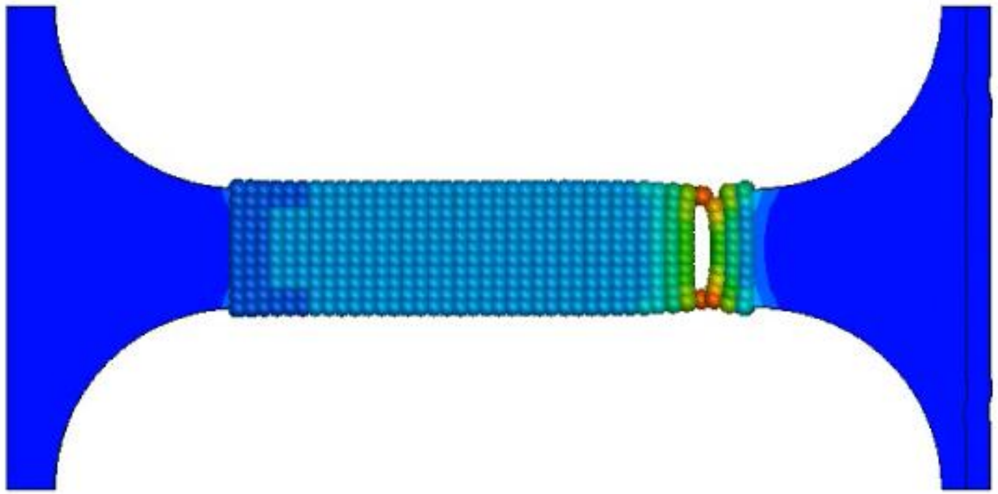


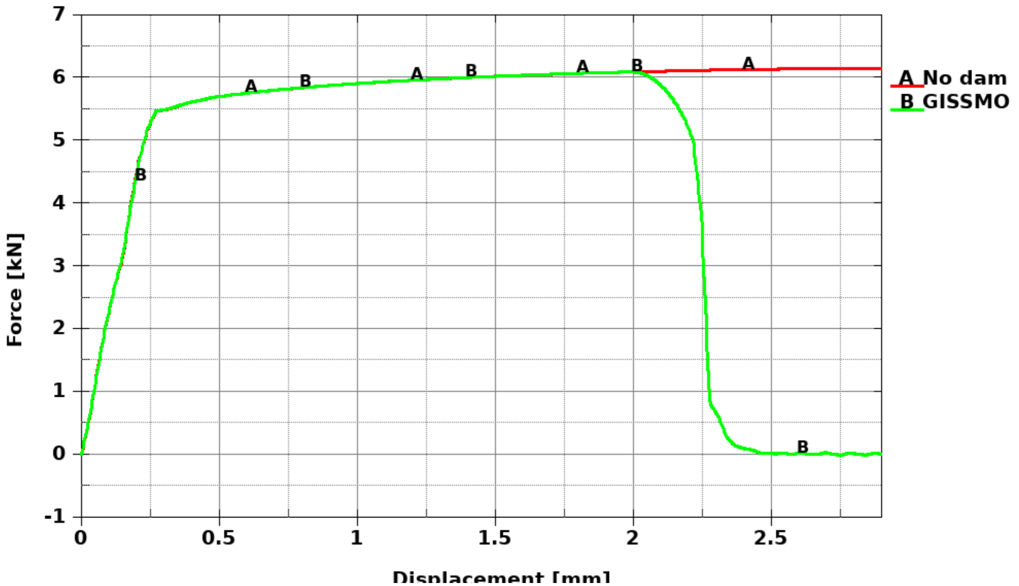


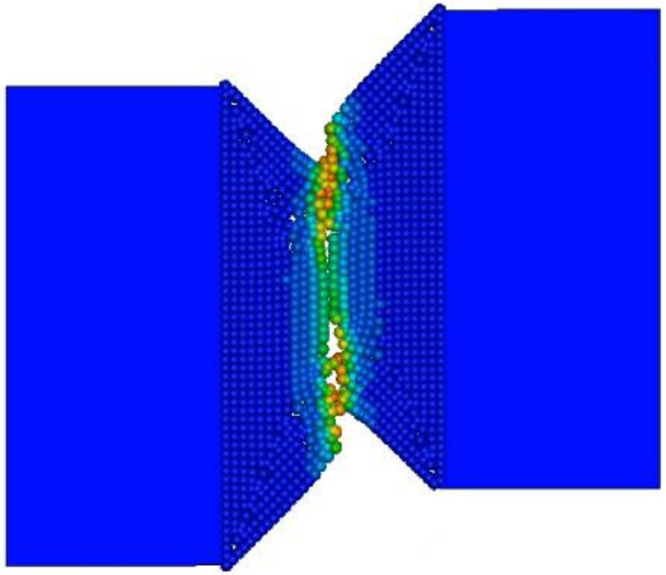
- Results can be postprocessed on element or particle
- Switch to Spere representation under Model and Part/Appear

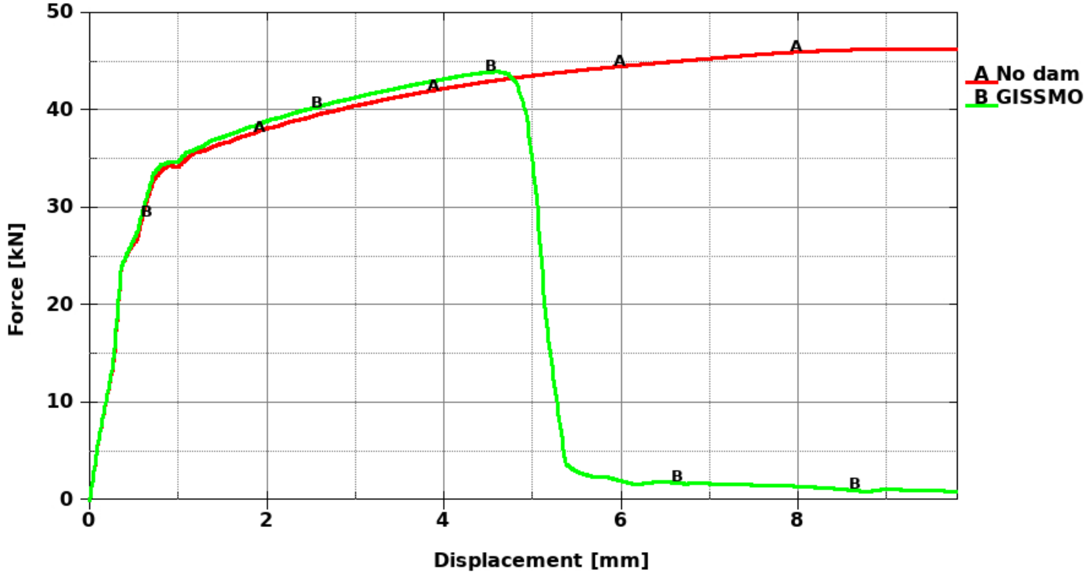


Postprocessing SPG









Summary

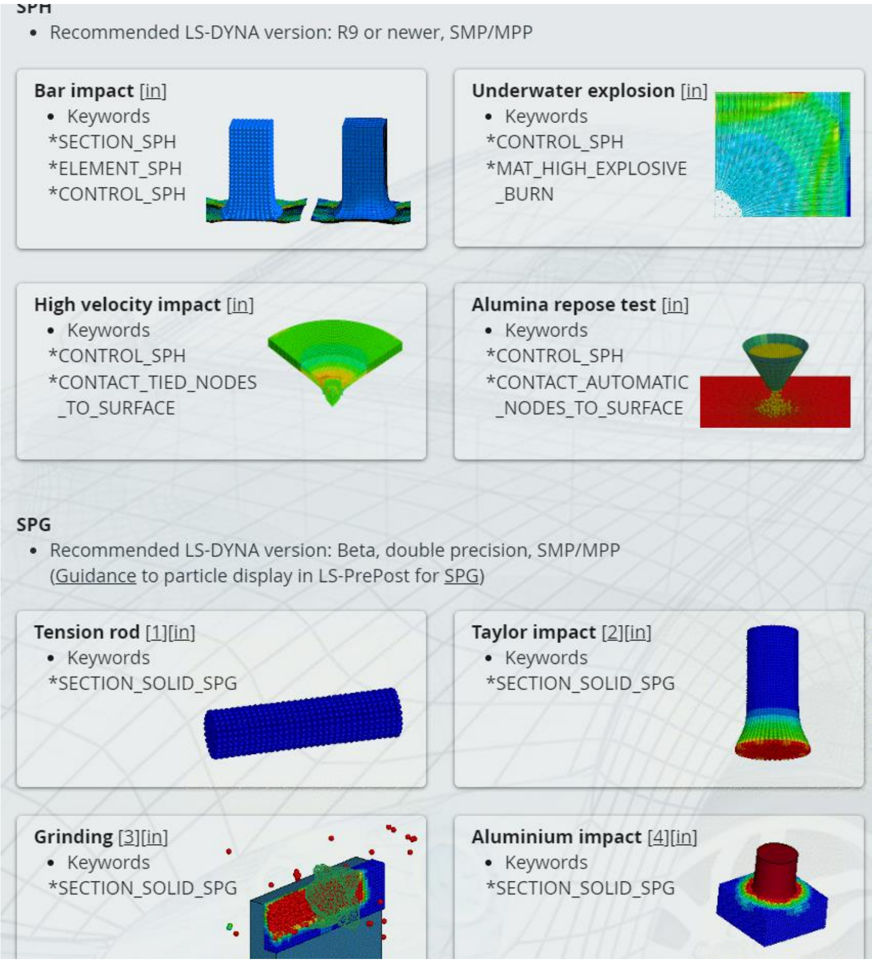
Summary

- LS-DYNA has several options if you want to do large deformation analysis
- The methods ranges from Lagrangian FEM with remeshing to Eulerian FEM, through EFG to element free methods like SPG and SPH
- The type of spatial discretization you choose depends on your application and the aim of your analysis.

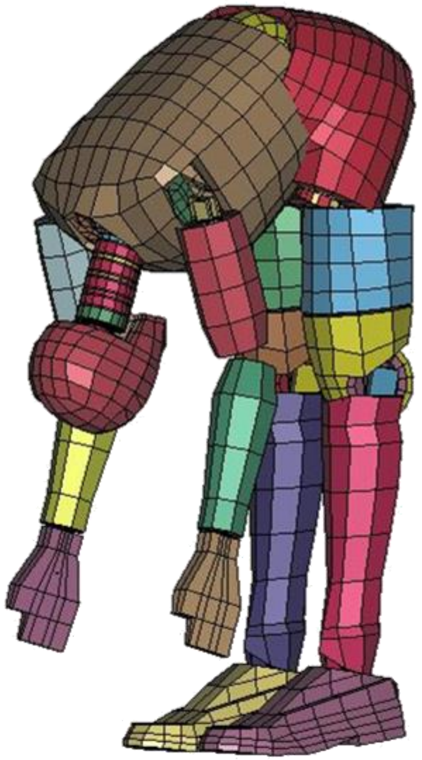
	FEM	Remeshing	ALE/S-ALE	EFG	SPH	SPG
Large Deformations	No	Yes	Yes	Yes (with adaptivity)	Yes	Yes
Failure	Damage	Material model (no separation)	Material model	Material model (no separation)	Material model	Damage
Thermal	Coupled	Coupled	Adiabatic	Coupled	Adiabatic	Adiabatic
Primary application	Structural	Forging/ Extrusion	Fluid/High speed impact	Forging/ Extrusion	Fluids/Cutting	Failure

Further information

- LST Computational and Multi-Scale Mechanics group
  - https://www.lstc-cmmg.org/multiphysics
- YouTube
  - Dynamore Express
    - Beyond FEA - The Element-Free Galerkin (EFG) Method
    - Beyond FEA - Smoothed Particle Hydrodynamics (SPH)
    - Beyond FEA - Arbitary Lagrangean-Eulerian (ALE) Method
- Dynalook
  - Papers from European and International conferences



Thank You



DYNAMore Nordic AB
Brigadgatan 5
587 58 Linköping, Sweden

Tel.: +46 - (0)13 23 66 80
info@dynamore.se

www.dynamore.se
www.dynaexamples.com
www.dynasupport.com
www.dynalook.com

Find us on

DYNAmore worldwide
Germany ▪ France ▪ Italy ▪ Sweden ▪ Switzerland ▪ USA