Beyond FEM:
The Element-Free Galerkin Method

Maik Schenke

5 August 2020

About me

- B. Sc. at University of Applied Science in Aachen
- Stress engineer at Airbus AG (Vertical Tail Plane A380)
- M. Sc. in Computational Mechanics (COMMAS) at University of Stuttgart
- Ph. D. at Institute of Applied Mechanics (civil engineering)
  - Material modelling of porous media
  - Soil mechanics
  - Numerical methods
- Since August 2017 at DYNAmore GmbH
  - Course director
  - Marketing
  - Technical support for Multiphysics

DYNAmore GmbH
Industriestr. 2
70565 Stuttgart
Germany

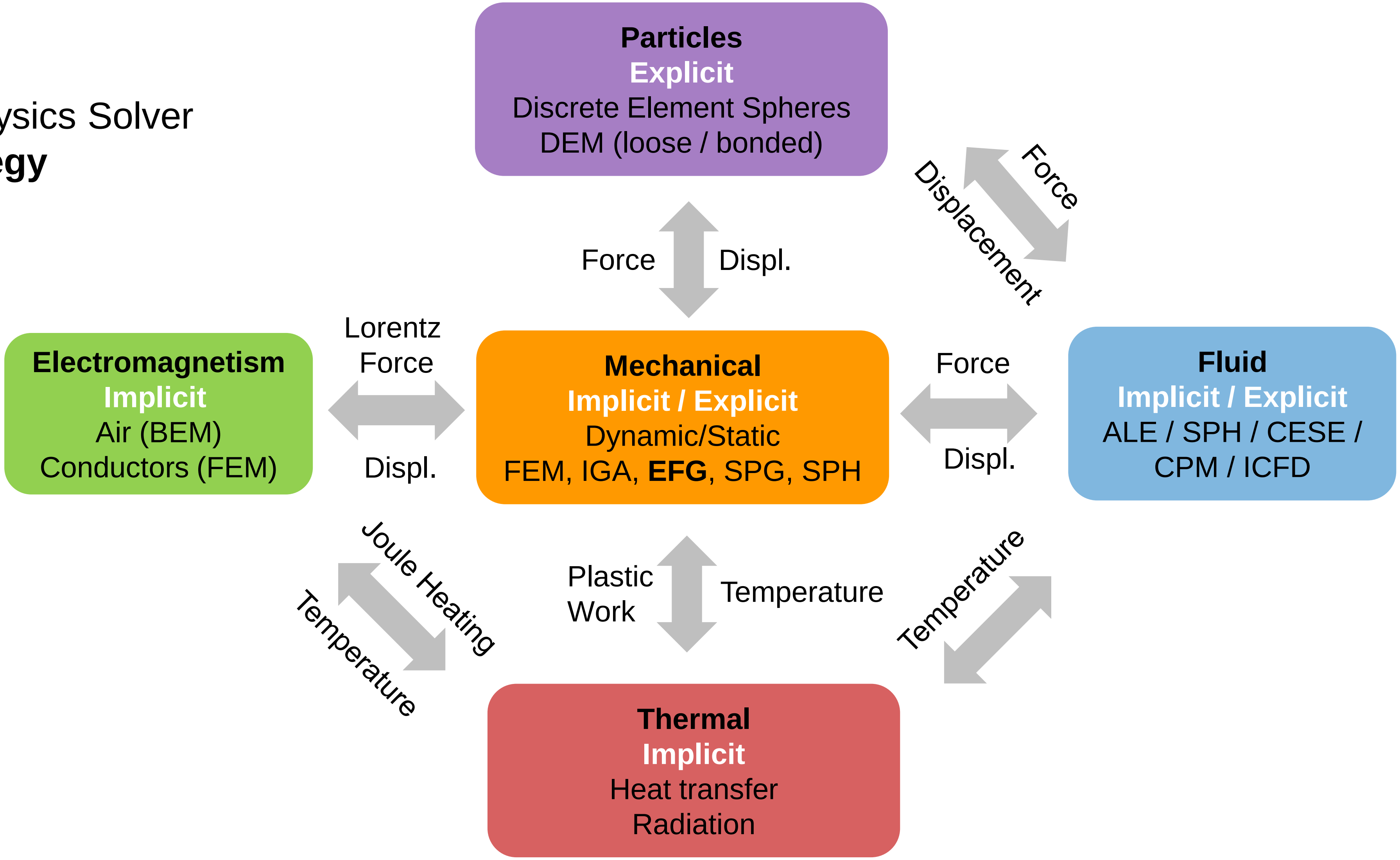
@ maik.schenke@dynamore.de
☎ +49 – (0)711 – 459 600 22
🖷 +49 – (0)711 – 459 600 29

Outline

- Introduction
- Element-Free Galerkin (EFG) Method
- EFG Usage in LS-DYNA
- Application Example
- Summary

Introduction

- LS-DYNA – The Multiphysics Solver using a **one-code strategy**
  - Efficient **multi-physics**
  - **Multi-stage** problems
  - **Massively parallel**



Element-Free Galerkin Method

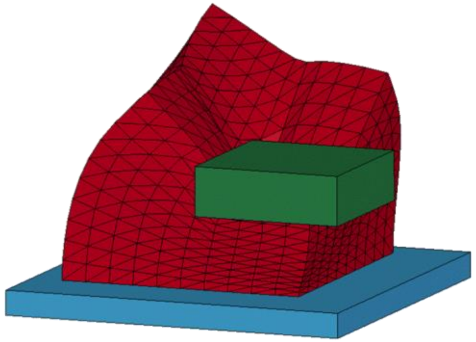
- **Numerical method** are needed **to solve space and time-dependent partial differential equations**, e. g. the momentum balance in structural mechanics

$$\rho \ddot{\mathbf{u}} = \operatorname{div} \mathbf{T}(\mathbf{u})$$

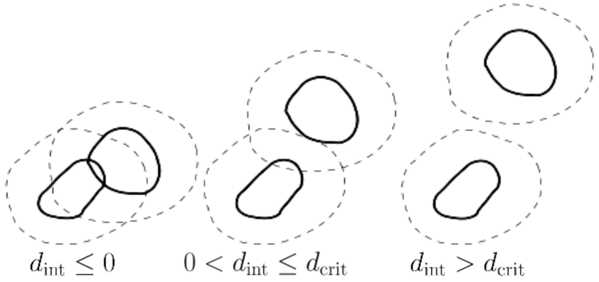
- In general, numerical methods are **based on space- and time-discretization schemes**
  - Space discretization accounts for spatial changes
  - Time discretisation accounts for temporal changes
- Spatial discretisation methods, e. g.,
  - Finite-Element Method (FEM)
  - Smoothed Particle Hydrodynamics (SPH)
  - **Element-Free Galerkin (EFG) Method**
- Time discretization methods, **explicit or implicit**, e. g.
  - Central Difference Method
  - Newmark Scheme

Element-Free Galerkin Method

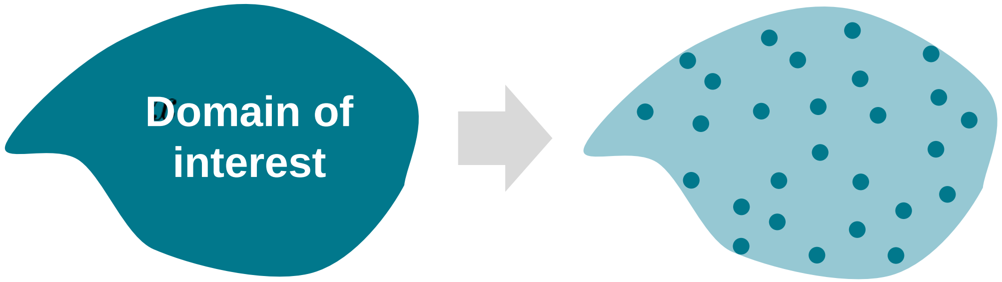
- Finite-Element Analysis (FEA) fails at large element distortions
  → Use of **particle-based (mesh-free) spatial discretization methods for continua**
  - Smoothed Particle Hydrodynamics (SPH)
  - **Element-Free Galerkin (EFG) Method**
  - Smooth Petrov Galerkin (SPG) Method
- EFG is a **continuum- and particle-based spatial discretization method**
- Discrete particle method (DEM) versus continuum-based particle method (CBPM)



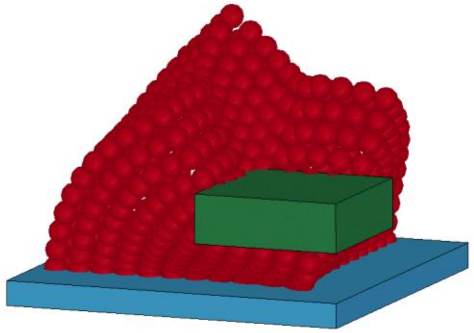
Finite-Element Analysis



In DEM, discrete particles interact via contact laws



In CBPM, particles serve as computational points for field variables inside of a continuum



Element-Free Galerkin

Element-Free Galerkin Method

- Basic EFG concept
  - **Decomposition of the domain of interest into material particles and a support domain** similar to SPH
  - **Polynoms to approximate field variables** similar to shape functions in FEM
  - Usage of **background mesh** to integrate the weak forms

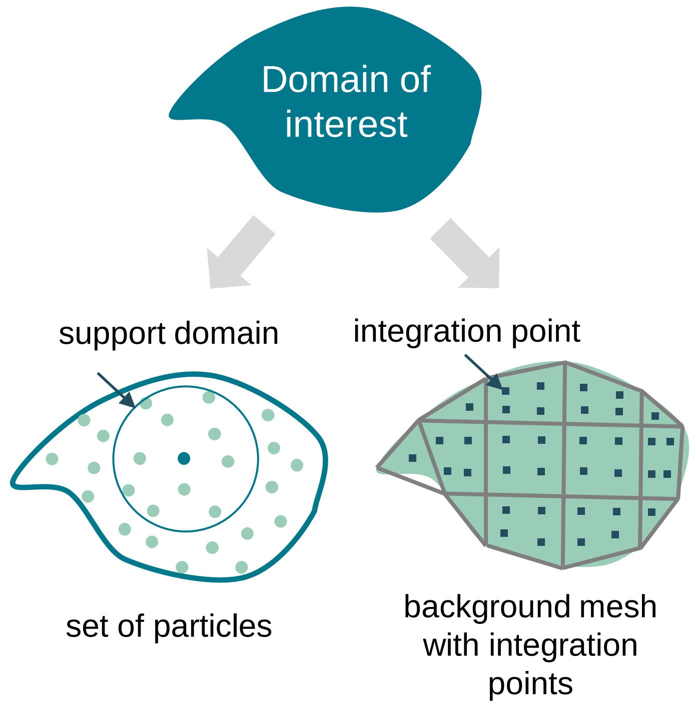
$$\rho\ddot{\mathbf{u}} = \operatorname{div}\mathbf{T}(\mathbf{u}) \quad \Rightarrow \quad \int_{\Omega_e} \rho\ddot{\mathbf{u}}\cdot\delta\mathbf{u}\,\mathrm{d}\Omega + \int_{\Omega_e} \operatorname{grad}\,\delta\mathbf{u}\cdot\mathbf{T}\,\mathrm{d}\Omega - \int_{\partial\Omega_e} \delta\mathbf{u}\cdot\mathbf{t}\,\mathrm{d}a = 0$$

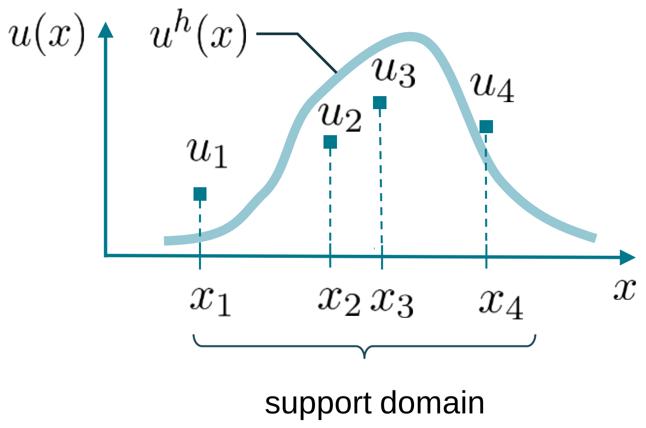
- **Approximating polynoms are constructed from monomials** with their parameters minimized inside support domain through Moving Least Squares (MLS)

$$u^h(\mathbf{x}) = \mathbf{H}^T(\mathbf{x})\,\mathbf{b}(\mathbf{x}) = \sum_{i=1}^{n} H_i(\mathbf{x})b_i(\mathbf{x})$$

with $\mathbf{b}(\mathbf{x})$ coefficient vector

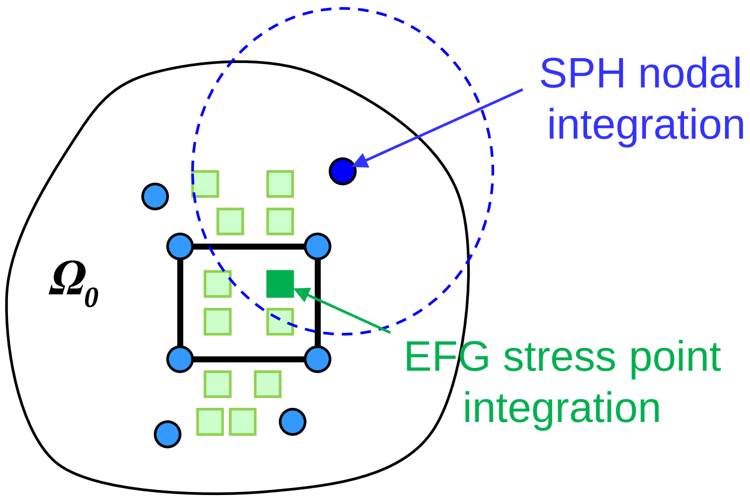
$\mathbf{H}^T(\mathbf{x})$ vector of monomials, i. e. $\mathbf{H}^T(\mathbf{x}) = [1, x, x^2, ..., x^n]^T$



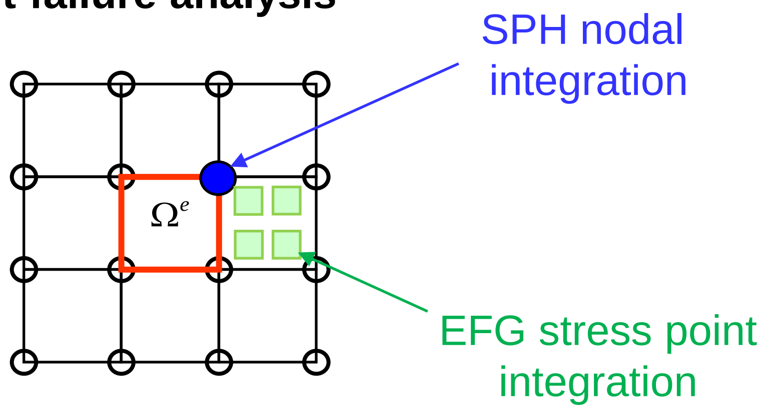


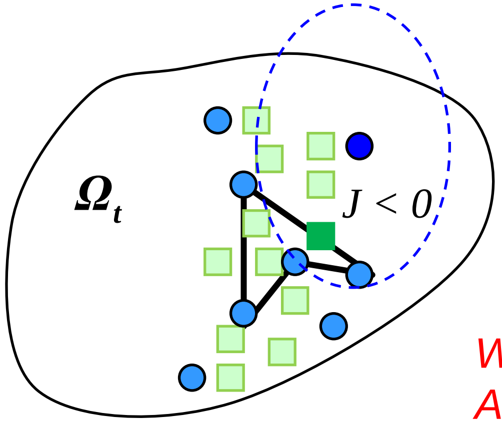
Element-Free Galerkin Method

- **Remarks on the background-mesh integration**
  - Advantages
    - Definition of the **physical domain**
    - Usage of **various contact types**
    - Application of **boundary conditions**
    - Volume integration via “stress points”
  - Disadvantages
    - **Mesh distortion issue**
    - **Difficult failure analysis**



initial configuration



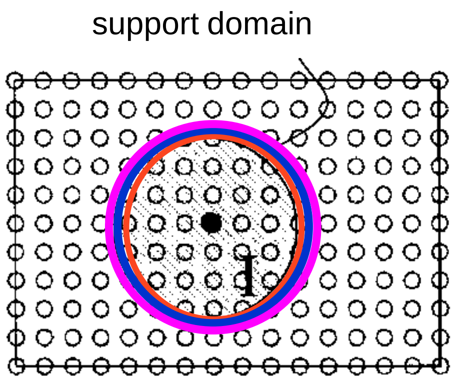


current configuration

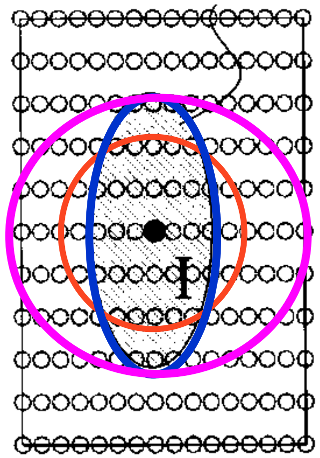
What will be the worst?
As bad as FEM!

Element-Free Galerkin Method

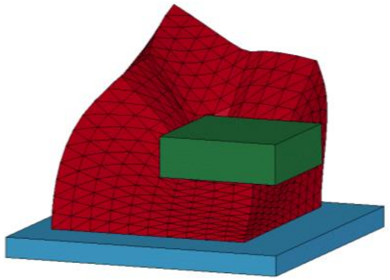
- **Remarks on the support domain**, i. e. EFG Kernel Function
  - Lagrangean Kernel
    - Support is defined in the initial configuration
    - Support covers the same set of material points throughout time
  - Eulerian Kernel
    - Support is defined in the current configuration
    - Support covers different material points throughout time
  - Semi-Lagrangean Kernel
    - Support is defined in the current configuration
    - Support covers the same number of material points throughout time



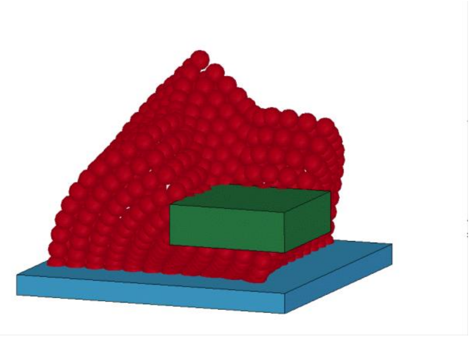
initial configuration



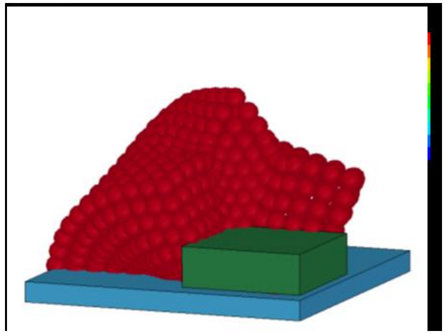
deformed configuration



FEM



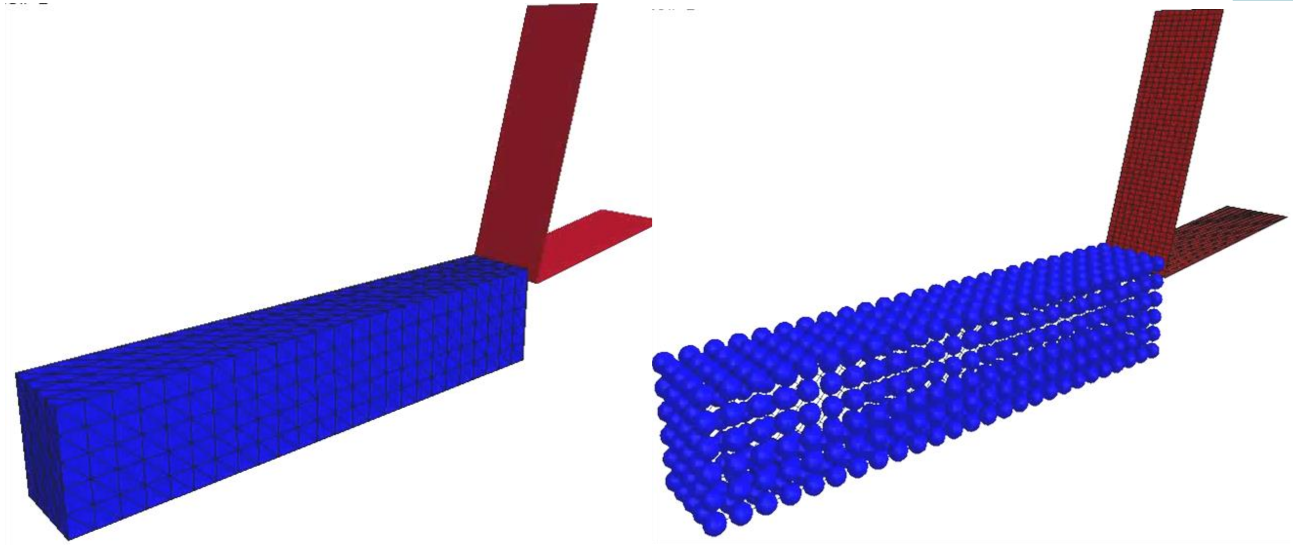
EFG + Lagrangean Kernel



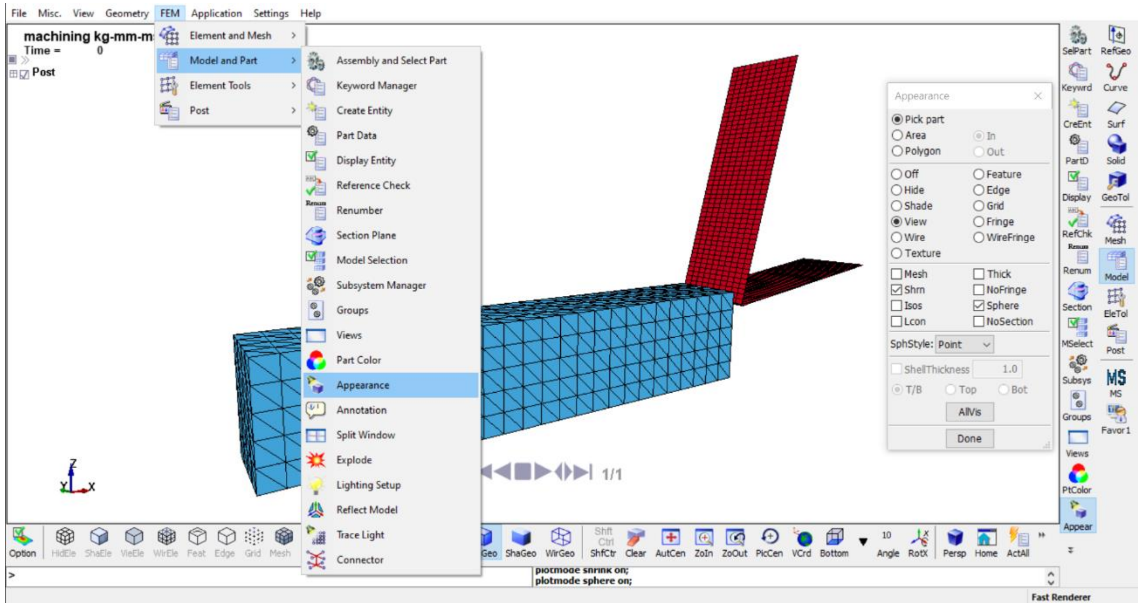
EFG + semi-Lagrangean Kernel

EFG in LS-DYNA

- Implementation features
  - Explicit and implicit time integration
  - Thermal coupling
  - Adaptive and remeshing strategies



- Activating an EFG analysis in LS-DYNA
  - Create your FE model as usual
  - Replace ***SECTION_SOLID** by ***SECTION_SOLID_EFG**
  - LS-PrePost particle representation
    - Go to *FEM* → `Model and Part` → `Appearance`
    - Tick the check boxes `Sphere`" and `Shrn`
    - Click the EFG part
    - Modify particle visualization in `Settings` → `General Settings`



EFG Usage in LS-DYNA

- Keyword ***SECTION_SOLID_EFG**

```
*SECTION_SOLID_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#   secid    elform
         1        42
$#      dx        dy        dz   ispline    idila      iebt      idim    toldef
       1.3       1.3       1.3                            3         2
$#     ips     stime      iken        sf      mid       ibr        ds      ecut
         1
```

fracture and other features

secid   section identifier

elform   element type of background mesh:
EQ.41: EFG solid (TET, HEX, PENT mesh)
EQ.42: EFG solid (TET mesh)

(SMP & MPP, implicit and explicit analysis and thermal-mechanical coupling available from R5 on upwards)

EFG Usage in LS-DYNA

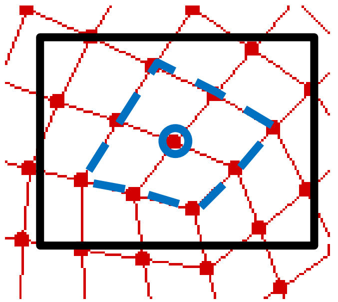
- Keyword ***SECTION_SOLID_EFG**

```
*SECTION_SOLID_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#    secid    elform
          1        42
$#       dx        dy        dz   ispline     idila      iebt      idim    toldef
        1.3       1.3       1.3                             3         2
$#      ips     stime      iken        sf       mid       ibr        ds      ecut
          1
```

fracture and other features

`dx,dy,dz` normalised dilatation parameter in x, y and z direction, respectively, where 1.0 < `dx, dy, dz` < 1.6 are recommended → **the higher the slower the performance**

	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



EFG Usage in LS-DYNA

- Keyword ***SECTION_SOLID_EFG**

```
*SECTION_SOLID_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#   secid    elform
         1        42
$#      dx        dy        dz   ispline     idila      iebt      idim    toldef
       1.3       1.3       1.3                            3         2
$#     ips     stime      iken        sf       mid       ibr        ds      ecut
         1
```

fracture and other features

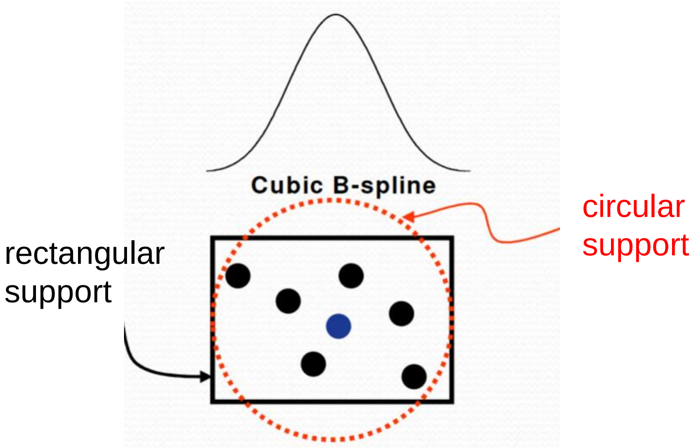
`ispline` choice of kernel function

EQ.0: Cubic spline function (rectangular support dx, dy, dz)
with linear base function (default)

EQ.1: Quadratic spline function (rectangular support with dx, dy, dz)

EQ.2: Cubic spline function (circular support),
where dx defines radius coefficient

EQ.10: Cubic spline function with bilinear base function
(rectangular support with dx, dy, dz)



EFG Usage in LS-DYNA

- Keyword ***SECTION_SOLID_EFG**

```
*SECTION_SOLID_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#    secid    elform
          1        42
$#       dx        dy        dz   ispline    idila      iebt      idim    toldef
        1.3       1.3       1.3                            3         2
$#      ips     stime      iken        sf       mid       ibr        ds      ecut
          1
```

fracture and other features

idim — Integration of the weak forms within the spatial domain

EQ. 1: Local boundary condition method

EQ. 2: Gauss integration (default)

EQ.-1: Stabilized EFG method (PENT and HEX background mesh only)

- One-point integration scheme + gradient type hourglass control
- Designed especially for foam and rubber materials
- Computational cost is between reduced integration FEM and full integration FEM

EQ.-2: Fractured EFG method (TET background mesh and SMP only)

DYNA MORE

EFG Usage in LS-DYNA

- Keyword ***SECTION_SOLID_EFG**

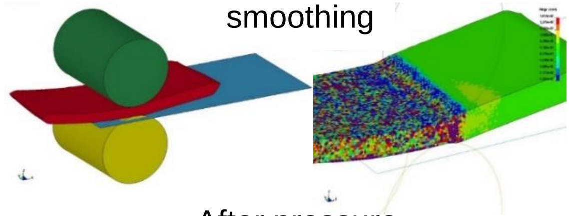
```
*SECTION_SOLID_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#    secid    elform
          1        42
$#       dx        dy        dz   ispline     idila      iebt      idim    toldef
        1.3       1.3       1.3                             3         2
$#      ips     stime      iken        sf       mid       ibr        ds      ecut
          1
```

fracture and other features

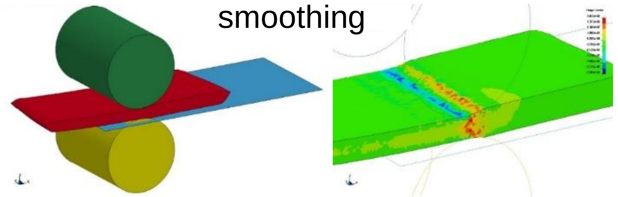
toldef — Deformation tolerance for the activation of adaptive EFG Semi-Lagrangian and Eulerian kernel

ips — Apply pressure smoothing by adding slight compressibility to the material
EQ. 0: No pressure smoothing (default)
EQ. 1: Moving-least squared pressure recovery (only for ELFFORM=42)
NOT recommended if
- there is numerical instability in contact (“hard contact”)
- Implicit analysis has convergence problem

Before pressure smoothing



After pressure smoothing



EFG Usage in LS-DYNA

- Mesh-adaptivity/Remeshing in LS-DYNA
  - Basically follows the remeshing keywords as in FEA (check out our forming class for more informations)
  - However, with some extensions merely regarding EFG

```
*Part
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#     pid       secid         mid     eosid      hgid      grav    adpopt      tmid
         1           1           1                                       2
```

adpopt — activate remeshing

- LT.0: adaptive remeshing for 2-D solids, adpopt gives the load curve ID that defines the element size as a function of time
- EQ.0: adaptive remeshing is inactive for this part ID (default)
- EQ.1: h-adaptive for 3-D shells.
- EQ.2: adaptive remeshing for 2-D solids, 3-D tetrahedrons and 3-D EFG
- EQ.3: axisymmetric r-adaptive remeshing for 3-D solid
- EQ.9: passive h-adaptive for 3-D shells. The elements in this part will not be split unless their neighboring elements in other parts need to be split more than one level.

EFG Usage in LS-DYNA

- Mesh-adaptivity/Remeshing in LS-DYNA
  - Basic remeshing parameters apply from FEA also apply to EFG

```
*CONTROL_ADAPTIVE
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$# adpfreq    adptol    adpopt    maxlvl    tbirth    tdeath     lcadp    ioflag
      0.01
$# adpsize    adpass    ireflg    adpene     adpth    memory    orient     maxel
                                       1
```

adpfreq	Time interval in which LS-DYNA checks the remeshing criteria, e.g. adpene
lcadp	Time interval between remeshing criteria-checks over time via *DEFINE_CURVE (overwrites adpfreq)
adpene	For shells (h-adaptive): GT.0: adaptivity takes place when the **forming contact surfaces are approaching** LT.0: adaptivity takes place when the **forming contact surfaces are penetrating** The refinement generally occurs before contact takes place and the refinement is based on the curvature of the tooling For 3-D solids (r-adaptive): GT.0: the mesh refinement is based on the curvature of the tooling

EFG Usage in LS-DYNA

- Mesh-adaptivity/Remeshing in LS-DYNA
  - Similar to ***CONTROL_REMESHING** for FEA, however, with the additional card 2 and 3 for EFG exclusively

```
*CONTROL_REMESHING_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#    rmin      rmax   vf_loss     mfrac    dt_min     icurv       cid    segang
      0.01       0.1
$#     ivt       iat      iaat       ier        mm
                                                 1
$#    iat1      iat2      iat3
```

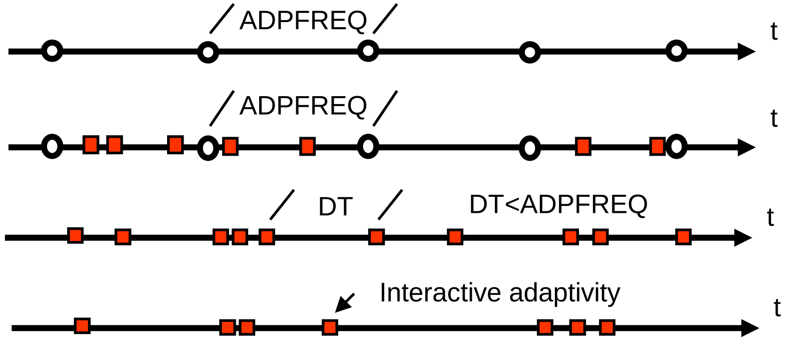
rmin/rmax	Minimum/Maximum edge length for the surface mesh surrounding the parts which should be remeshed
ivt	Internal variable transfer in adaptive EFG EQ.1: Moving Least square approximation with Kronecker-delta property (recommended in general case) EQ.-1: Moving Least square approximation without Kronecker-delta property EQ.2: Partition of unity approximation with Kronecker-delta property EQ.-2: Partition of unity approximation without Kronecker-delta property EQ.-3: Finite element approximation

EFG Usage in LS-DYNA

- Mesh-adaptivity/Remeshing in LS-DYNA
  - Additional to predefined adaptivity steps (see `adpfreq` in **`*CONTROL_ADAPTIVE`**), an interactive remeshing can be defined as well using additional criteria, i. e. `iat1`, `iat2` and `iat3`

```
*CONTROL_REMESHING_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#      rmin      rmax   vf_loss     mfrac    dt_min     icurv       cid    segang
        0.01       0.1
$#       ivt       iat      iaat       ier        mm
                                                   1
$#      iat1      iat2      iat3
```

`iat`	Flag for interactive adaptivity	
	EQ.0:	no interactive adaptivity (default)
	EQ.1:	additional interactive adaptivity in between predefined adaptivity steps
	EQ.2:	purely interactive adaptivity, however, remeshing time interval is bounded by `adpfreq`
	EQ.3:	purely interactive adaptivity.



EFG Usage in LS-DYNA

- Mesh-adaptivity/Remeshing in LS-DYNA

```
*CONTROL_REMESHING_EFG
$---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
$#     rmin      rmax   vf_loss     mfrac    dt_min     icurv       cid    segang
       0.01       0.1
$#      ivt       iat      iaat       ier        mm
                                                  1
$#     iat1      iat2      iat3
```

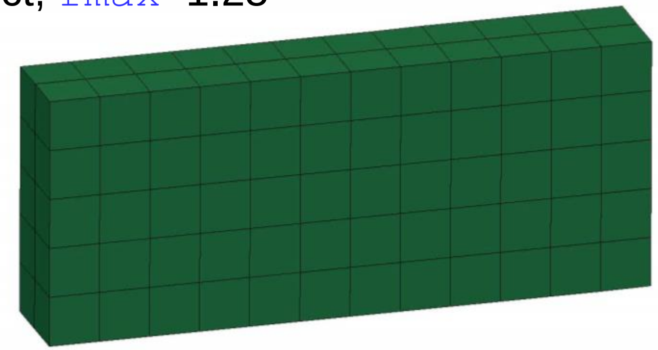
`mm` Interactive adaptive remeshing with monotonic resizing
EQ.1: The adaptive remeshing cannot coarsen a mesh
the current implementation only supports `iat` = 1, 2, 3 and `ier` = 0

`iat<1,2,3>` Tolerance for interactive adaptivity for ELFORM=42, e. g.
- `iat1` shear strain
- `iat2` max.-to-min. element edge ratio
- `iat3` volume-change tolerance

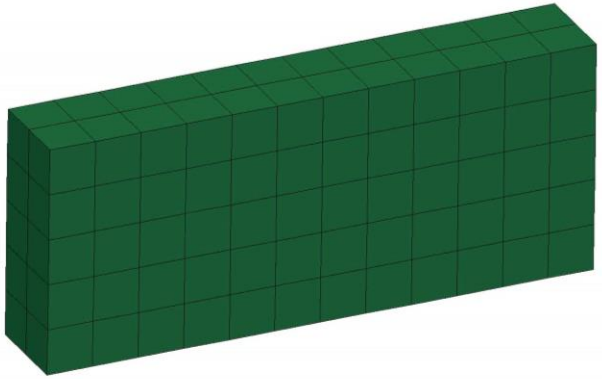
EFG Usage in LS-DYNA

- Mesh-adaptivity/Remeshing in LS-DYNA

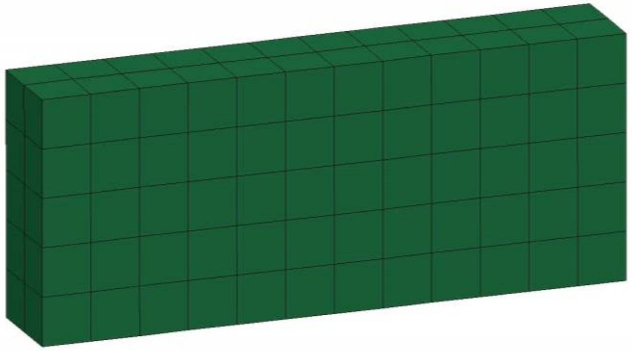
adpene=0, mm=0
rmin has no effect, rmax=1.25



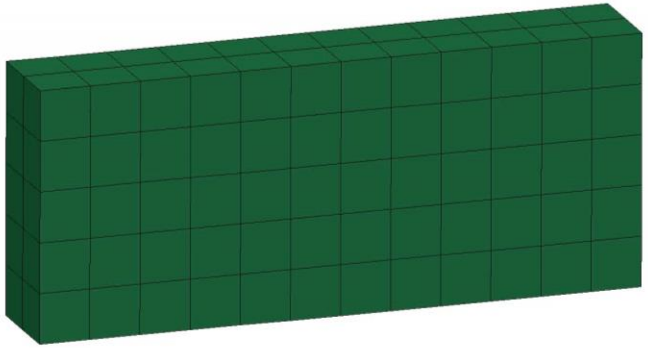
adpene=1, mm=0
rmin=0.15, rmax=2.25



adpene=0, mm=1
rmin has no effect, rmax=1.25

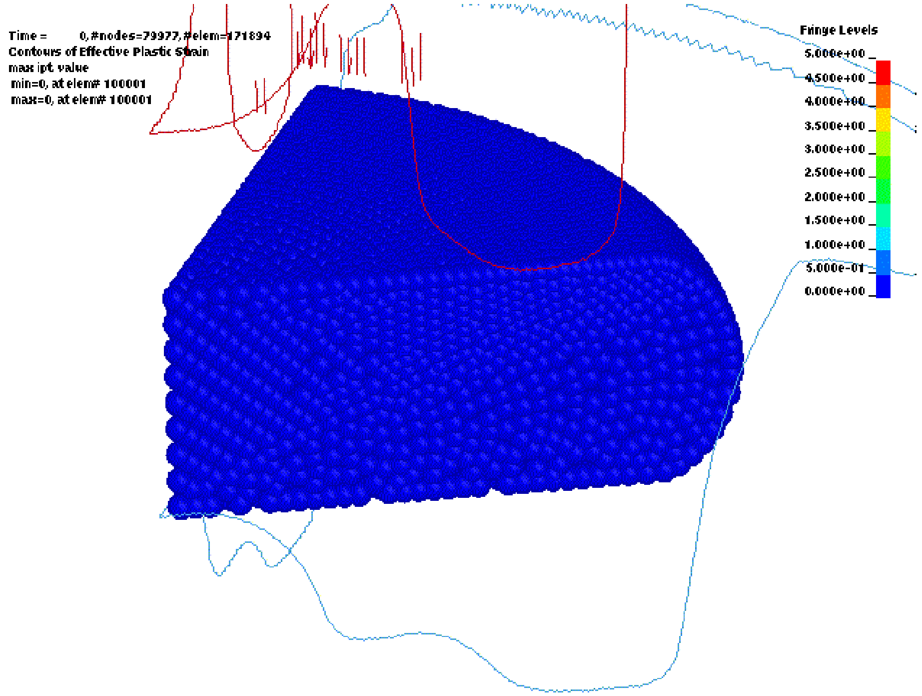


adpene=1, mm=1
rmin=0.15, rmax=2.25

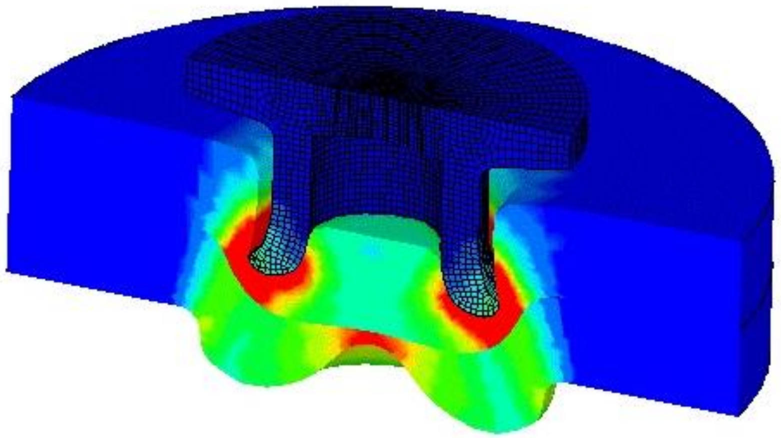


Application examples

- Overview



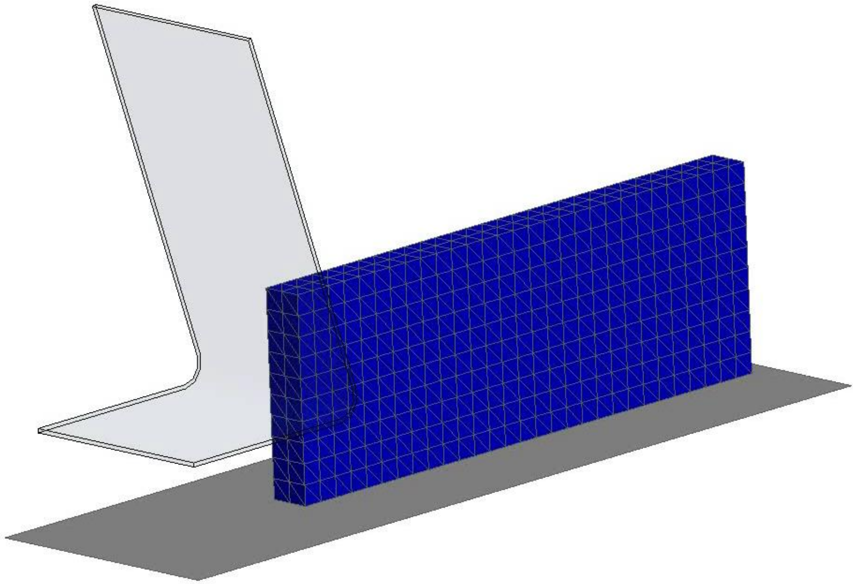
Bulk forming



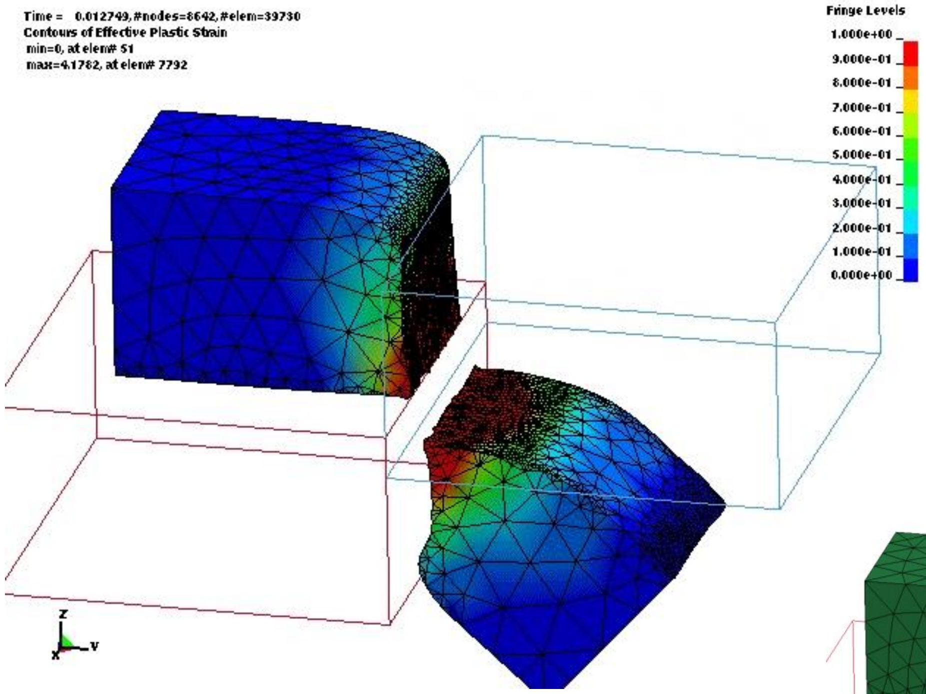
Self-piercing rivet

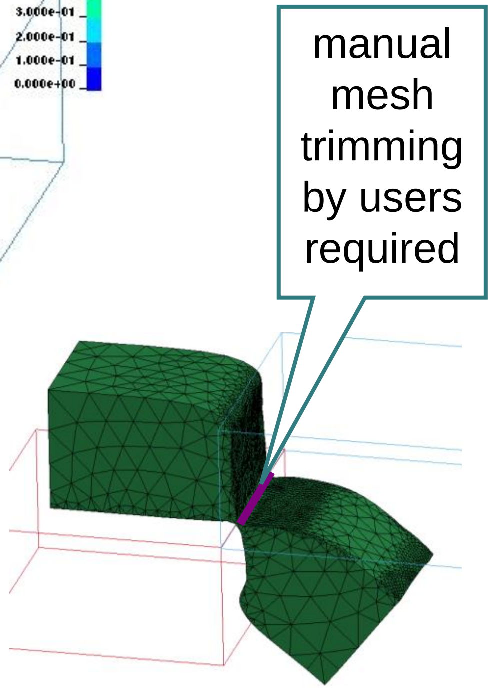
Application examples

- Overview



Cutting simulation
(without manual trimming)

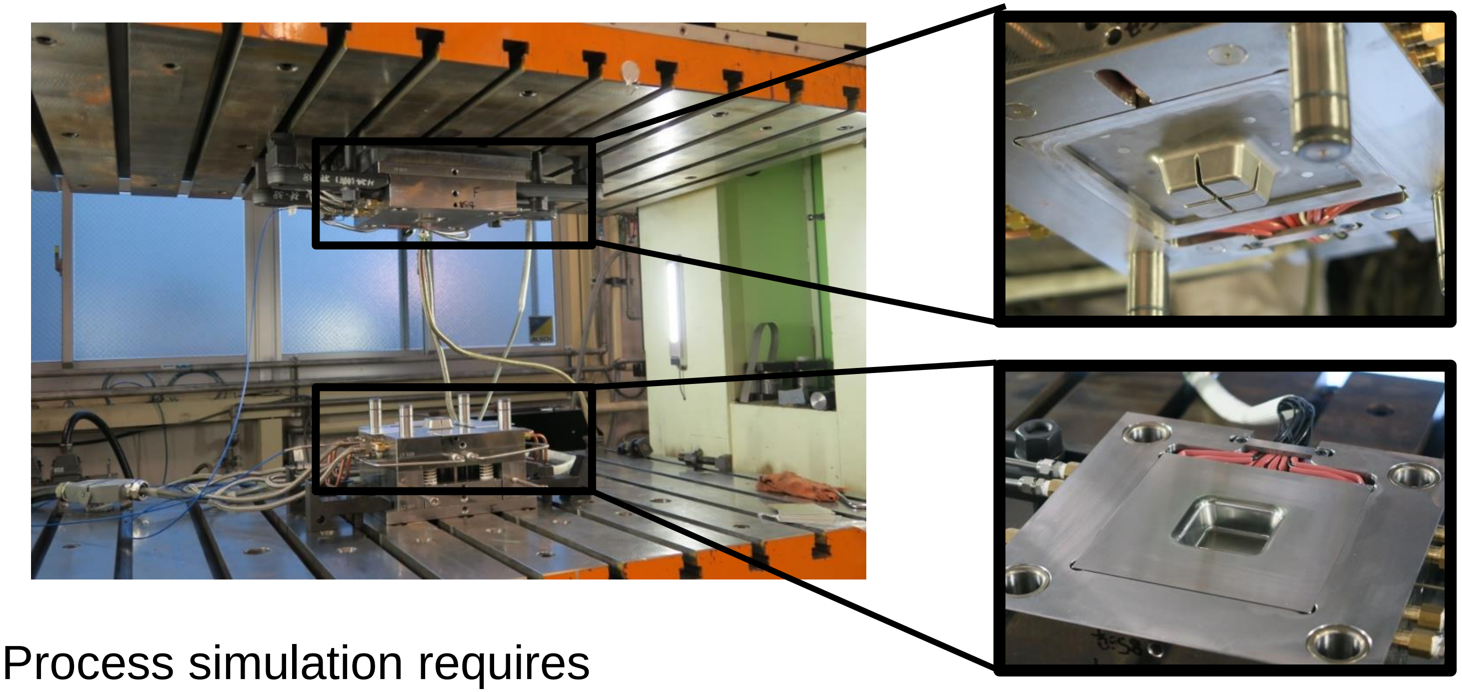


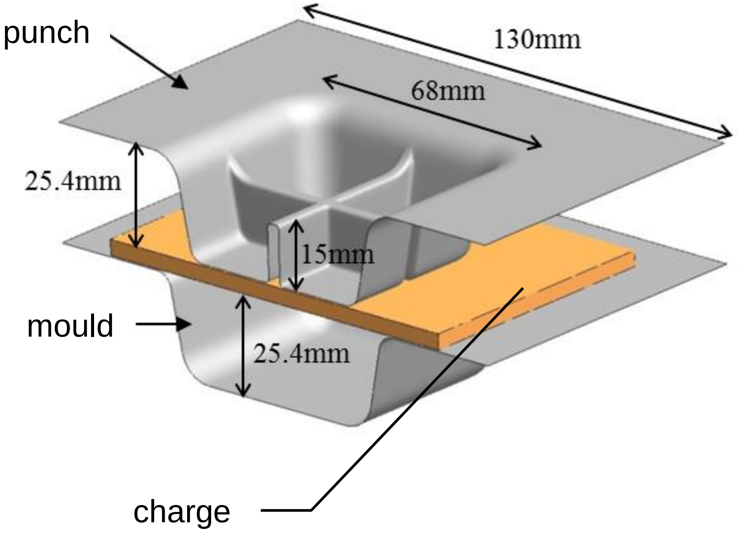


Cutting simulation
(with manual trimming)

Application examples

- Modelling of a sheet-molding compound (with courtesy of JSOL)

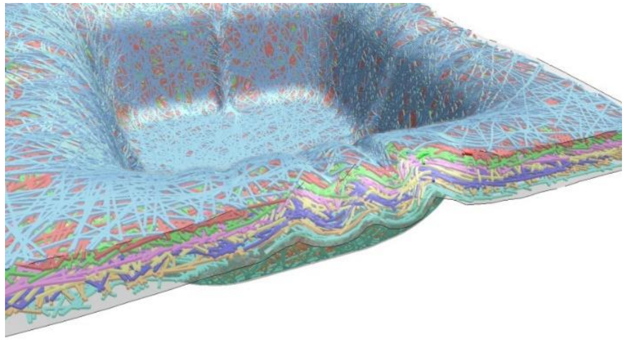
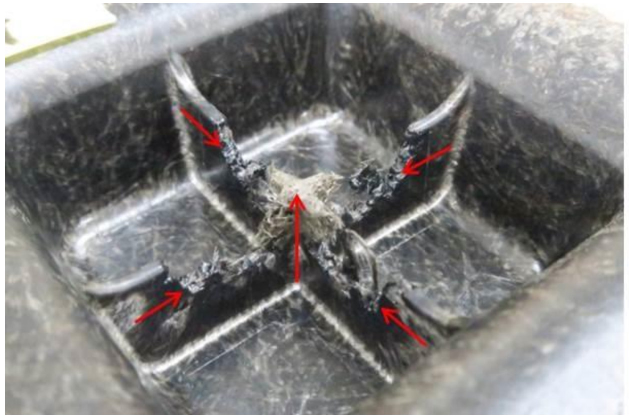
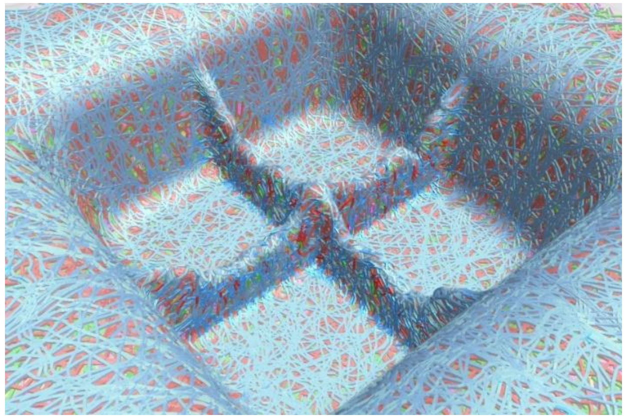




- Process simulation requires
  - Remeshing
  - Thermal Coupling
  - Beam-Fiber couplling

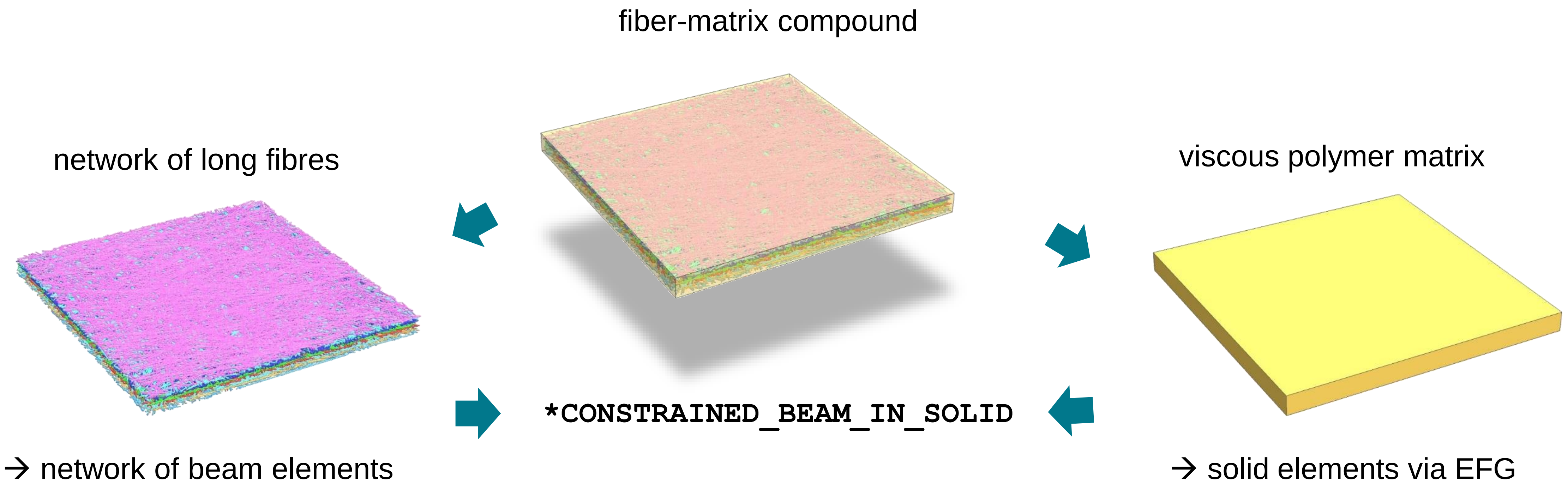
Application example

- Modelling of a sheet-molding compound (with courtesy of JSOL)

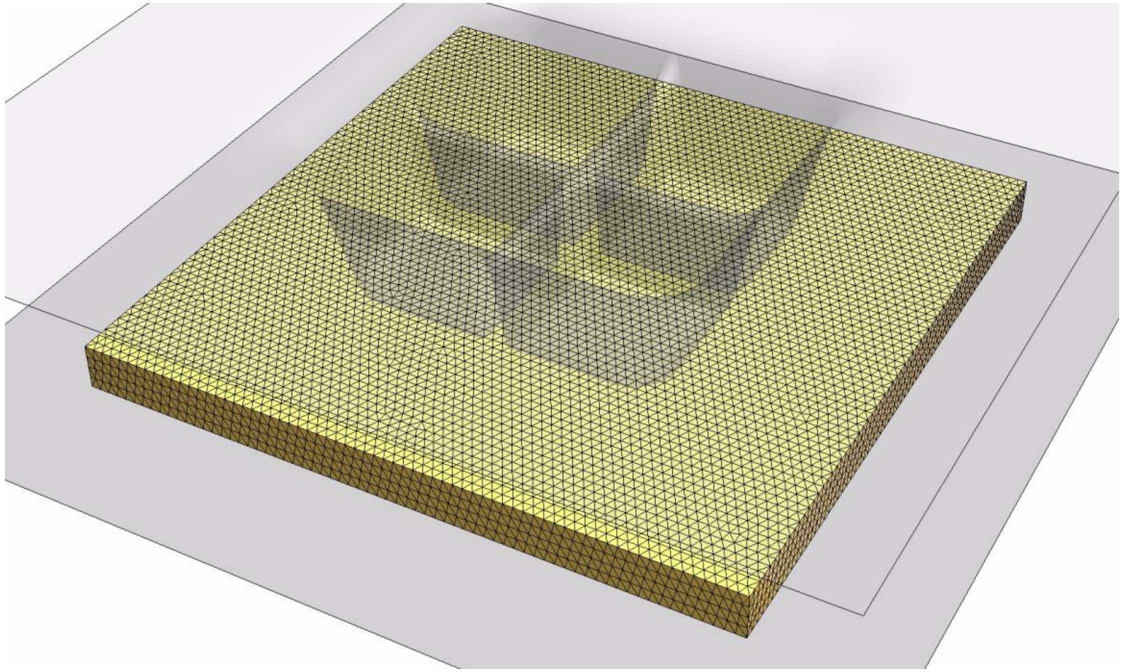
Application example

- Modelling of a sheet-molding compound (with courtesy of JSOL)
  - Large deformations of the fiber-matrix compound via Element-Free Galerkin (EFG)
  - Fiber-matrix interaction via **`*CONSTRAINED_BEAM_IN_SOLID`** (CBIS)
  - Fiber-fibre interaction (fibre network) through **`*CONTACT_AUTOMATIC_GENERAL`**

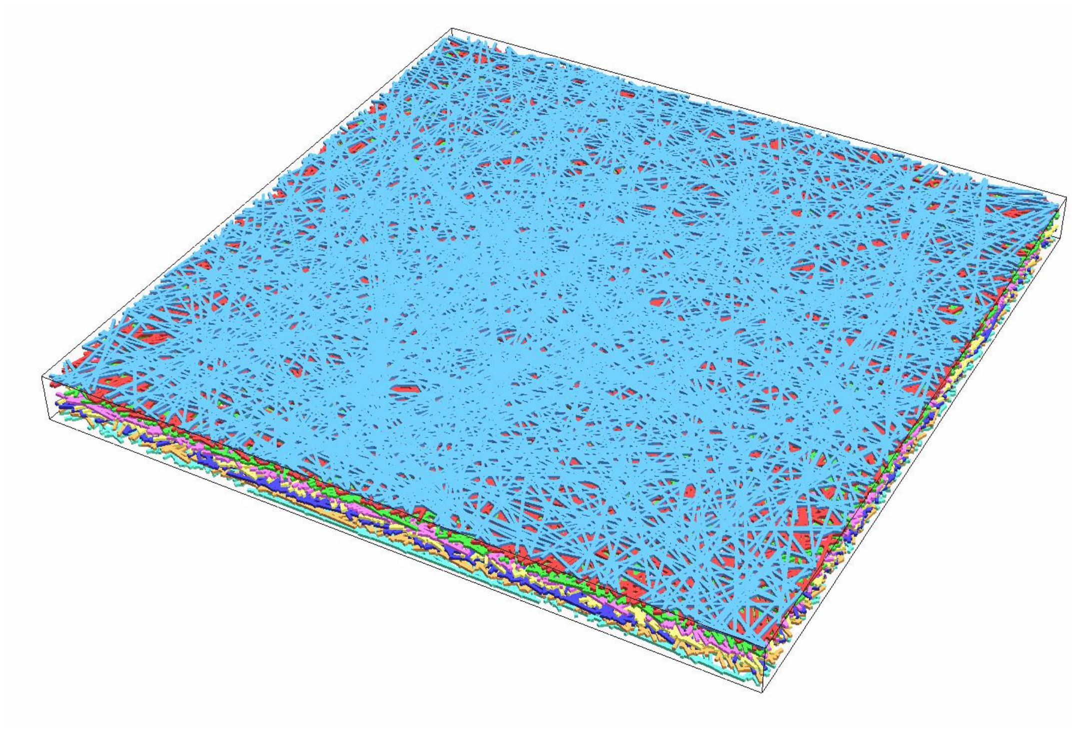


Application example

- Modelling of a sheet-molding compound (with courtesy of JSOL)



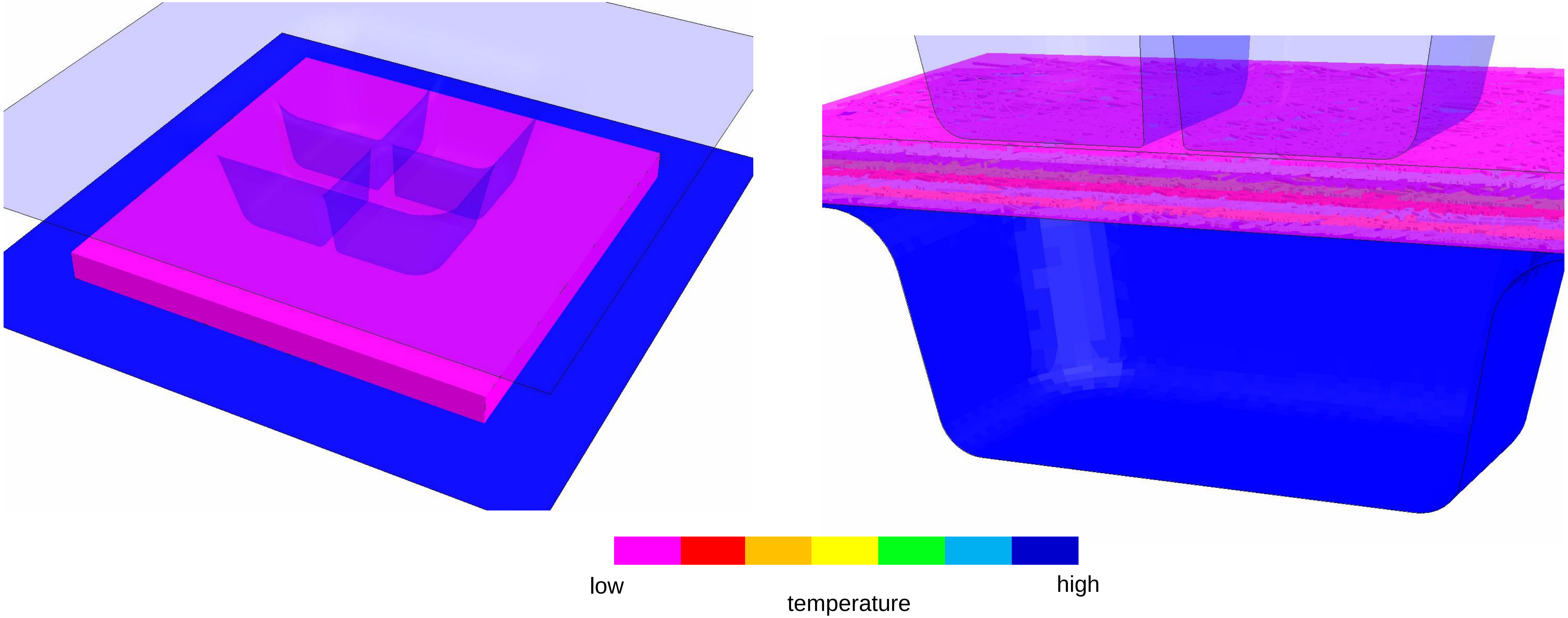
Matrix compound



Fiber network

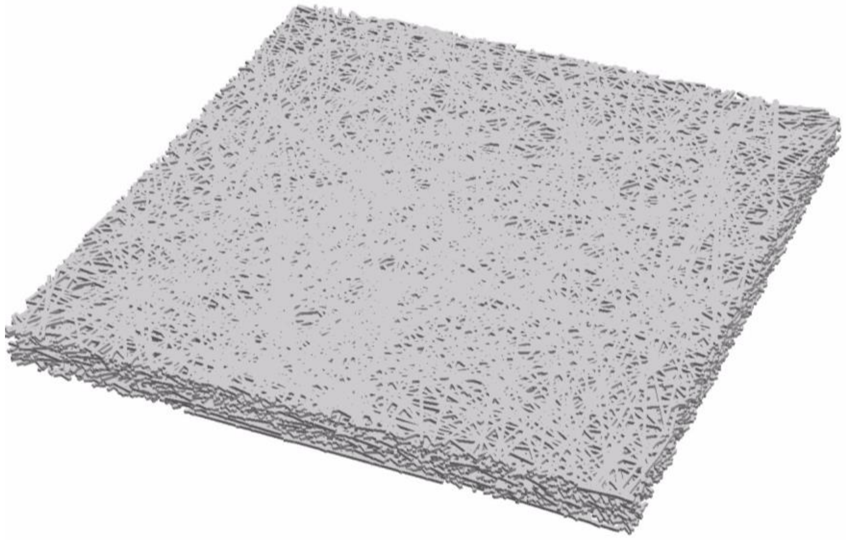
Application example

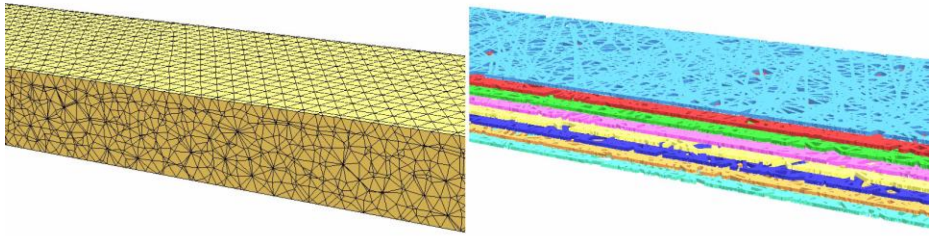
- Modelling of a sheet-molding compound



- Summary
  - Available for **explicit and implicit time-integration schemes**
  - EFG more **suitable for large deformations** compared to FEA, e. g.
    - Forging
    - Cutting
  - **Remeshing** extents an even wider range of application but
    - Geometry is slightly changed
    - Contact force might be reduced
    - Part of the solution is lost due to variable transfer

    → Trigger as few remeshing steps as possible!
  - **Thermal coupling** possible
  - However, in general, EFG is
    more computational expensive than FEA

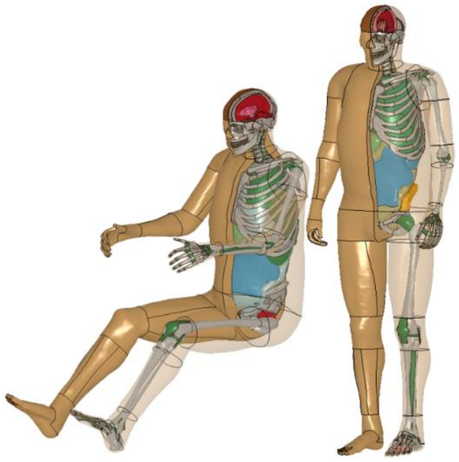




Thank you for your attention!

- LS-DYNA
  - Support / Tutorials / Examples / FAQ
    www.dynasupport.com
  - More Examples
    www.dynaexamples.com
  - Conference Papers
    www.dynalook.com
- LS-PrePost
  - Support / Tutorials / Download
    www.lstc.com/lspp
- LS-OPT
  - Support / Tutorials / Examples
    www.lsoptsupport.com

- Livermore Software Technology Corp. (LSTC)
  www.lstc.com
- DYNAmore GmbH
  - Website
    www.dynamore.de
  - Social Media
    https://twitter.com/LSTCandDYNAmore
    www.youtube.com/c/DYNAmoreGmbH



[THUMS® www.dynamore.de]

DYNA MORE

Your LS-DYNA distributor and more