University of Glasgow

MoFEM: An open source, parallel finite element library – MFront integration

Karol Lewandowski & Łukasz Kaczmarczyk,
Andrei Shvarts, Ignatios Athanasiadis, Mebratu Wakeni, Hoang Nguyen, Daniele Barbera, Christophe-Alexandre Chalons-Mouriesse, Adriana Kulikova & Chris Pearce

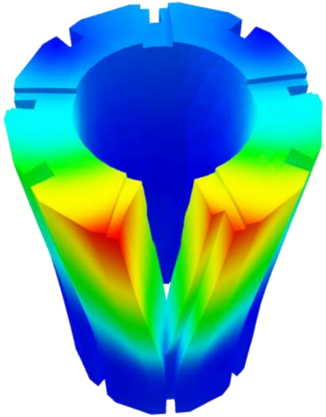
WORLD CHANGING GLASGOW

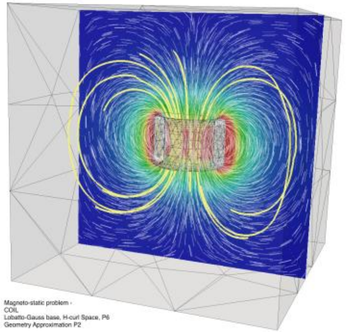
MoFEM in a nutshell

mofem.eng.gla.ac.

Mesh-oriented Finite Element Method:

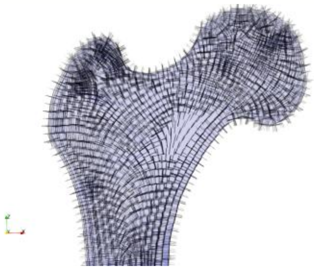
- C++ open-source library (permitting private user modules)
- Mixing different element types and shapes in one mesh
- Using hierarchical basis functions (*hp*-adaptivity)
- Support for L2, H-div, H-curl, and H1 approximation spaces
- Supporting multi-field and multi-physics problems
- Native support for parallelisation
- Chained with state-of-the art libraries for managing Topology (MOAB) and Linear Algebra (PETSc) – developed at the Argonne National Laboratory
- Validated and currently used by EDF Energy and Jacobs

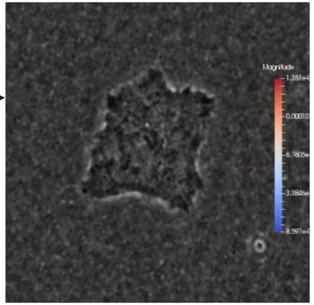


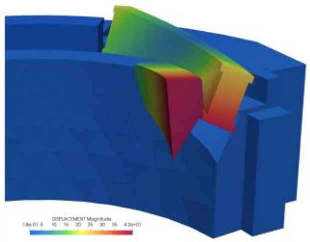


Design objectives

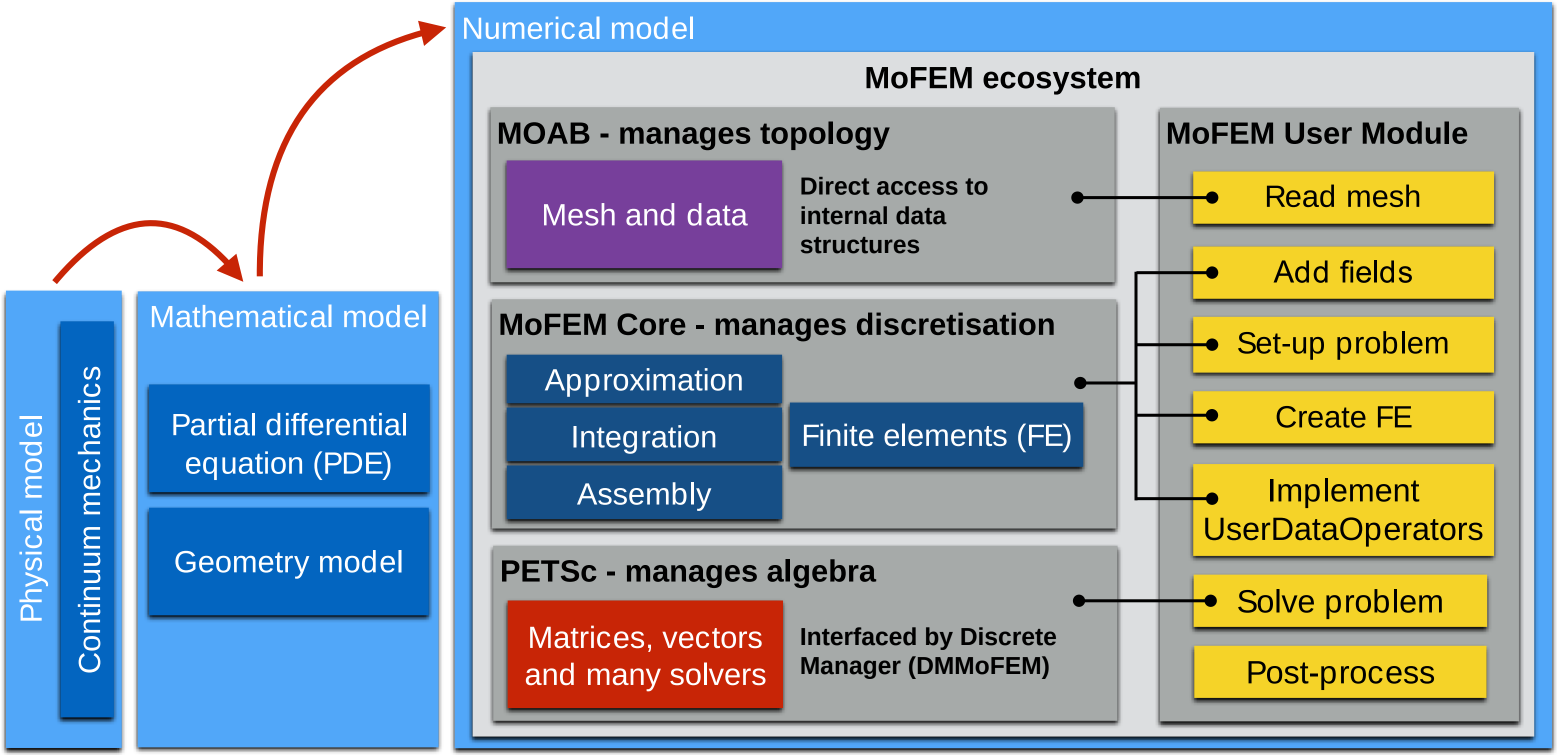
- Design for research & development
- Solve practical engineering problems
- Allow for rapid development for nonstandard niche problems
- Create competitive advantage for industrial/research partner
- Enabling to break complex problem into small simple parts
- Enabling collaborative work distributed in time and space
- Structure allowing testing small parts separately
- Handling of error and unified logging



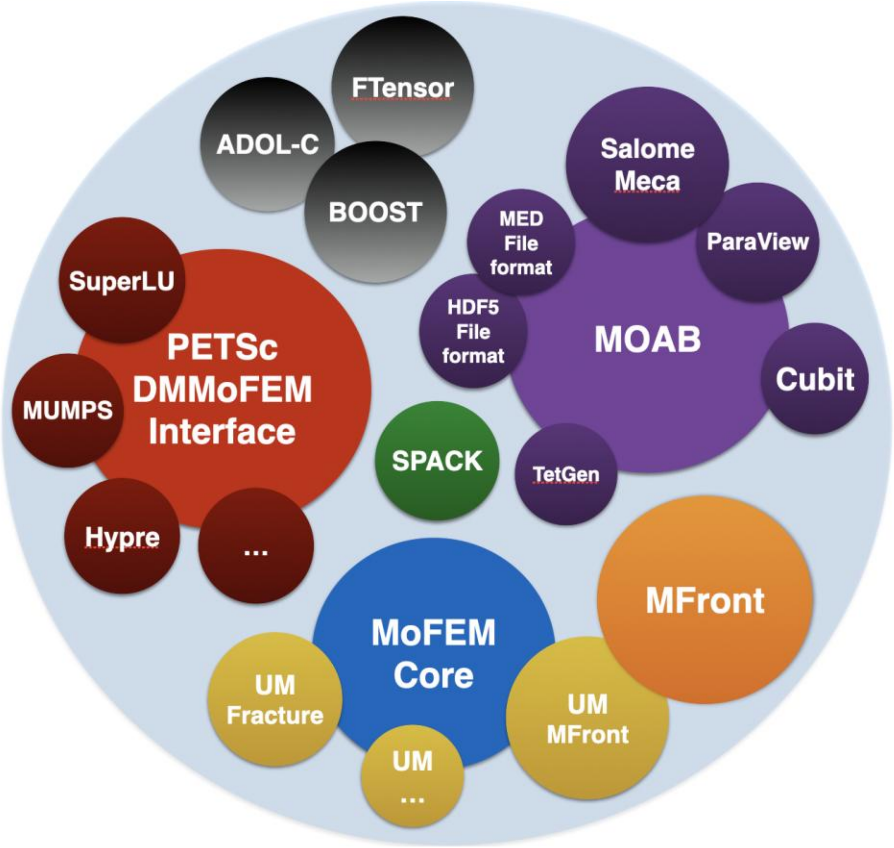


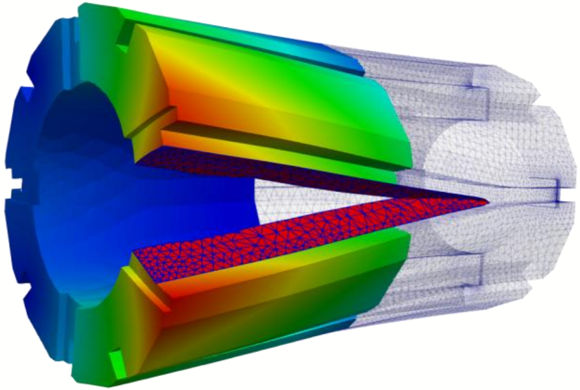


Basic design of MoFEM

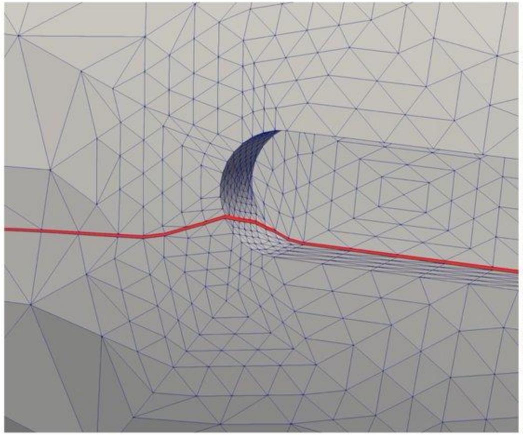


MoFEM ecosystem



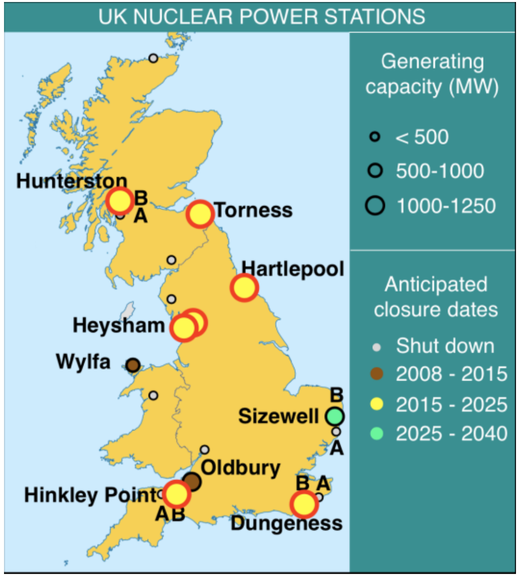


MoFEM is used by EDF/Jacobs Energy to assess structural integrity of nuclear reactors in the UK.

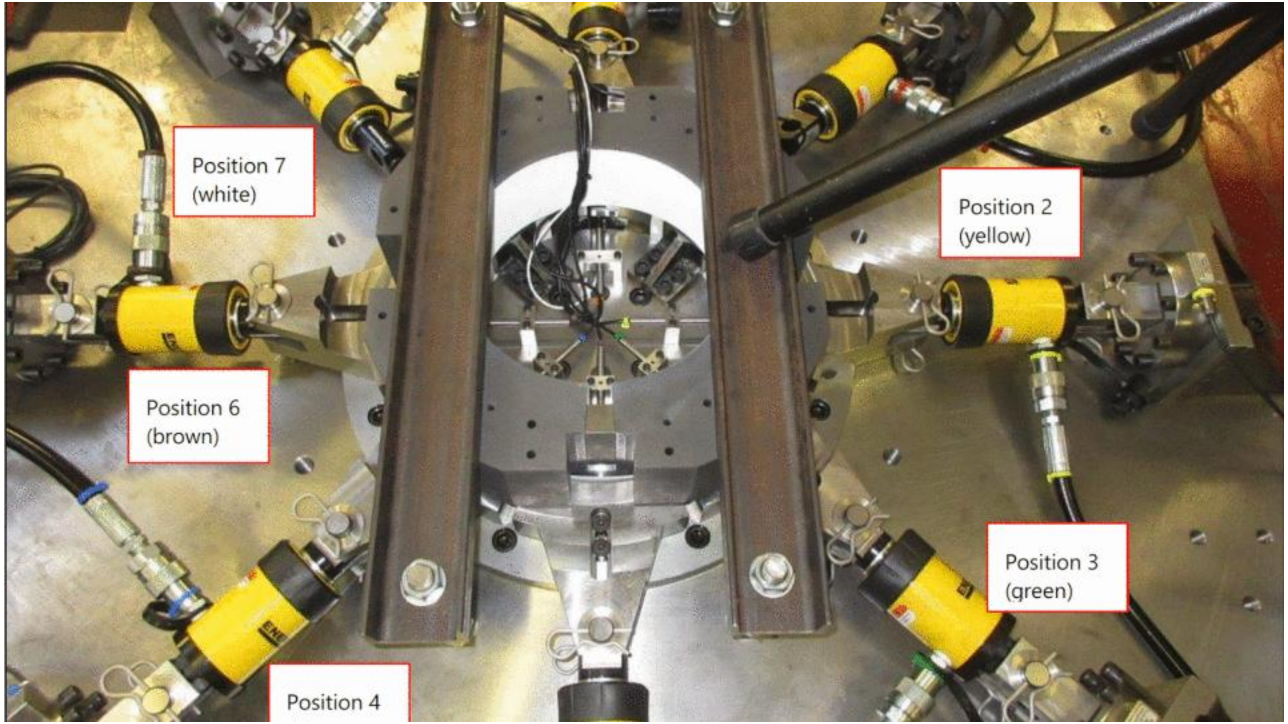


Simulation

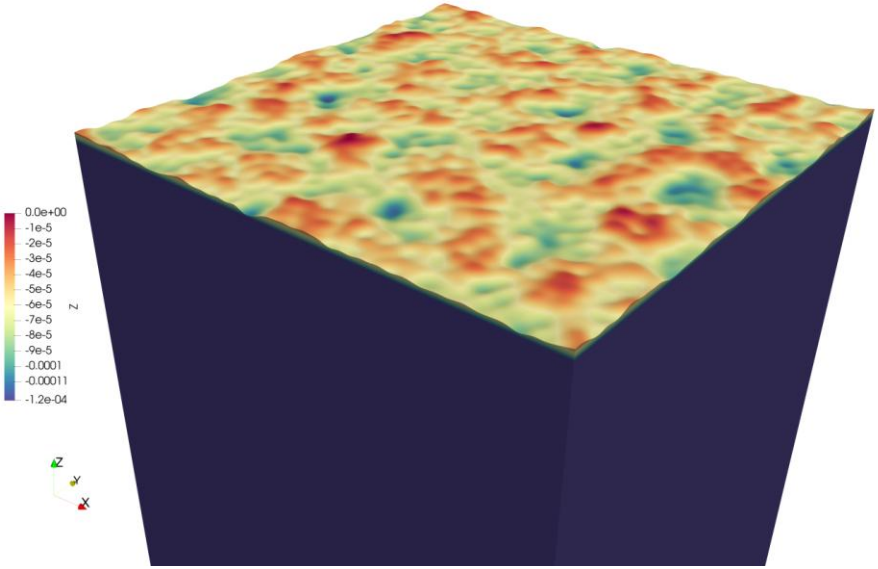
Experiment



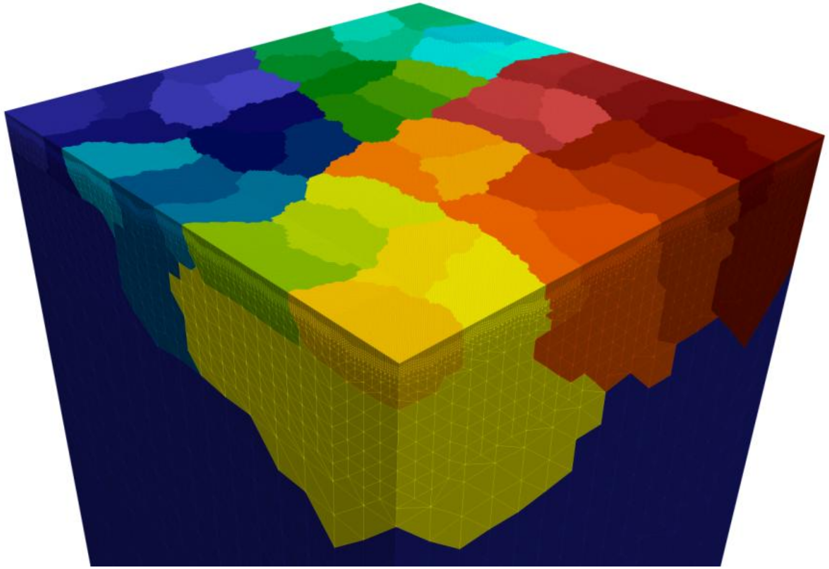
Comparing rig test and simulation



Rough surface contact: FE mesh

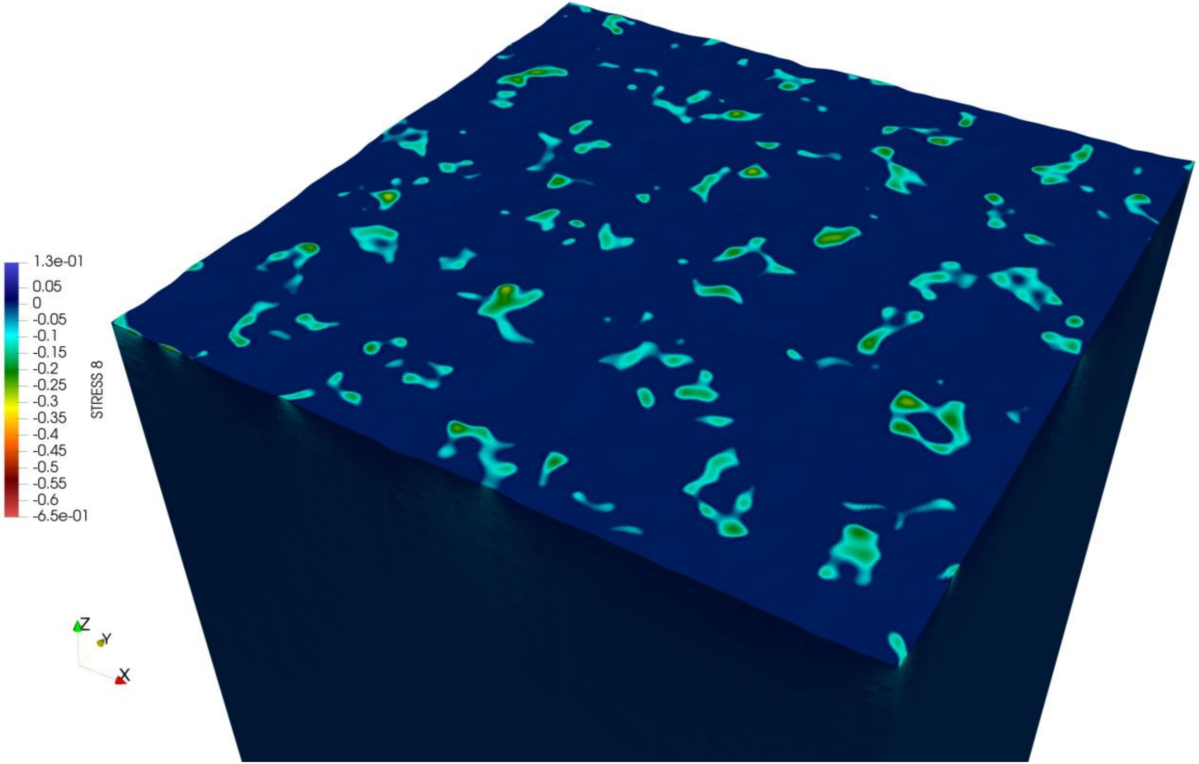


Surface height projected on the mesh

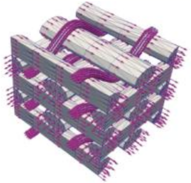


Mesh partitioned into 48 parts

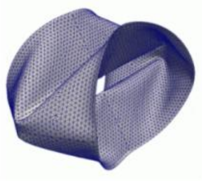
Rough surface contact: normal traction



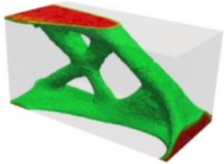
Examples of user modules



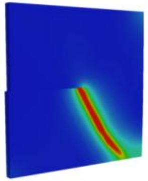
Homogenisation



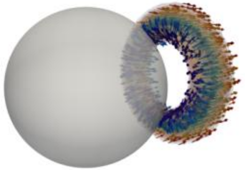
Solid shell



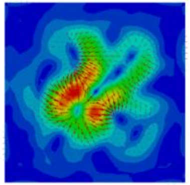
Topology optimisation



Phase field fracture

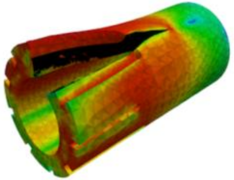


Navier-Stokes flow

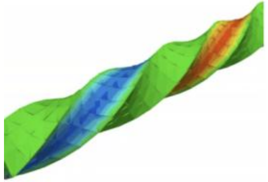


Cell traction microscopy

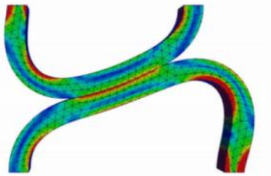
Bone remodelling



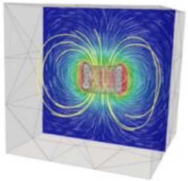
Configurational fracture



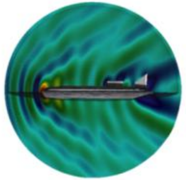
Plasticity



Mortar contact



Magnetostatics

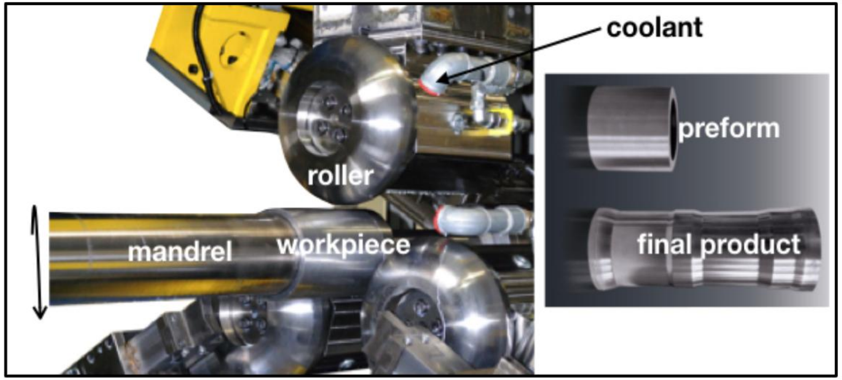


Acoustic wave propagation

Multifield plasticity

$$
\begin{cases}
\dfrac{\partial \sigma_{ij}}{\partial x_j} - b_i = 0 & \forall x \in \Omega \\
\varepsilon_{ij} = \frac{1}{2}\left(\dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i}\right) & \\
\sigma_{ij} = D_{ijkl}\left(\varepsilon_{kl} - \varepsilon^p_{kl}\right) & \\
\dot{\varepsilon}^p_{kl} - \dot{\tau}\left(\left.\dfrac{\partial f}{\partial \sigma_{kl}}\right|_{(\sigma,\tau)}\right) = 0 & \\
f(\sigma,\tau) \leq 0,\ \dot{\tau} \geq 0,\ \dot{\tau} f(\sigma,\tau) = 0 & \\
u_i = \bar{u}_i & \forall x \in \partial\Omega_u \\
\sigma_{ij} n_j = \bar{t}_i & \forall x \in \partial\Omega_\sigma \\
\Omega_u \cup \Omega_\sigma = \Omega & \\
\Omega_u \cap \Omega_\sigma = \emptyset &
\end{cases}
$$

Incremental cold flow forming



$$
\begin{cases}
\left(\dfrac{\partial \delta u_i}{\partial x_j}, \sigma_{ij}\right)_\Omega - (\delta u_i, b_i)_\Omega - (\delta u_i, \bar{t}_i)_{\partial\Omega_\sigma} = 0 \\
\left(\delta\varepsilon^p_{kl}, D_{ijkl}\left(\dot{\varepsilon}^p_{kl} - \dot{\tau} A_{kl}\right)\right) = 0 \\
\left(\delta\tau, c_n\dot{\tau} - \frac{1}{2}\left\{c_n\dot{\tau} + \left(f(\boldsymbol{\sigma},\tau) - \sigma_y\right) + \left\| c_n\dot{\tau} + \left(f(\boldsymbol{\sigma},\tau) - \sigma_y\right)\right\|\right\}\right) = 0
\end{cases}
$$

$$
\begin{aligned}
&\forall \delta u_i \in H^1(\Omega) \\
&\forall \delta\varepsilon^p_{ij} \in L^2(\Omega) \cap \mathcal{S} \\
&\forall \delta\tau \in L^2(\Omega)
\end{aligned}
$$

McBride, Pearce, Steinmann, Barbera

University of Glasgow

Adding fields

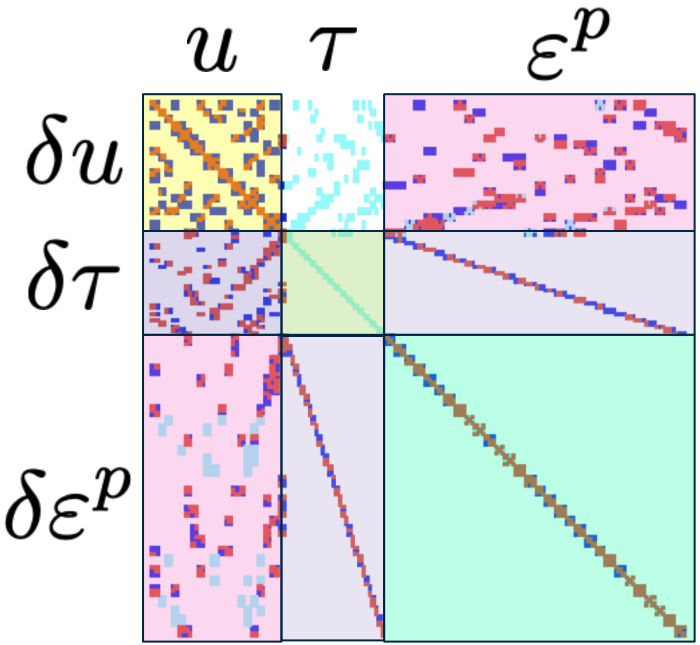
```
//! [Set up problem]
MoFEMErrorCode Example::setupProblem() {
  MoFEMFunctionBegin;
  Simple *simple = mField.getInterface<Simple>();
  // Add field
  CHKERR simple->addDomainField("U", H1, AINSWORTH_LEGENDRE_BASE, 2);
  CHKERR simple->addDomainField("TAU", L2, AINSWORTH_LEGENDRE_BASE, 1);
  CHKERR simple->addDomainField("EP", L2, AINSWORTH_LEGENDRE_BASE, 3);
  CHKERR simple->addBoundaryField("U", H1, AINSWORTH_LEGENDRE_BASE, 2);
  CHKERR simple->setFieldOrder("U", order);
  CHKERR simple->setFieldOrder("TAU", order-1);
  CHKERR simple->setFieldOrder("EP", order-1);
  CHKERR simple->setUp();
  MoFEMFunctionReturn(0);
}
//! [Set up problem]
```

u

τ

ε^p

Matrix

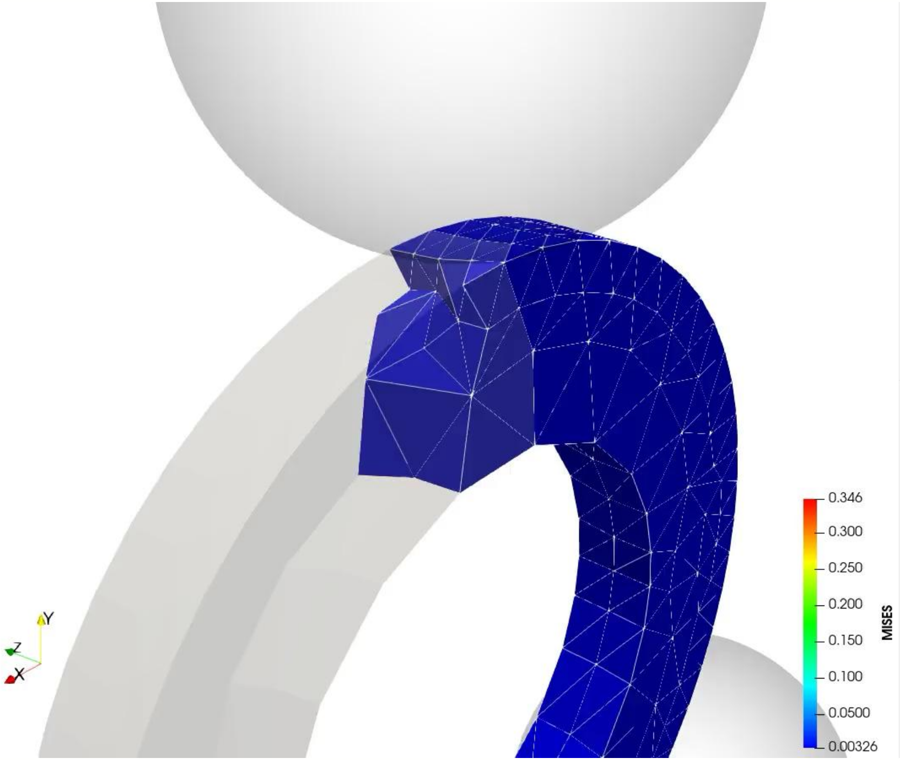


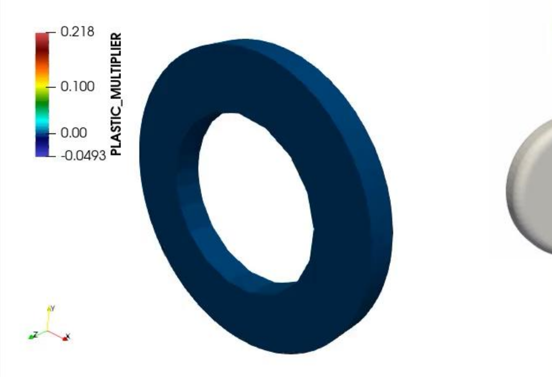
Conservation of linear momentum

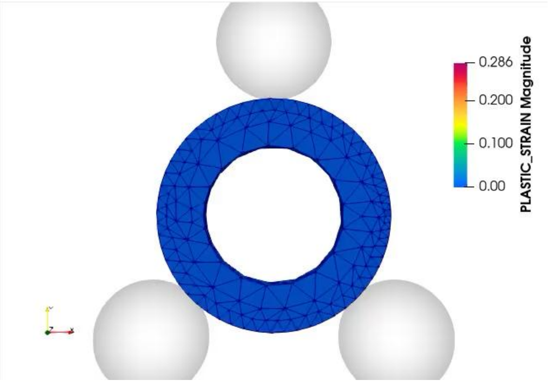
Plastic constrains

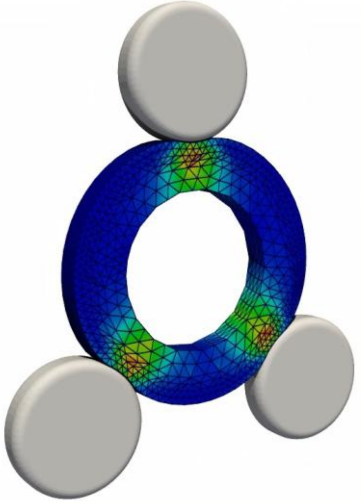
Flow rule

Multifield plasticity



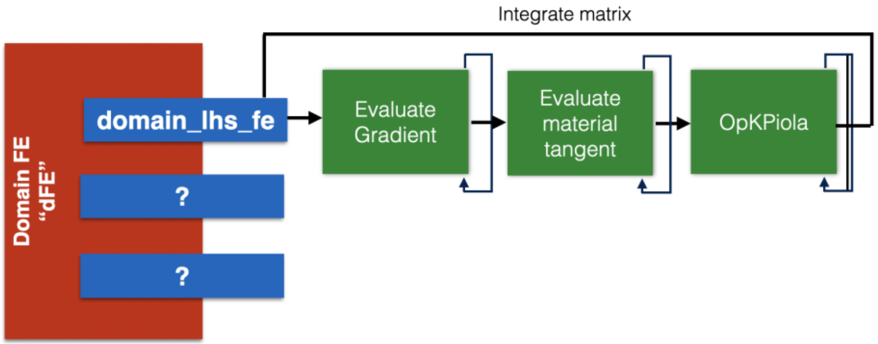


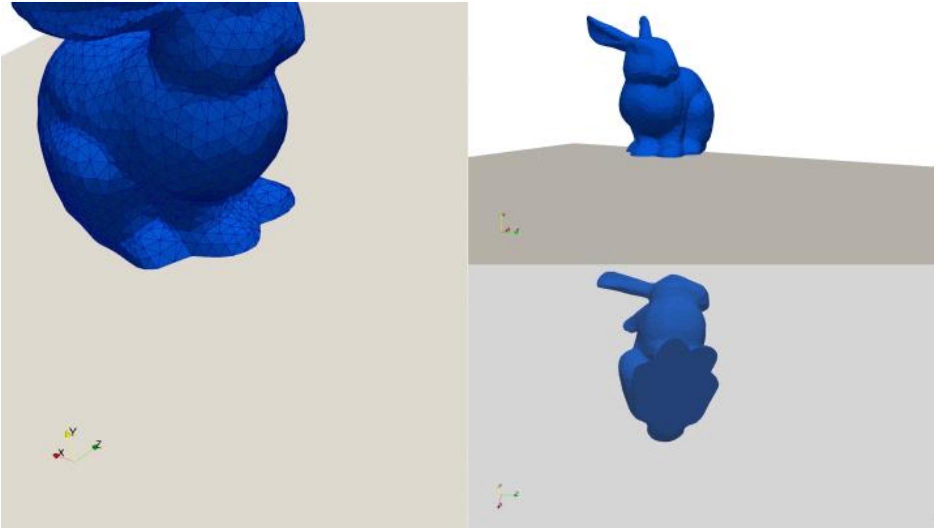




Form integrators

```
// Only used with Henky/nonlinear material
using OpKPiola = FormsIntegrators<DomainEleOp>::Assembly<PETSC>::BiLinearForm<
    GAUSS>::OpGradTensorGrad<1, SPACE_DIM, SPACE_DIM, 1>;
using OpInternalForcePiola = FormsIntegrators<DomainEleOp>::Assembly<
    PETSC>::LinearForm<GAUSS>::OpGradTimesTensor<1, SPACE_DIM, SPACE_DIM>;
```





http://mofem.eng.gla.ac.uk/mofem/html/tutorial_contact_problem.html

University of Glasgow

MFront integration module installation

Spack

Install spack:

```
git clone https://github.com/lorak41/spack.git
```

Install MoFEM + MFront:

```
spack install mofem-cephas+mgis
```

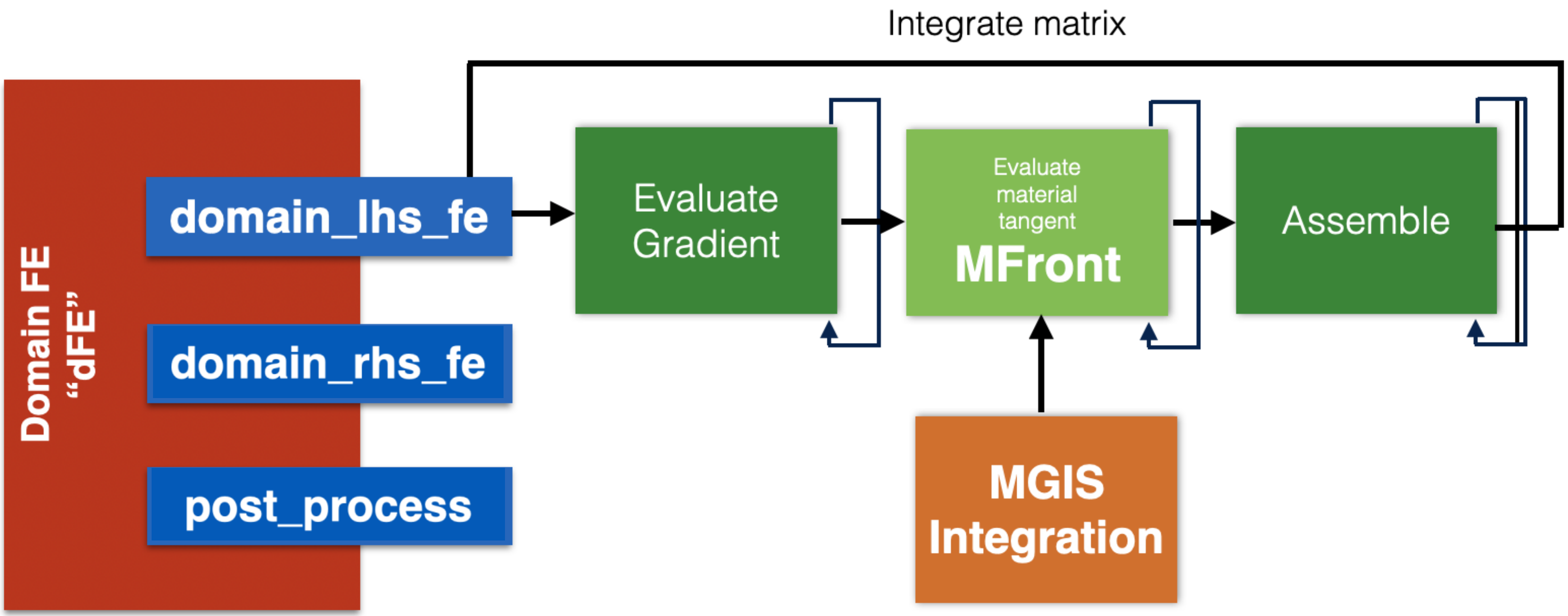
Download MFront integration module into MoFEM

```
cd $MOFEM_INSTALL_DIR/mofem-cephas/mofem/users_modules
git clone git@bitbucket.org:karol41/um_mfront_interface.git  mfront_interface
```

Configure new user module in your build

```
cd $MOFEM_INSTALL_DIR/users_modules_build
export MGIS_PATH=$(spack  find -l --path mgis | awk 'END{print}' | awk 'NF{ print $NF }')"/lib"
touch CMakeCache.txt
./spconfig -DMGIS_PATH=$MGIS_PATH
make –j4
```

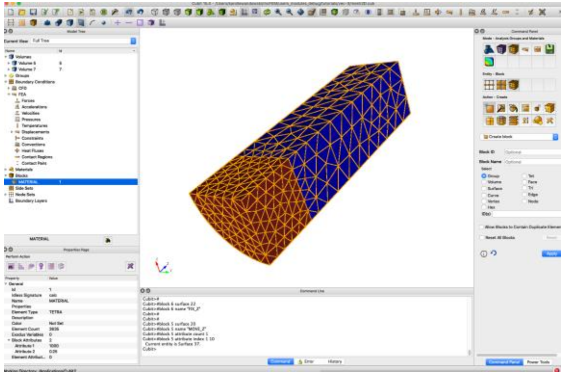
MFront Integration Pipelines

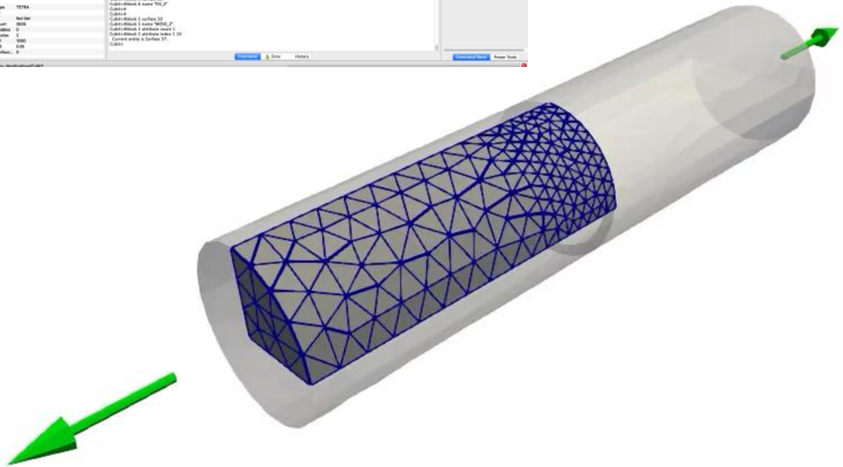


MFront integration
Pre-processing

Boundary conditions:
Force, pressure, displacement, rotation, temperature etc.

Block id: **1**
name: **MATERIAL1**
optional:
param 1: 230e6
param 2: 0.3
…
param n:





C. Miehe, N. Apel, M. Lambrecht (2002)
Anisotropic additive plasticity in the logarithmic strain space. Modular kinematic formulation and implementation based on incremental minimization principles for standard materials

Logarithmic Strain Plasticity Behaviour

LogarithmicStrainPlasticityMiehe.mfront

```
@DSL Implicit;
@Behaviour LogarithmicStrainPlasticity;
@Author Thomas Helfer/Jérémy Bleyer;
@StrainMeasure Hencky;

@Algorithm NewtonRaphson;
@Epsilon 1.e-14;
@Theta 1;

@MaterialProperty stress s0;
s0.setGlossaryName("YieldStress");
@MaterialProperty stress H0;
H0.setEntryName("HardeningSlope");
@MaterialProperty real Rult;
Rult.setEntryName("UltimateStrength");
@MaterialProperty real be;
be.setEntryName("HardeningExponent");

@Brick StandardElastoViscoPlasticity{
  stress_potential : "Hooke" {
                young_modulus : 206913,
                poisson_ratio : 0.29
        },
  inelastic_flow : "Plastic" {
   criterion : "Mises",
                isotropic_hardening : "Linear" {R0 : s0, H : H0},
                isotropic_hardening : "Voce" {R0 : 0, Rinf : Rult, b : be}
  }
};
```

MFront integration
Analysis execution

Command line:

```
./compile_behaviours.sh  LogarithmicStrainPlasticityMiehe.mfront
```

```
#!/bin/bash
mfront --obuild --interface=generic "$@"
```

Saves material library by default into:

```
src/libBehaviour.so
```

Run MoFEM/MFront analysis:

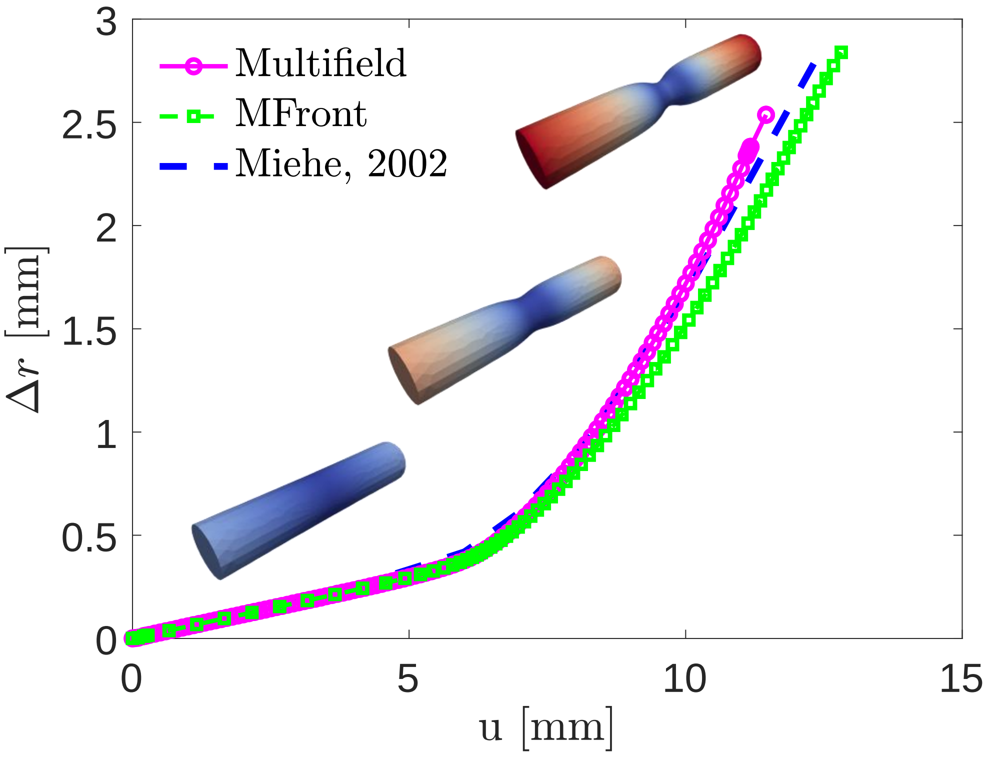
```
./mfront_interface  -file_name necking_mesh.h5m  \
-block_1 LogarithmicStrainPlasticity \
-lib_path_1  src/libBehaviour.so  \ #not necessary
-param_1_0  450 -param_1_1  129 \
-param_1_2  265 -param_1_3  16.93 \
-load_history  load_history.in  \
-order  2 -ts_dt  0.01
```

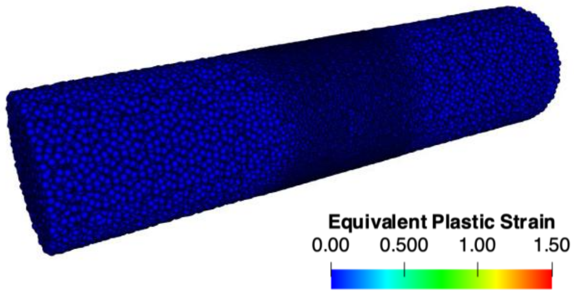
```
LogarithmicStrainPlasticity behaviour loaded on block 1.
Finite Strain Kinematics
Internal variables:
: ElasticStrain
: EquivalentPlasticStrain
External variables:
: Temperature
Material properties:
0 : YieldStrength = 450.
1 : HardeningSlope = 130.
2 : UltimateStrength = 265.
3 : HardeningExponent = 17.
```

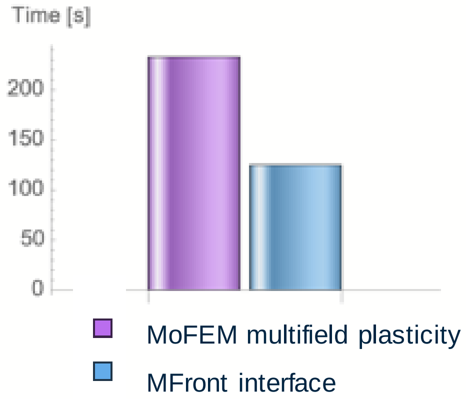
Remaining parameters:

.petscrc

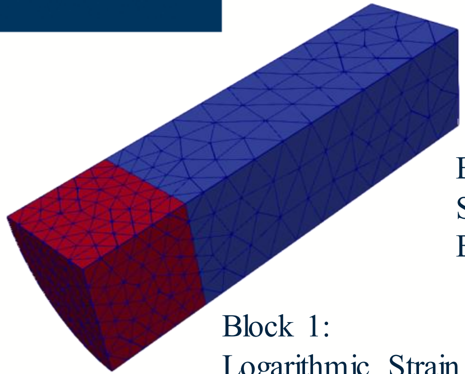
Example results







Example with multiple blocks



Block 2:
SaintVenant Kirchhoff
Elasticity

Block 1:
Logarithmic Strain
Plasticity

Define on the mesh:

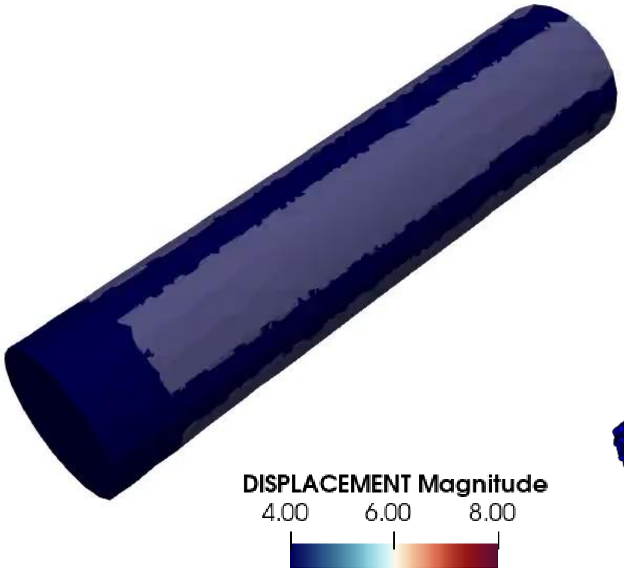
Block id: **1**
name: **MATERIAL1_PLASTIC**
Block id: **2**
name: **MATERIAL2_ELASTIC**

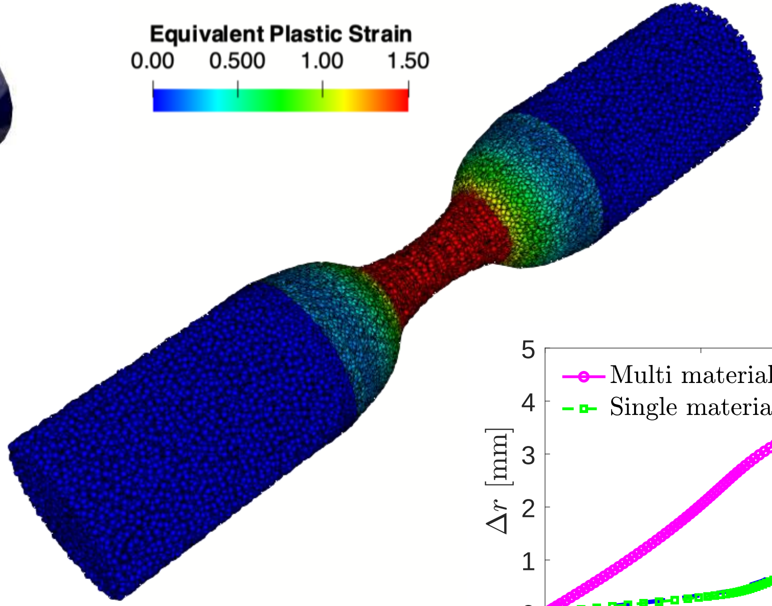
```
./compile_behaviours.sh LogarithmicStrainPlasticityMiehe.mfront SaintVenantKirchhoffElasticity.mfront
```

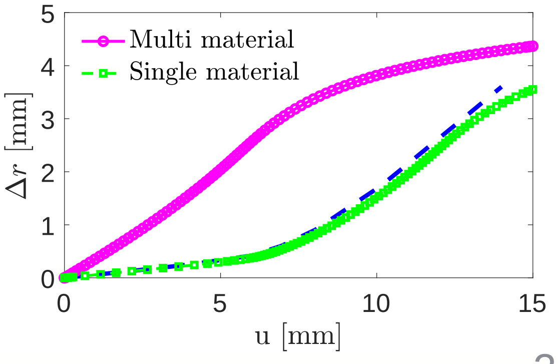
Run MoFEM/MFront analysis:

```
./mfront_interface -file_name necking_mesh2.h5m \
-block_1 LogarithmicStrainPlasticity \
-param_1_0 450 -param_1_1 129 -param_1_2 265 -param_1_3 16.93 \
-block_2 SaintVenantKirchhoffElasticity \
-param_2_0 2e6 -param_2_1 0.3 \
```

Example with multiple blocks







Summary

MFront user module:

- Generic, parallel and flexible implementation
- Arbitrary and heterogeneous order of approximation
- Supports both small and large strains behaviours


University of Glasgow
Thank you!
#UofGWorldChangers
@UofGlasgow