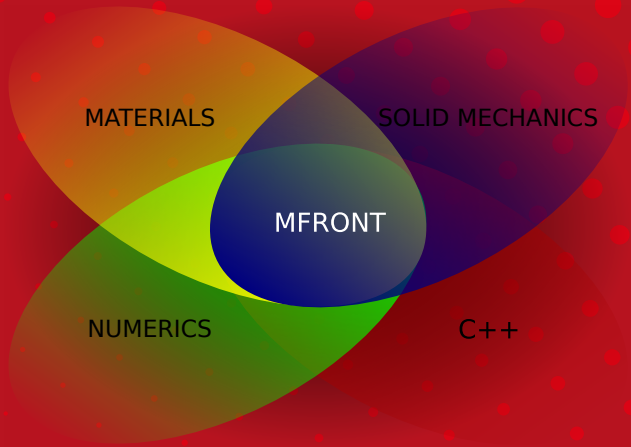
Présentation de MFront, MGIS et MFrontGallery



DE LA RECHERCHE À L'INDUSTRIE

GRD Bois

17 Novembre 2022

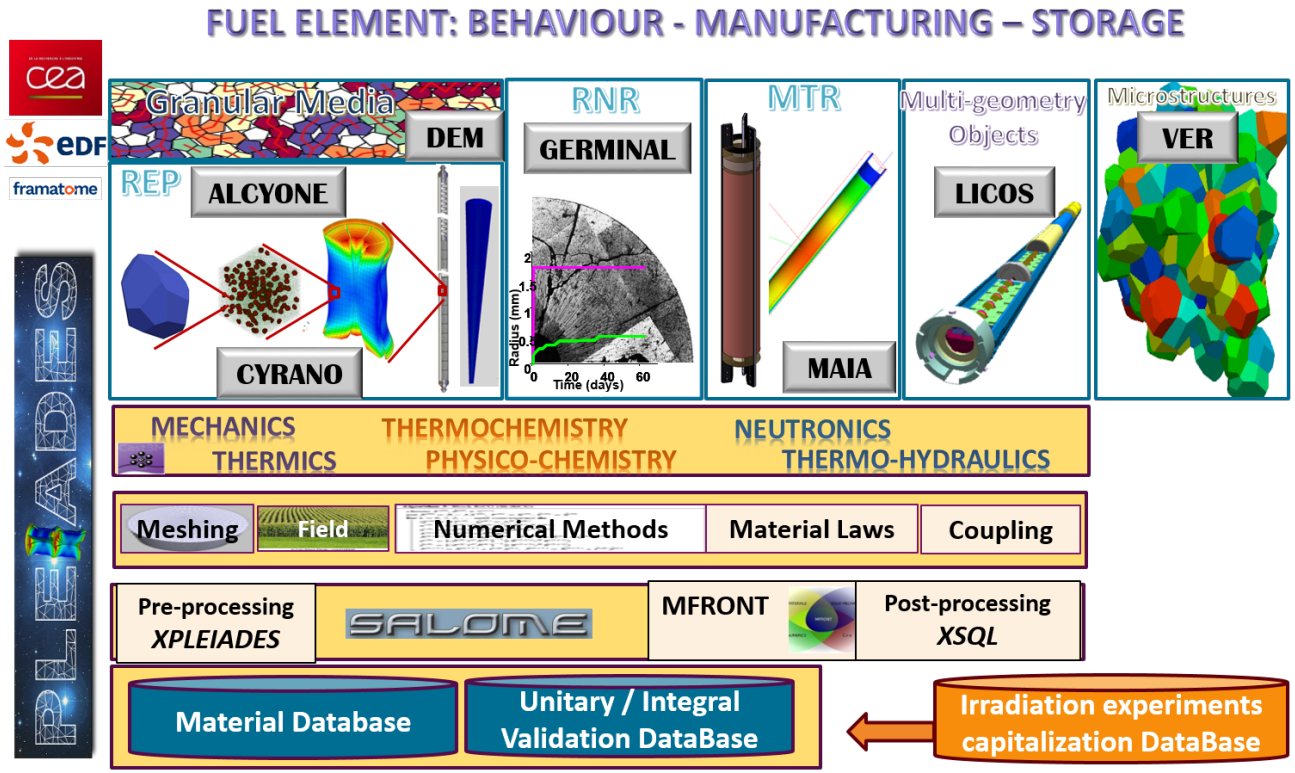
T. Helfer, M. Wangermez

Outline

- **Context**
- **Some applications of MFront**
- **Mechanical behaviours**
- **An overview of `MFront`**
- **An overview of `MGIS`**
- **The MFrontGallery project**

Context

The Pleiades platform



- A wide range of materials (ceramics, metals, composites).
- A wide range of mechanical phenomena and behaviours.
  - Creep, swelling, irradiation effects, phase transitions, etc..
- A wide range of mechanical loadings.

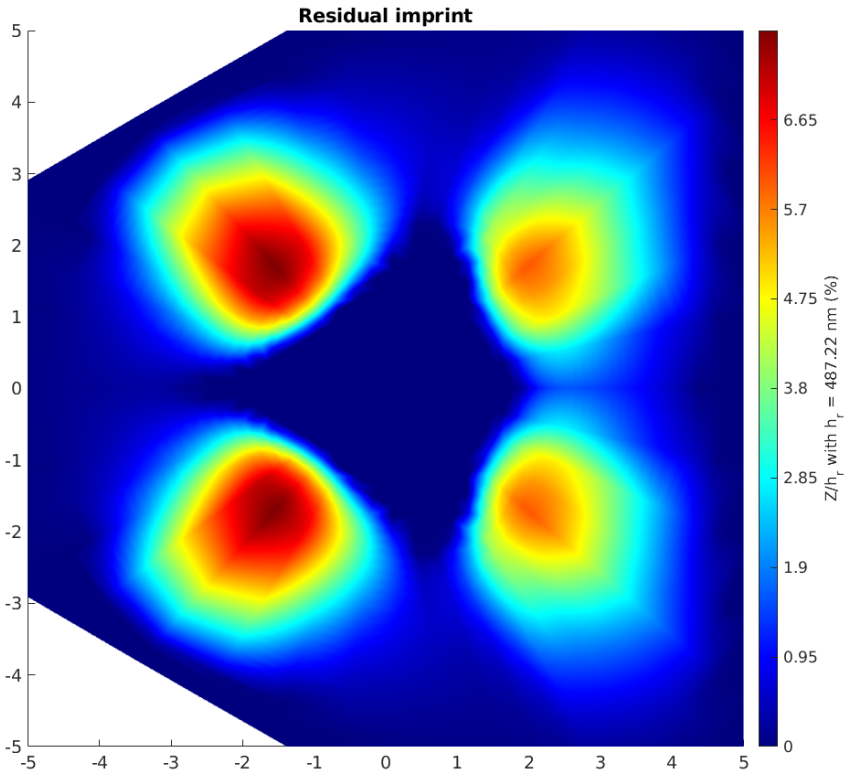
Open-source projects developped in the Pleiades platform

- MFront.
  - https://www.sciencedirect.com/science/article/pii/S0898122115003132
  - https://thelfer.github.io/tfel/web/index.html
  - https://thelfer.github.io/tfel/web/gallery.html
- MGIS (MFrontGenericInterfaceSupport).
  - https://joss.theoj.org/papers/10.21105/joss.02003
  - https://thelfer.github.io/mgis/web/index.html
  - https://thelfer.github.io/mgis/web/bindings-cxx.html
- MFrontGallery.
  - https://github.com/thelfer/MFrontGallery/tree/master/docs/papers/joss
  - https://thelfer.github.io/MFrontGallery/web/index.html

Some applications of MFront

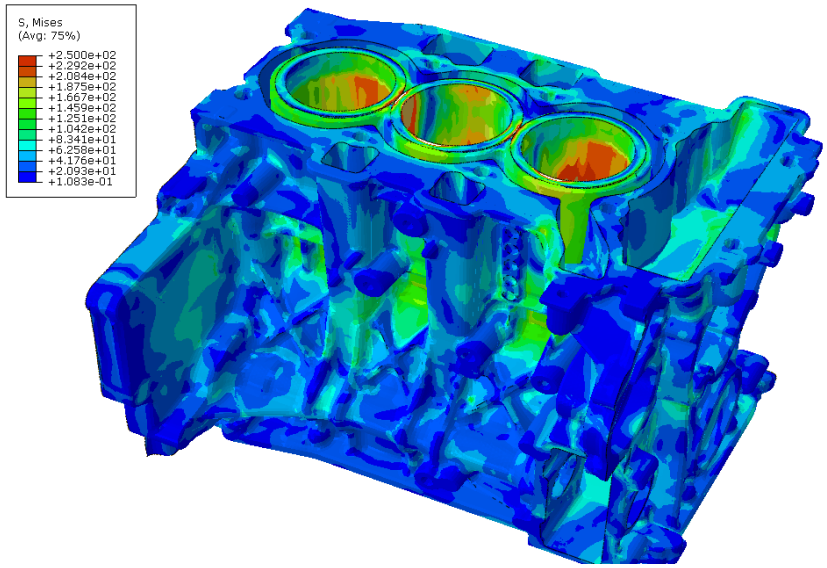
Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

Berkovich indentation on a single crystal



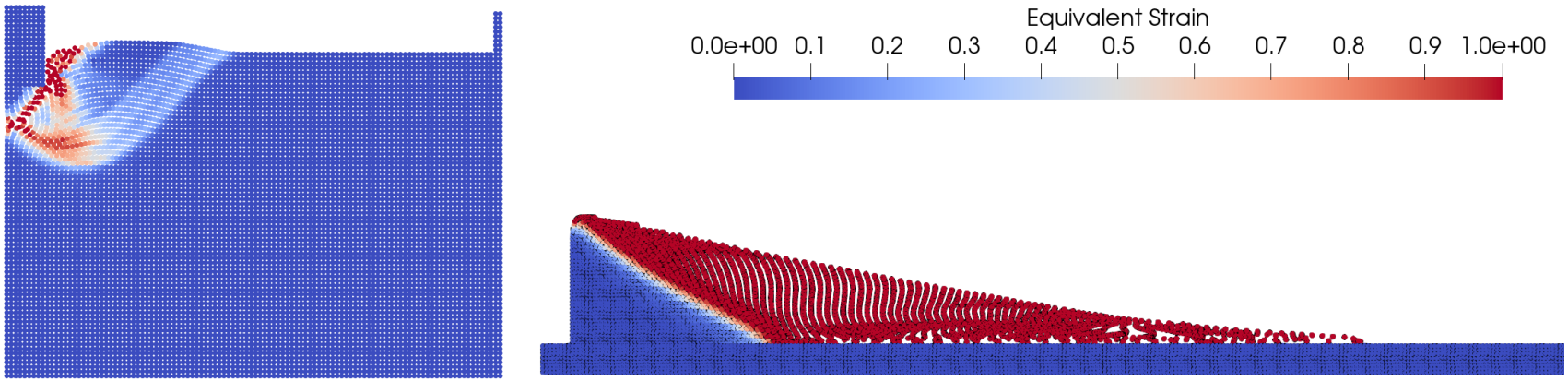
- Normalised residual topography after an indentation test on a single crsytal of copper with Méric-Cailletaud' finite stain behaviour using `Ansys`
- Courtesy of A. Bourceret, FEMTO

Design of a cylinder block



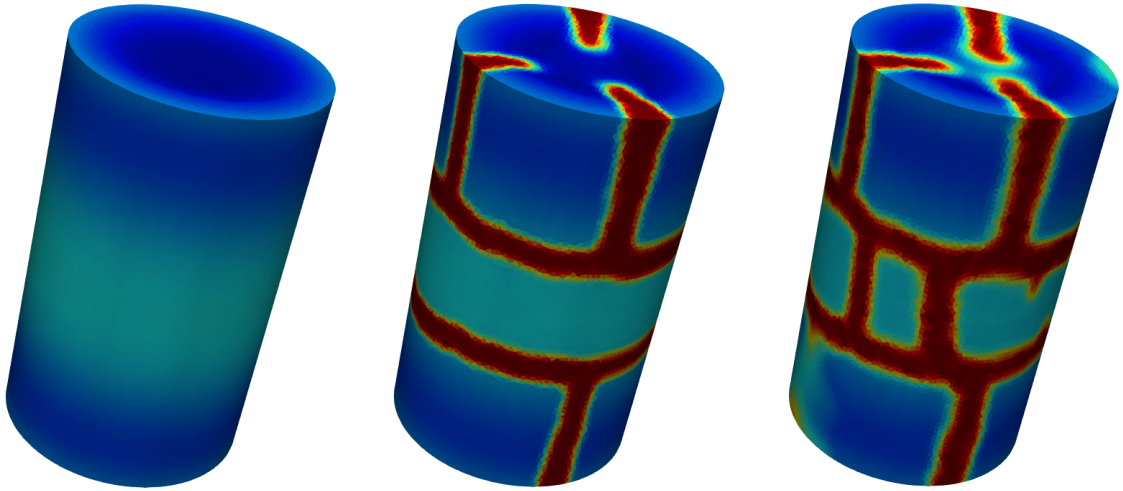
- Industrial thermomechanical design of a cylinder block with an with `MFront` and `Abaqus` at Groupe PSA.
- This study is one result of the PhD thesis of L. Jacquinot which provides a continuous modelling of the AlSi9Cu3Mg aluminium alloy behaviour from manufacturing to final usage (see the attached figure).
- The finite element model contains 11e6 dofs (3e6 elements) and has been solved using a 72 cores computers with Abaqus 2016.

Geo-materials simulated using a material point method

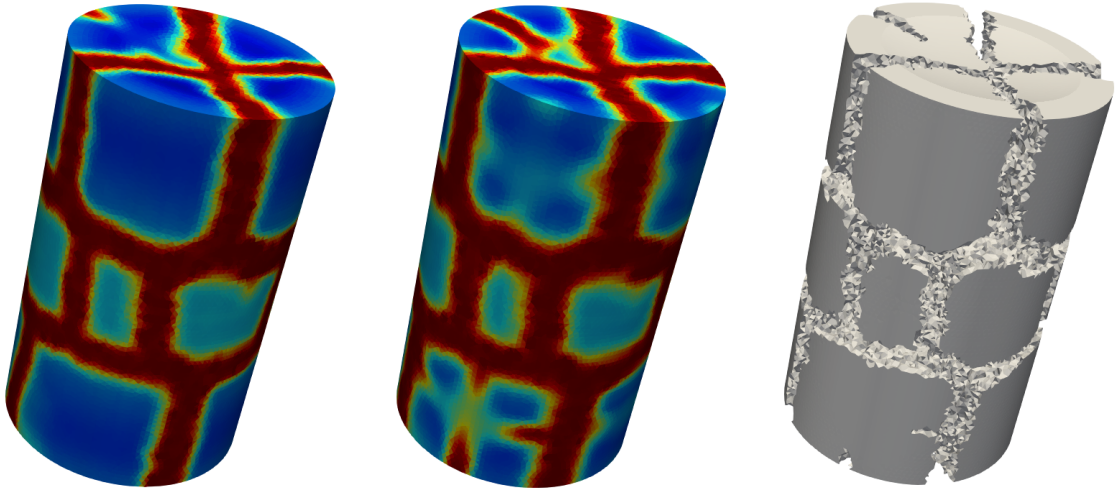


- Equivalent strain contour from the material point method simulations : the footing (left) and the column collapse
- Courtesy of Ning Guo, Wenlong Li (Zhejiang University)

Nuclear fuel rod fragmentation (Ye Lu)

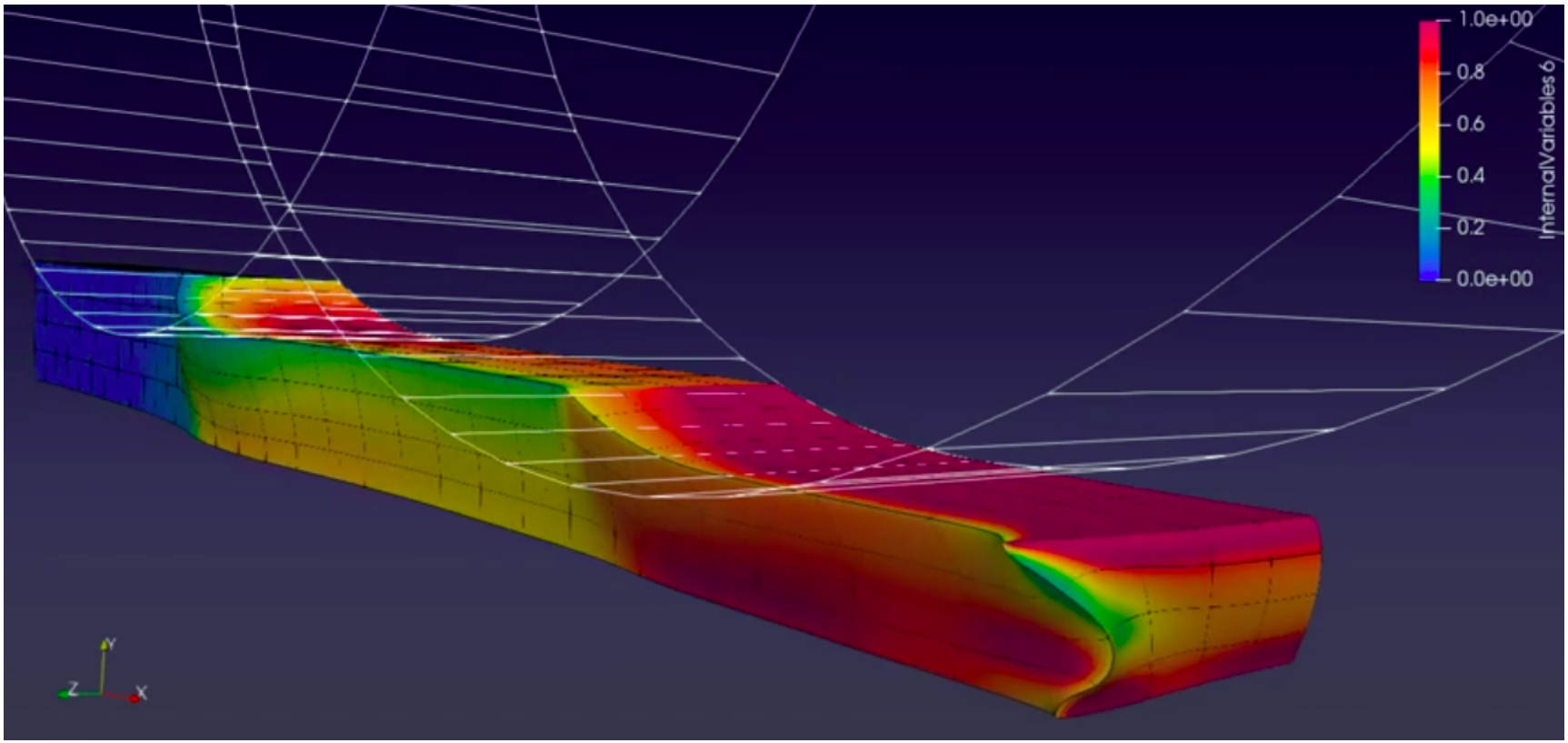


(a) 10 K/mm (b) 20 K/mm (c) 50 K/mm



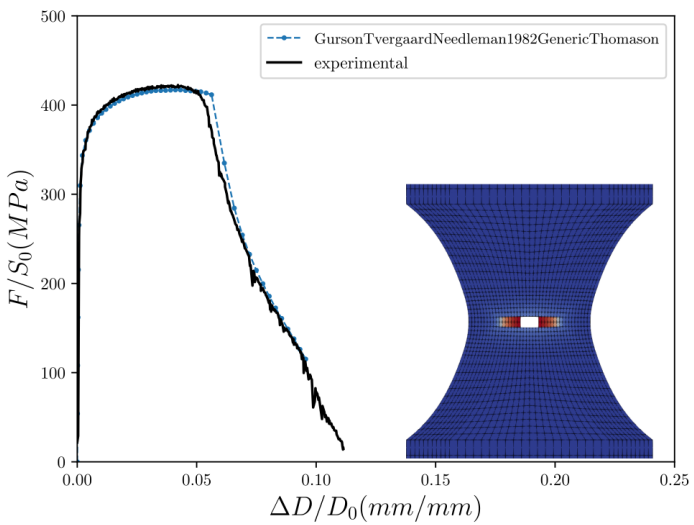
(d) 90 K/mm (e) 150 K/mm (f) 150 K/mm

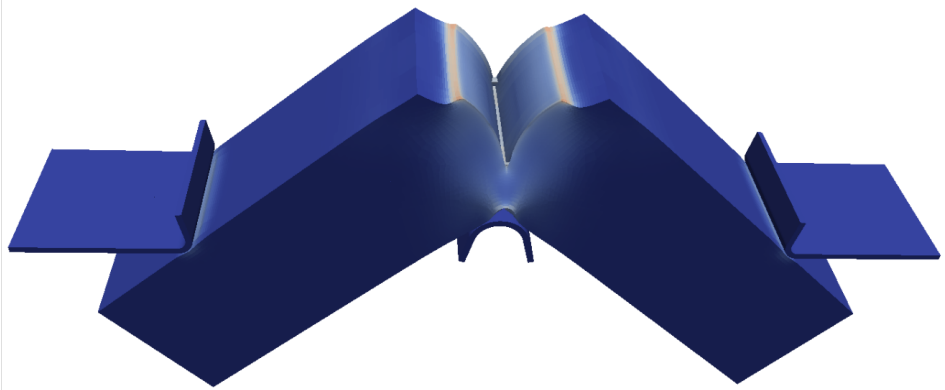
Simulation of rolling



- Simulation of rolling using the innovative CEA' proto-application `MEFISTO` (implicit/explicit solver)
- Courtesy of O. Jamond, CEA

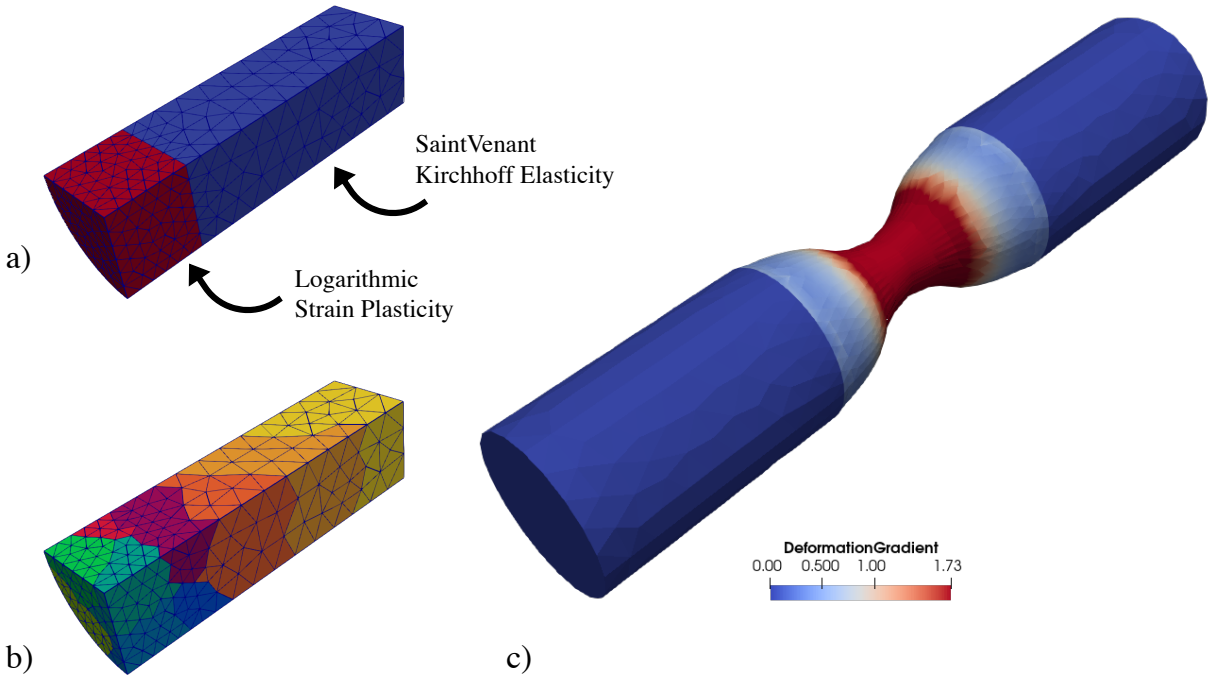
Ductile failure





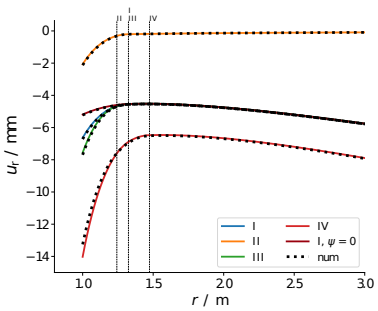
- Simulation of ductile failure using CEA' `Cast3M` finite element solver.
- GTN behaviour in the logarithmic space.
- Illustration of the extension of the `StandardElastoViscoPlasticity` brick to porous plasticity.
- Courtesy of M. Shokeir and J. Hure, CEA

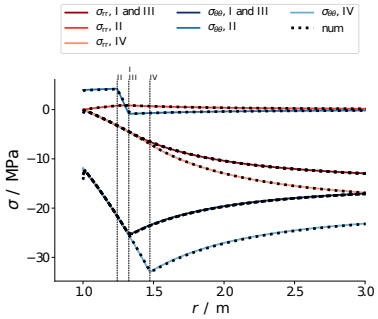
Necking of a rod in finite strain plasticity



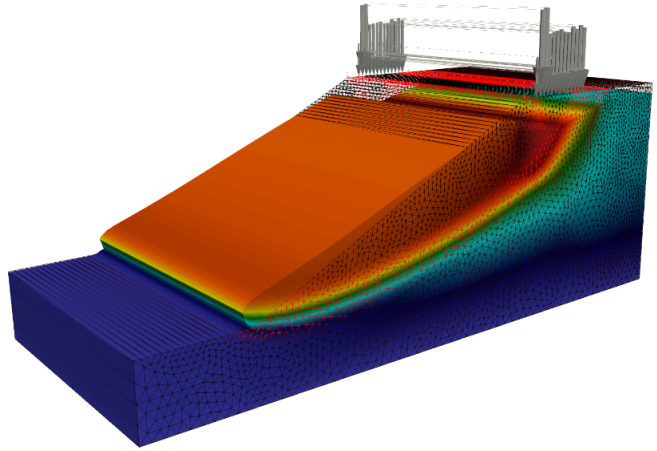
- Example of usage of `MFront` in the `MoFEM` finite element solver.
- a) Discretised one eighth of the geometry. b) Parallel domain decomposition. c) Deformed specimen and distribution of deformation gradient.
- Courtesy of K. Lewandowski and L. Kaczmarczyk, University of Glasgow.

Slope failure analysis



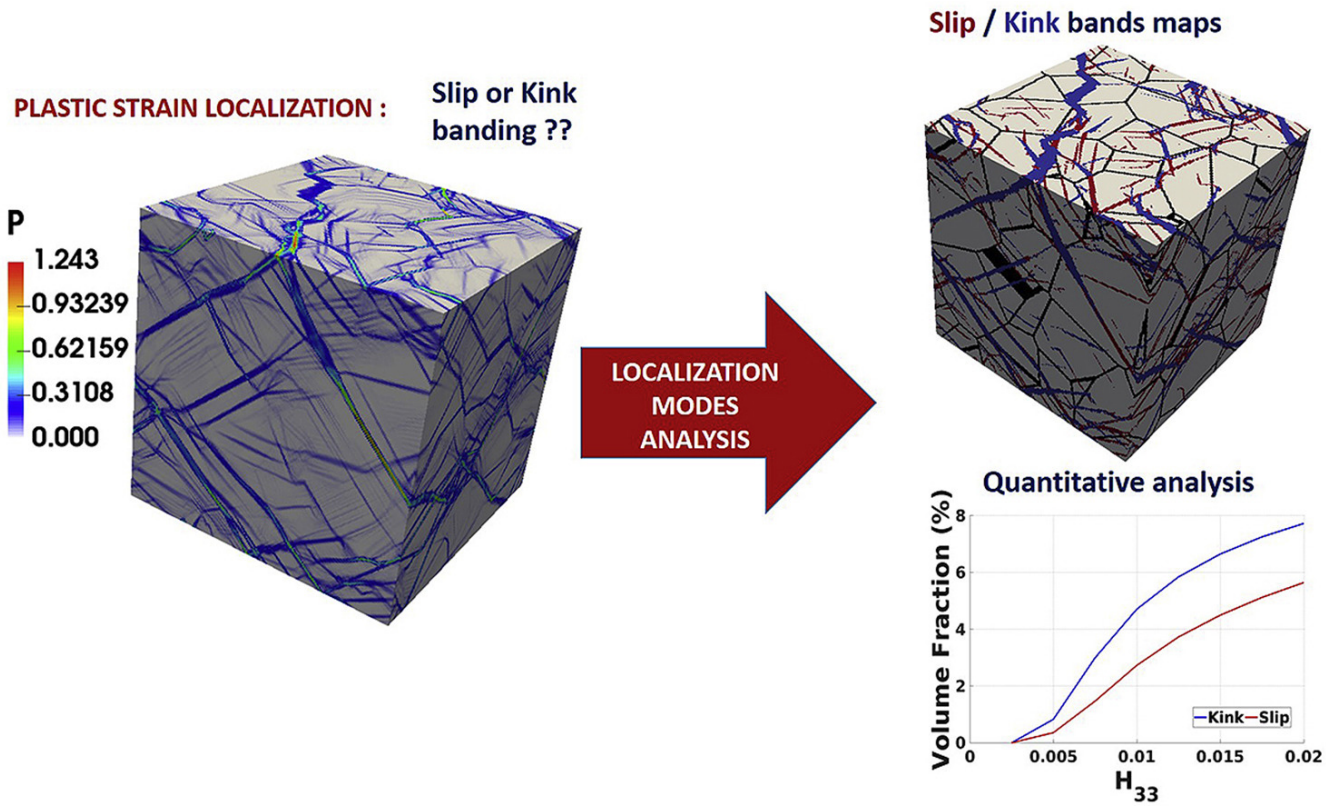


Slope failure analysis with phi-c-reduction in OpenGeoSys



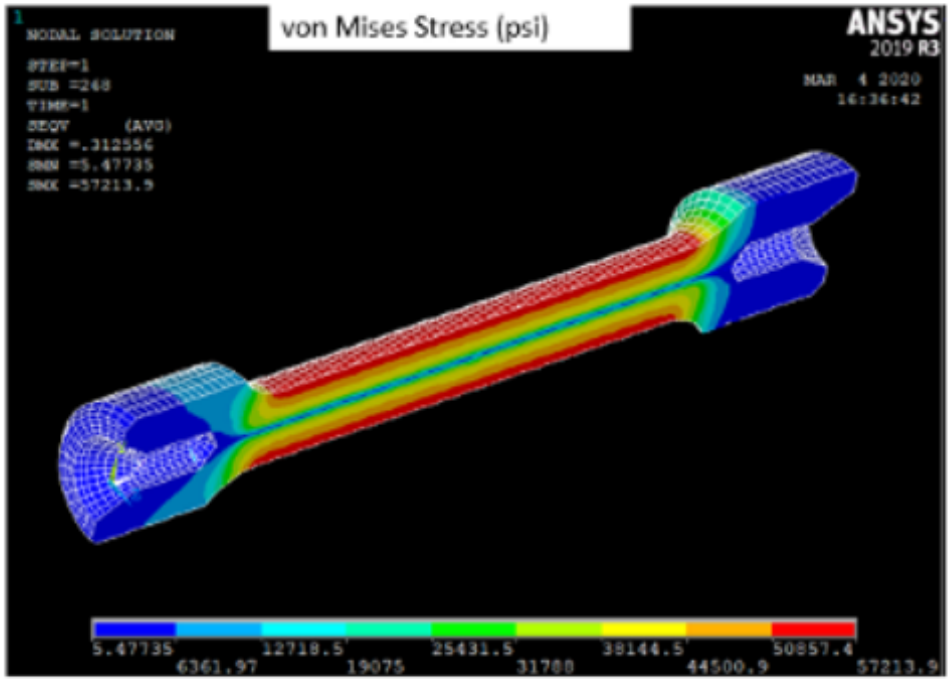
- Analytical verification (left)
- Slope failure analysis with strength reduction in `OpenGeoSys` by T. Deng and T. Nagel (Technische Universität Bergakademie Freiberg)
- For details, see `https://opengeosys.org/docs/benchmarks/small-deformations/slope_stability.pdf`.
- The implementation of the behaviour is described here : `http://tfel.sourceforge.net/MohrCoulomb.html`

Intragranular localization induced by softening crystal plasticity



- https://www.sciencedirect.com/science/article/pii/S1359645419303696
- Based on the CEA' AMITEX_FFTP solver
- Courtesy of L. Gelebart (CEA)

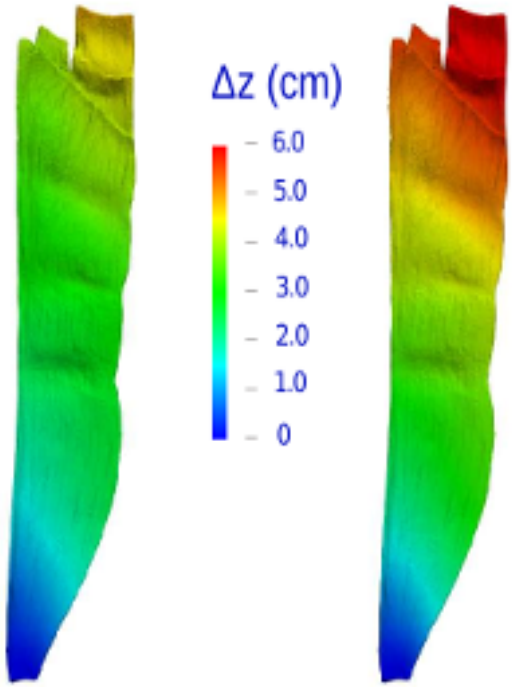
Torsional twist of a bar



Torsional twist of a notched bar using an Hosford plastic behaviour with a bilinear hardening law
Alex Grishin
Ansys MAPDL 2020

- Plastic behaviour based on the Hosford criterion. Implementation described here : `http://tfel.sourceforge.net/hosford.html`

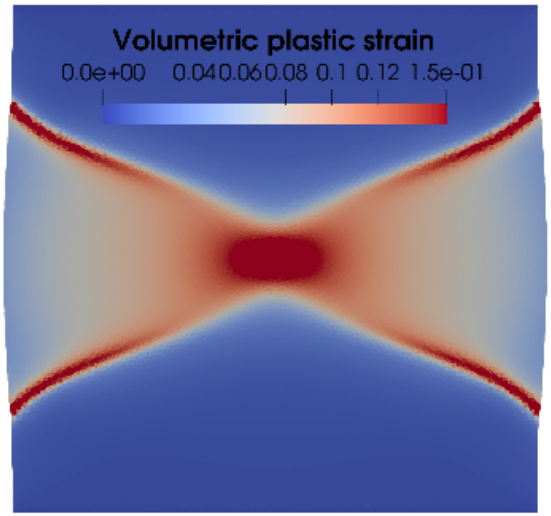
Modelling of abdominal muscles

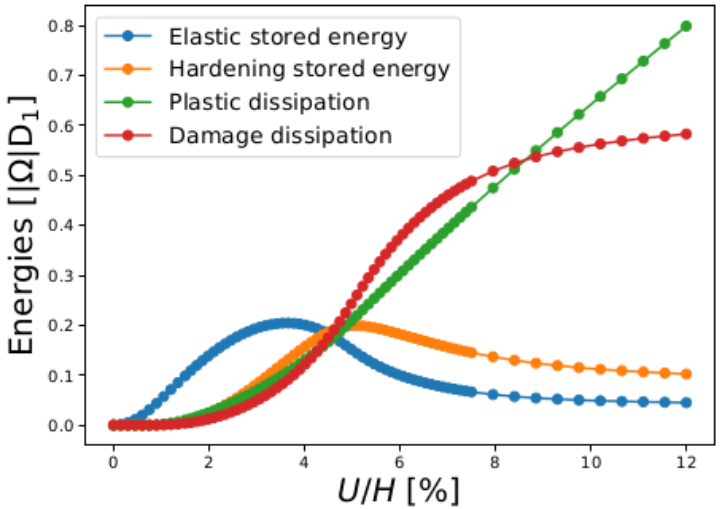


Modelling of abdominal muscles
Lluís Tuset, Gerard Fortuny, Joan Herrero, Dolors Puigjaner, Josep M. López
Code_Aster
2019

- Hyperelastic behaviour in `code_aster`

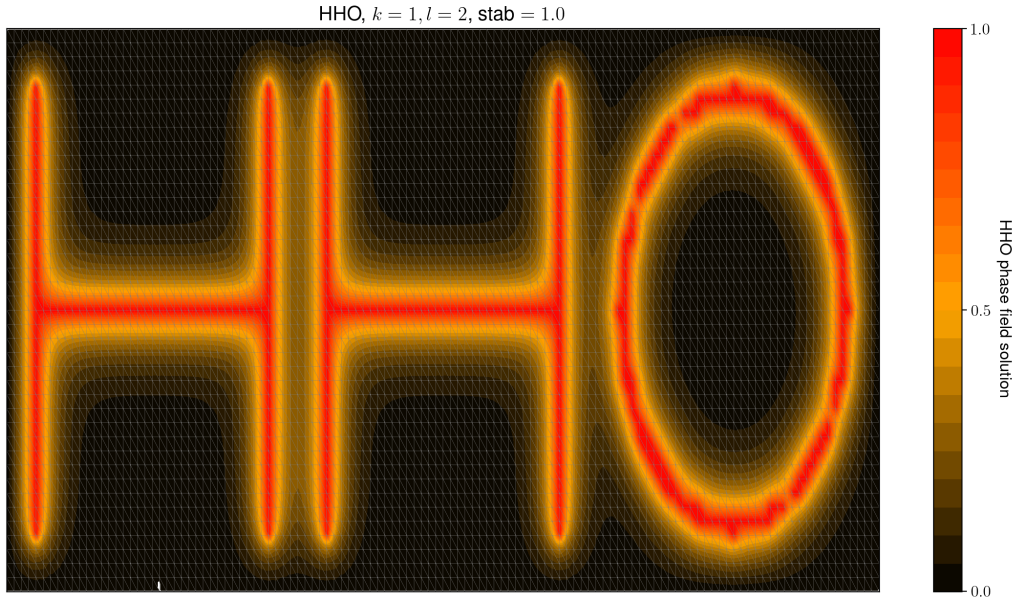
Variational approaches to geomaterials





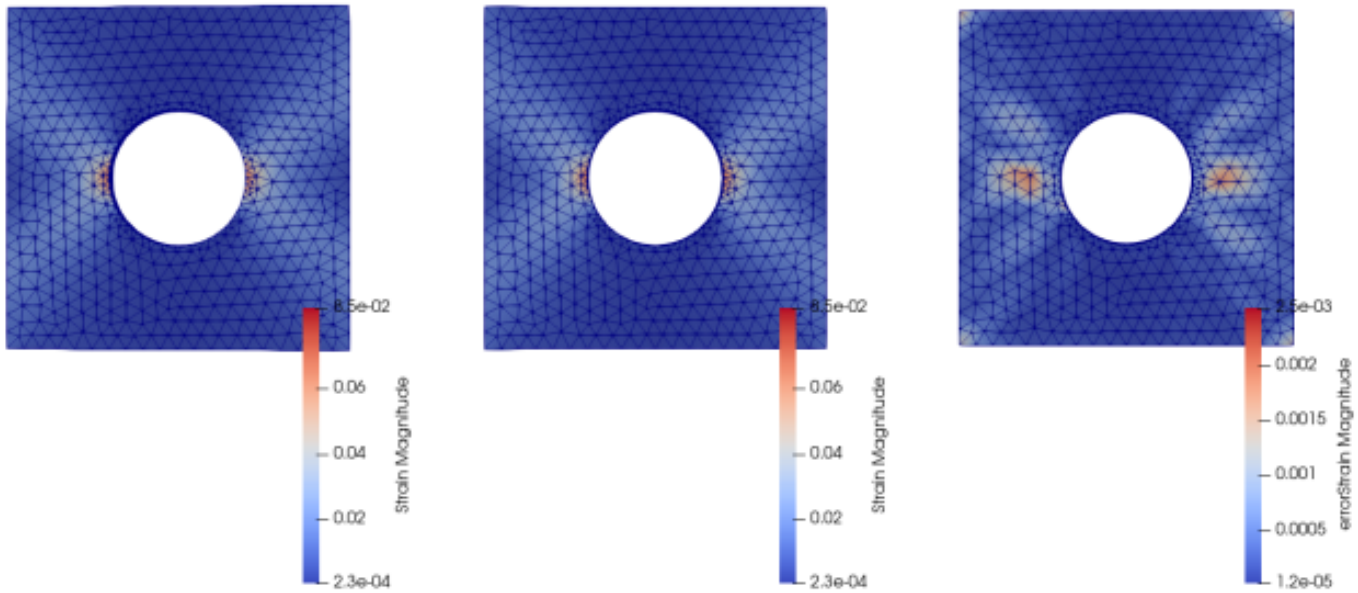
- Coupling damage and plasticity accounts for classical geomaterials behaviours (contractancy and dilatancy)
- Currently tested with the `mgis.fencis` 'python' module. Shall be introduced in `code_aster` during the PhD of G. Bacquaert.
- Courtesy of G. Bacquaert, V. Alves, Fernandes , J. Bleyer, D. Kondo, C. Maurini, S. Raude, F. Voldoire

Phase field approach to fracture using the Hybrid High Order method



- PhD of D. Siedel.
- Local treatment of the irreversibility constraint at the element level.

Neural network based constitutive models



- Internship of M. Duvillard. Courtesy of L. Giraldi.
- Neural network trained on unit tests at the material point level.

Mechanical behaviours

Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

Role of the mechanical behaviours

- Mechanical equilibrium : find $\Delta\vec{\mathbb{U}}$ such as :

$$\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{O}} \quad \text{avec} \quad \vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{F}}_i\left(\Delta\vec{\mathbb{U}}\right) - \vec{\mathbb{F}}_e$$

Role of the mechanical behaviours

- Mechanical equilibrium : find $\Delta\vec{\mathbb{U}}$ such as :

$$\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{O}} \quad \text{avec} \quad \vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{F}}_i\left(\Delta\vec{\mathbb{U}}\right) - \vec{\mathbb{F}}_e$$

- element contribution to inner forces :

$$\begin{aligned}\vec{\mathbb{F}}_i^e &= \int_{V^e} \underline{\sigma}_{t+\Delta t}\left(\Delta\,\underline{\epsilon}^{to}, \Delta\, t\right) : \underline{\mathbf{B}}\ \mathrm{d}V \\ &= \sum_{i=1}^{N^G} \left(\underline{\sigma}_{t+\Delta\, t}\left(\Delta\underline{\epsilon}^{to}\left(\vec{\eta}_i\right), \Delta\, t\right) : \underline{\underline{\mathbf{B}}}\left(\vec{\eta}_i\right)\right) w_i\end{aligned}$$

where $\underline{\mathbf{B}}$ gives the relationship between $\Delta\,\underline{\epsilon}^{to}$ and $\Delta\vec{\mathbb{U}}$

Role of the mechanical behaviours

- Mechanical equilibrium : find $\Delta\vec{\mathbb{U}}$ such as :

$$\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{O}} \quad \text{avec} \quad \vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{F}}_i\left(\Delta\vec{\mathbb{U}}\right) - \vec{\mathbb{F}}_e$$

- element contribution to inner forces :

$$\vec{\mathbb{F}}_i^e = \sum_{i=1}^{N^G} \left(\underline{\sigma}_{t+\Delta t}\left(\Delta\underline{\epsilon}^{to}\left(\vec{\eta}_i\right), \Delta t\right) : \underline{\underline{\mathbf{B}}}\left(\vec{\eta}_i\right)\right) w_i$$

- Resolution using the Newton-Raphson algorithm :

$$\Delta\vec{\mathbb{U}}^{n+1} = \Delta\vec{\mathbb{U}}^n - \left(\left.\frac{\partial\vec{\mathbb{R}}}{\partial\Delta\vec{\mathbb{U}}}\right|_{\Delta\vec{\mathbb{U}}^n}\right)^{-1} .\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}^n\right) = \Delta\vec{\mathbb{U}}^n - \underline{\underline{\mathbb{K}}}^{-1} .\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}^n\right)$$

Role of the mechanical behaviours

- Mechanical equilibrium : find$\Delta\vec{\mathbb{U}}$ such as :

$$\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{O}} \quad \text{avec} \quad \vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}\right) = \vec{\mathbb{F}}_i\left(\Delta\vec{\mathbb{U}}\right) - \vec{\mathbb{F}}_e$$

- element contribution to inner forces :

$$\vec{\mathbb{F}}_i^e = \sum_{i=1}^{N^G} \left(\underline{\sigma}_{t+\Delta t}\left(\Delta\underline{\epsilon}^{to}\left(\vec{\eta}_i\right), \Delta t\right) : \underline{\underline{\mathbf{B}}}\left(\vec{\eta}_i\right)\right) w_i$$

- Resolution using the Newton-Raphson algorithm :

$$\Delta\vec{\mathbb{U}}^{n+1} = \Delta\vec{\mathbb{U}}^n - \underline{\underline{\mathbb{K}}}^{-1}.\vec{\mathbb{R}}\left(\Delta\vec{\mathbb{U}}^n\right)$$

- element contribution to the stiffness :

$$\underline{\underline{\mathbb{K}}}^e = \sum_{i=1}^{N^G} {}^t\underline{\underline{\mathbf{B}}}\left(\vec{\eta}_i\right) : \frac{\partial\Delta\underline{\sigma}}{\partial\Delta\underline{\epsilon}^{to}}\left(\vec{\eta}_i\right) : \underline{\underline{\mathbf{B}}}\left(\vec{\eta}_i\right) w_i$$

$\frac{\partial\Delta\underline{\sigma}}{\partial\Delta\underline{\epsilon}^{to}}$ is the **consistent tangent operator**

Main functions of the mechanical behaviour

$$\left(\underline{\epsilon}^{to}\big|_t \, , \vec{Y}\Big|_t \, , \Delta \underline{\epsilon}^{to} \, , \Delta t \right) \underbrace{\Longrightarrow}_{behaviour} \left(\underline{\sigma}|_{t+\Delta t} \, , \vec{Y}\Big|_{t+\Delta t} \, , \frac{\partial \Delta \underline{\sigma}}{\partial \Delta \underline{\epsilon}^{to}} \right)$$

- Given a strain increment $\Delta \underline{\epsilon}^{to}$ over a time step Δt, the mechanical behaviour must compute :
  - The value of the stress $\underline{\sigma}|_{t+\Delta t}$ at the end of the time step.
  - The value of internal state variables, noted $\vec{Y}\Big|_{t+\Delta t}$ at the end of the time step.
  - The consistent tangent operator : $\frac{\partial \Delta \underline{\sigma}}{\partial \Delta \underline{\epsilon}^{to}}$
- For specific cases, the mechanical behaviour shall also provide :
  - a prediction operator
  - the elastic operator (Abaqus-Explicit, Europlexus)
  - estimation of the stored and dissipated energies (Abaqus-Explicit)

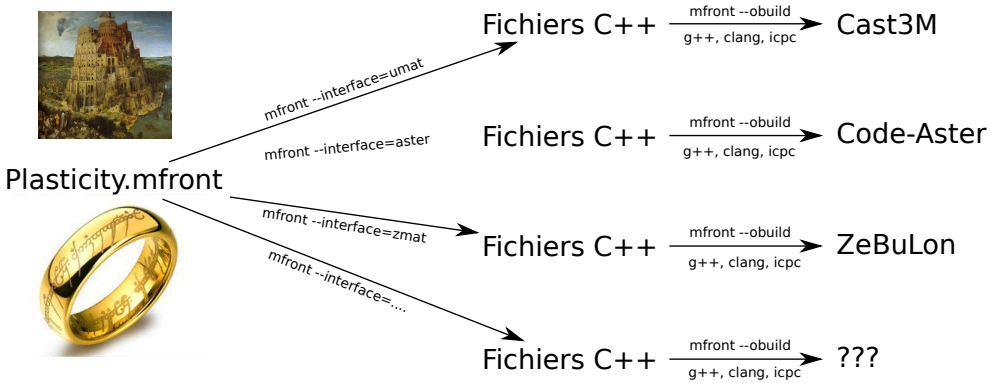
Other functions of the mechanical behaviour

- Provide a estimation of the next time step for time step automatic adaptation
- Check bounds :
  - Physical bounds
  - Standard bounds
- Clear error messages
- Parameters
  - It is all about Quality Assurance !
  - Parametric studies, identification, etc. . .
- Generate 'MTest' files on integration failures
- Generated example of usage :
  - Generation of MODELISER/MATERIAU instructions (Cast3M)
  - Input file for Abaqus, Ansys
- Provide information for dynamic resolution of inputs (MTest/Aster/Europlexus) :
  - Numbers Types (scalar, tensors, symmetric tensors)
  - Entry names /Glossary names. . .

An overview of MFront and MGIS

Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

MFront goals



- `MFront` is a code generation tool dedicated to material knowledge (material properties, mechanical behaviours, point-wise models) :
  - Support for small and finite strain behaviours, cohesive zone models, **generalised behaviours** (non local and or multiphysics).
- Main goals :
  - Numerical efficiency (see various benchmarks on the website).
  - Portability (`Cast3M`, `Cyrano`, `code_aster`, `Europlexus`, `TMFTT`, `AMITEX_FFTP`, `Abaqus`, `CalculiX`, `MTest`).
  - **Ease of use** : *Longum iter est per praecepta, breve et efficax per exempla* (It's a long way by the rules, but short and efficient with examples).

An example of the StandardElasticity-VicoPlasticity brick

```
@DSL Implicit;
@Behaviour MohrCoulomAbboSloan3;

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

@Brick StandardElastoViscoPlasticity{
  stress_potential : "Hooke" {
    young_modulus : 150.e3,
    poisson_ratio : 0.3
  },
  inelastic_flow : "Plastic" {
    criterion : "MohrCoulomb" {
      c : 3.e1,          // cohesion
      phi : 0.523598775598299,    // friction angle or dilatancy angle
      lodeT : 0.506145483078356,  // transition angle as defined by Abbo and Sloan
      a : 1e1          // tension cuff-off parameter
    },
    flow_criterion : "MohrCoulomb" {
      c : 3.e1,          // cohesion
      phi : 0.174532925199433,    // friction angle or dilatancy angle
      lodeT : 0.506145483078356,  // transition angle as defined by Abbo and Sloan
      a : 3e1          // tension cuff-off parameter
    },
    isotropic_hardening : "Linear" {R0 : "0"}
  }
};
```

- The `StandardElasticityVicoPlasticity` brick allows to implement complex visco-plastic behaviours with a declarative syntax using predefined components.

An simple example with the Implicit DSL and the StandardElasticity brick

```
@DSL Implicit;
@Behaviour Norton;
@Brick StandardElasticity;

@MaterialProperty stress E;
E.setGlossaryName("YoungModulus");
@MaterialProperty real ν, A, nn;
ν.setGlossaryName("PoissonRatio");
A.setEntryName("NortonCoefficient");
nn.setEntryName("NortonExponent");

@StateVariable real    p;
p.setGlossaryName("EquivalentViscoplasticStrain");

@Integrator{
  constexpr const auto Mᵉ = Stensor4::M();
  const auto μ = computeMu(E, ν);
  const auto σᵉ = sigmaeq(σ);
  const auto iσᵉ = 1 / (max(σᵉ, real(1.e-12) · E));
  const auto vᵖ = A · pow(σᵉ, nn);
  const auto ∂vᵖ/∂σᵉ = nn · vᵖ · iσᵉ;
  const auto n = 3 · deviator(σ) · (iσᵉ / 2);
  // Implicit system
  fεᵉˡ += Δp · n;
  fp -= vᵖ · Δt;
  // jacobian
  ∂fεᵉˡ/∂Δεᵉˡ += 2 · μ · θ · dp · iσᵉ · (Mᵉ - (n ⊗ n));
  ∂fεᵉˡ/∂Δp = n;
  ∂fp/∂Δεᵉˡ = -2 · μ · θ · ∂vᵖ/∂σᵉ · Δt · n;
} // end of @Integrator
```

- Implicit integration.
- Implicit system :

$$\begin{cases} f_{\underline{\epsilon}^{el}} = \Delta\,\underline{\epsilon}^{el} - \Delta\,\underline{\epsilon}^{to} + \Delta\,p\,\underline{\mathbf{n}} \\ f_p = \Delta\,p - A\,\sigma_{eq}^{n} \end{cases}$$

- Jacobian :

$$\begin{cases} \dfrac{\partial f_{\underline{\epsilon}^{el}}}{\partial \Delta\,\underline{\epsilon}^{el}} = \underline{\underline{\mathbf{I}}} + \dfrac{2\,\mu\,\theta\,\Delta\,p}{\sigma_{eq}}\left(\underline{\underline{\mathbf{M}}} - \underline{\mathbf{n}} \otimes \underline{\mathbf{n}}\right) \\ \dfrac{\partial f_{\underline{\epsilon}^{el}}}{\partial \Delta\,p} = \underline{\mathbf{n}} \\ \dfrac{\partial f_p}{\partial \Delta\,\underline{\epsilon}^{el}} = -2\,\mu\,\theta\,A\,n\,\sigma_{eq}^{n-1}\,\Delta\,t\,\underline{\mathbf{n}} \end{cases}$$

- **All programming and numerical details are hidden (by default).**

Support for quantities in MFront

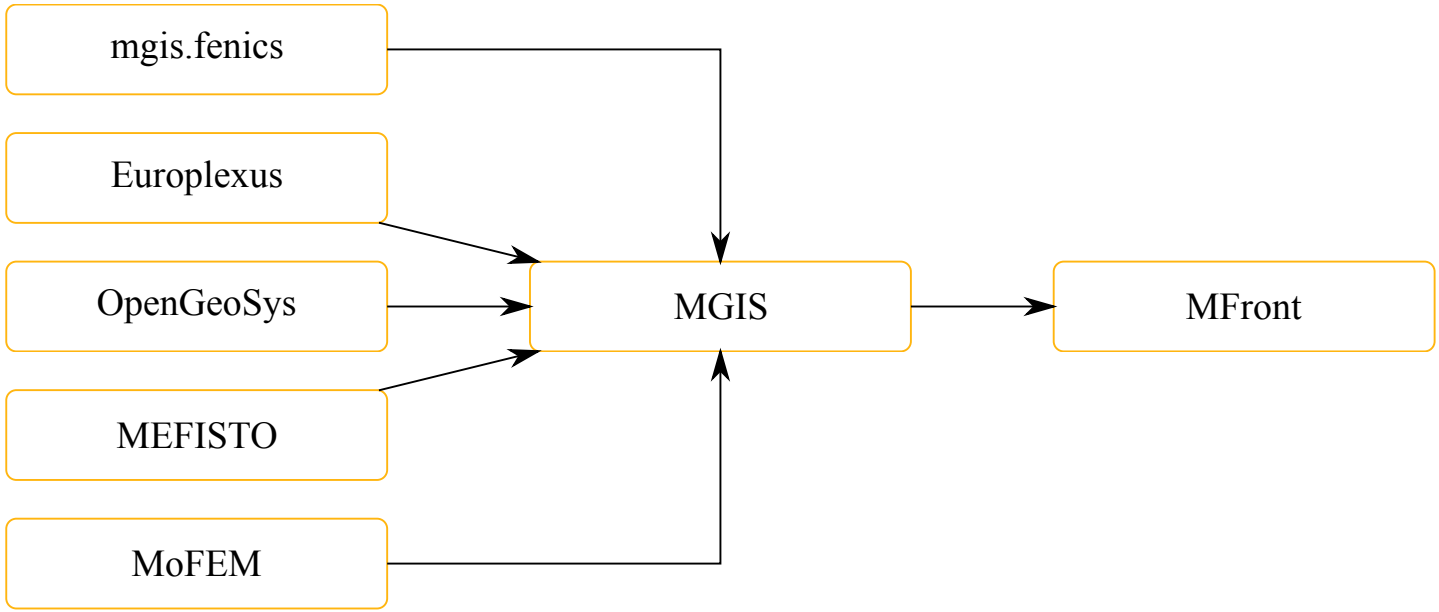
```
@Parameter strainrate A = 8.e-67;
@Parameter real E = 8.2;
@Parameter stress K = 1;
@Parameter stress R0 = 20e6;
@Parameter stress Rinf = 40e6;
@Parameter real bvp = 10;

@Integrator {
  ...
  const auto seq = sigmaeq(sig);                    // seq has the unit of a stress
  const auto iseq = 1 / max(seps, seq);             // iseq has the unit of the inverse of a stress
  const auto n = 3 * deviator(sig) * (iseq / 2);    // normal has no unit
  const auto exp_bvp = exp(-bvp * (p + theta * dp)); // exp_bvp has no unit
  Rvp = R0 + (Rinf - R0) * (1 - exp_bvp);           // Rvp has the unit of a stress
  if (seq > Rvp) {
    const auto vp = A * pow((seq - Rvp) / K, E);    // vp has the unit of a strainrate
    fp -= vp * dt;                                  // fp has the unit a a strain
    // fp -= pow((seq - Rvp) / K, E) * dt;          // This would not compile !
    // fp -= A * pow(seq - Rvp, E) * dt;            // This would not compile !
  }
  feel += dp * n;
}
```

- The `UseQt` keyword actives the use of quantities.
- Dedicated documentation of the declaration of variables in `MFront`
- Quantities are supported in DSLs associated with material properties and behaviours.

An overview of MGIS

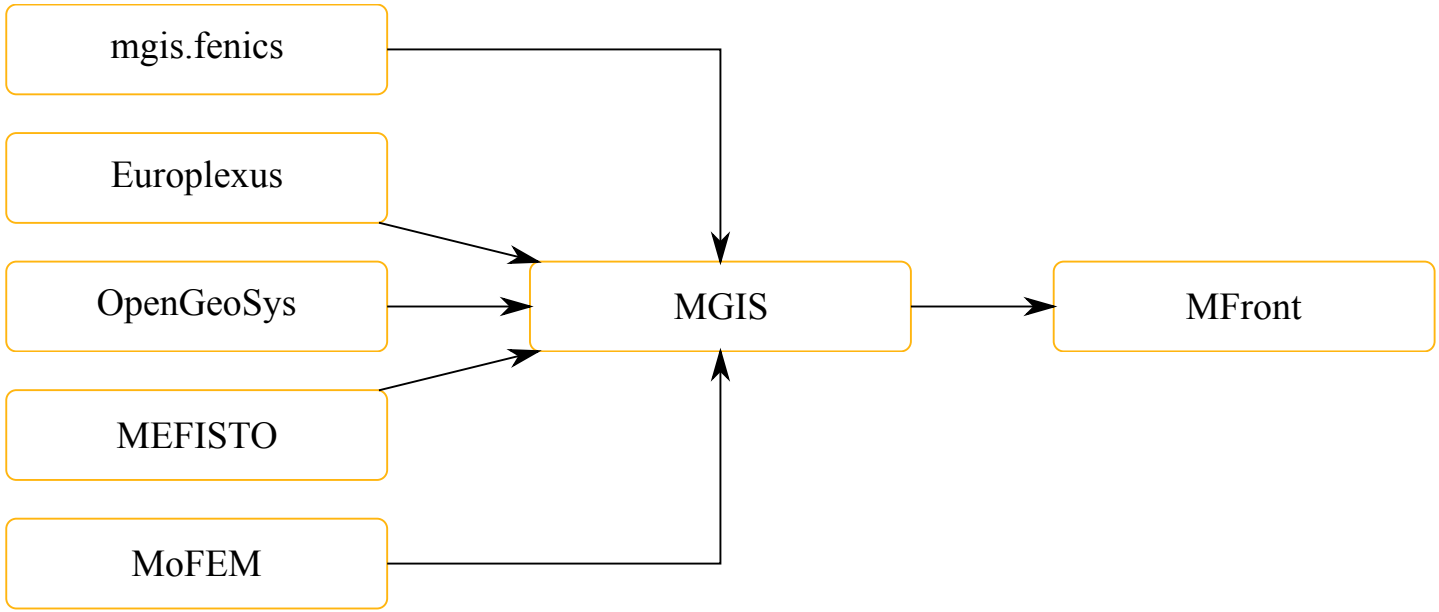
Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

Overview of the `MFrontGenericInterfaceSupport` project (MGIS)



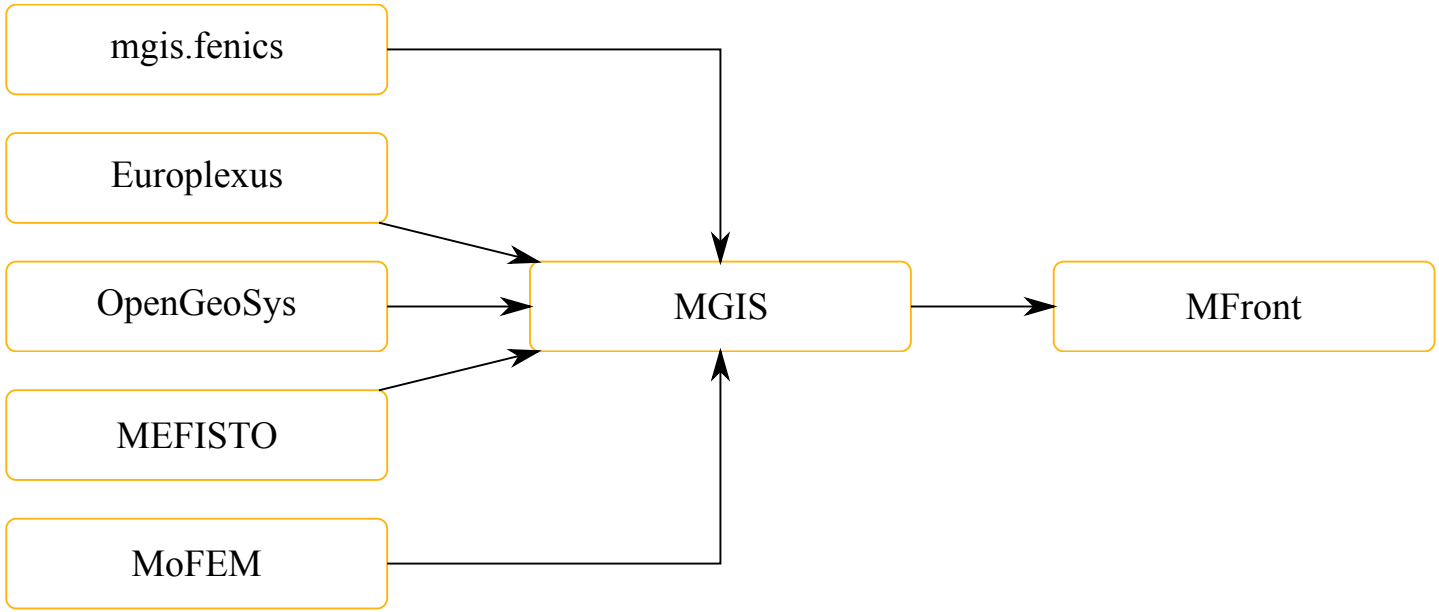
- The `MGIS` project provides classes on the solver side to retrieve **metadata** from an `MFront` behaviour and call the behaviour integration over a time step.

Overview of the `MFrontGenericInterfaceSupport` project (MGIS)



- The `MGIS` project provides classes on the solver side to retrieve **metadata** from an `MFront` behaviour and call the behaviour integration over a time step.

Overview of the `MFrontGenericInterfaceSupport` project (MGIS)

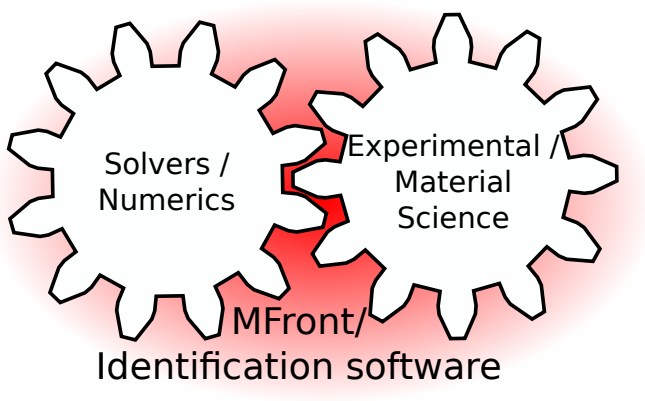


- The `MGIS` project provides classes on the solver side to retrieve **metadata** from an `MFront` behaviour and call the behaviour integration over a time step.
- Written in `C++`. Bindings exists for `C`, `Fortran2003`, `python`, `Julia`. And also used/tested in `XPer`, `Kratos Multiphysics`, `JuliaFEM`, `NairmMPM`, `esys.escript`, `DUNE`, `HELIX` (based on `MFEM`).

The MFrontGallery project

Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

Material knowledge management



- The need to guarantee the quality of engineering studies has never been so high and is constantly growing.
- Every part of a study must be covered by strict AQ procedures :
  - The finite element solver on the one hand (see the `code_aster` documentation and unit tests).
  - The material knowledge (material properties, **mechanical behaviours**) and experimental data on the other hand.
- **One must guarantee a complete consistency from experimental data to engineering studies**

MFrontGallery

- A `cmake` infrastructure to build, deploy and tests `MFront` implementations.
  - `https://github.com/thelfer/MFrontGallery`
  - `https://thelfer.github.io/MFrontGallery/web/index.html`
- Meant to serve as a basis for derived projects :
  - `https://thelfer.github.io/MFrontGallery/web/creating-derived-project.html`

Conclusions and perspectives

Current state and future works

- `MFront` is an ever improving code generation tool dedicated to material knowledge with one foot in the industrial world and one foot in the academic world.
- The development of `TFEL-4.0` has been driven by :
  - The port to the `C++-17` standard. `MFront` files will be backward-compatible.
- Future developments of `TFEL` and `MGIS` will be driven by :
  - The port to the `C++-20` standard. `MFront` files will be backward-compatible.
  - Port to GPUs (MGIS+MFront)