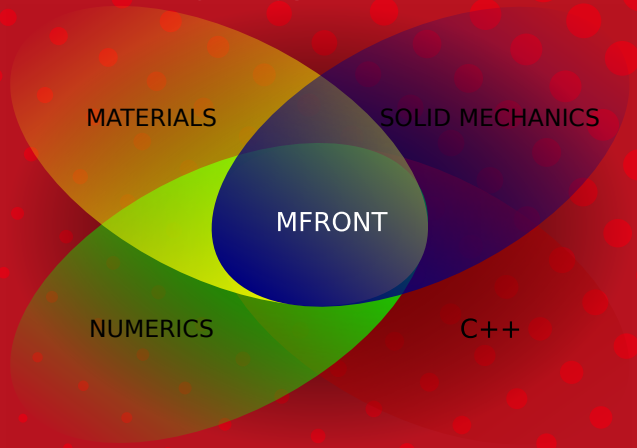
Overview of TFEL-3.4 and MGIS-1.2



DE LA RECHERCHE À L'INDUSTRIE

MFront User Meeting

25/11/2020

T. Helfer[(1)], G. Marois[(1)] and (so) many others
[(1)] CEA, DES, IRESNE, DEC, SESC, LSC, Cadarache, FranceCEA

Outline

- **A very coarse overview of `MFront` and `MGIS`**
- **Some noticeable applications of `MFront` in 2020**
- **New features in `TFEL-3.4` and `MGIS`**
- **Conclusions and perspectives**

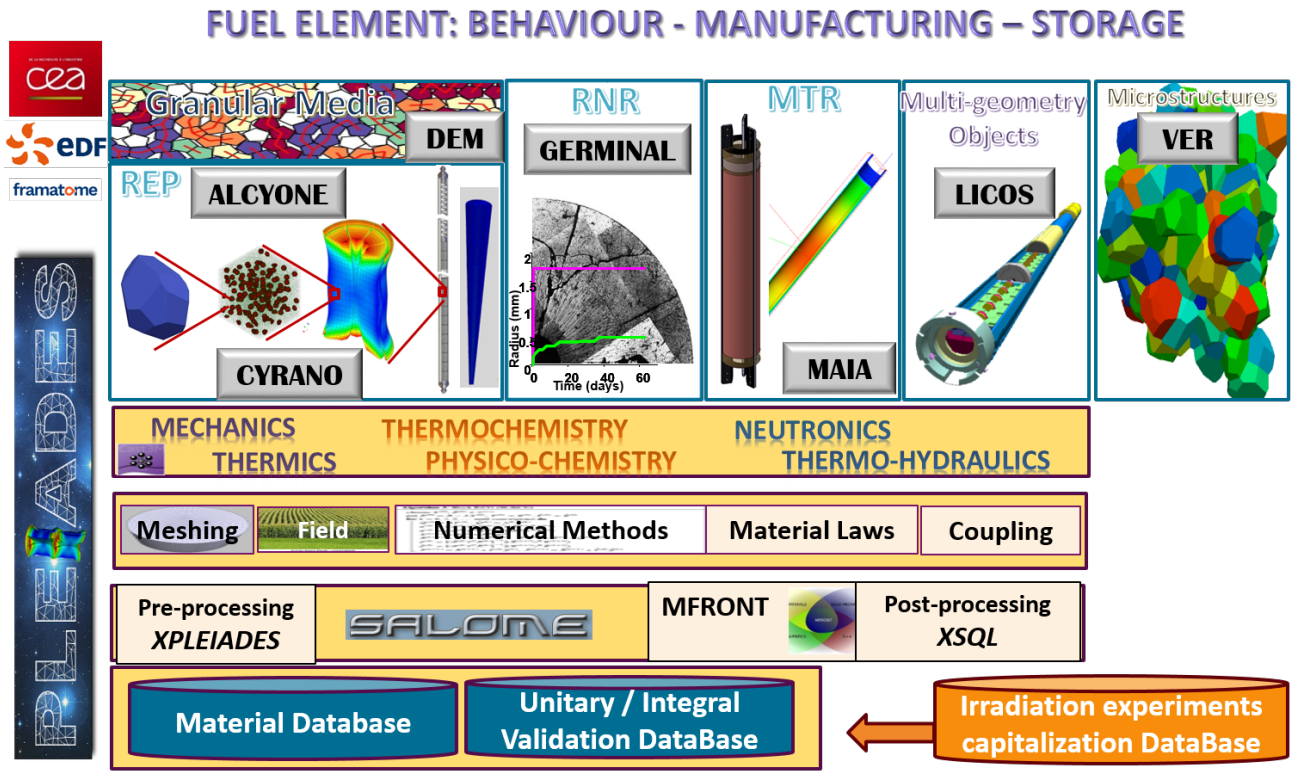
A very coarse overview of MFront and MGIS

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

Word of caution

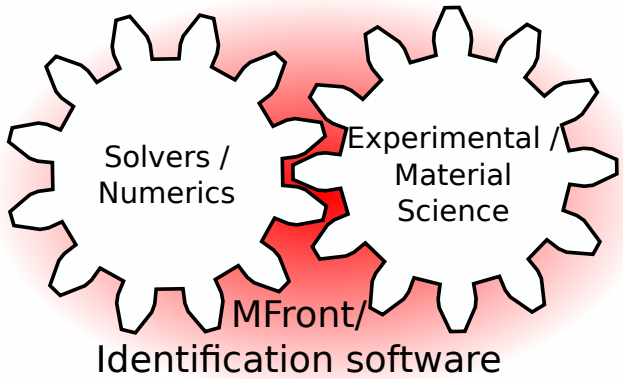
- For a more complete overview of `MFront` and `MGIS`, see :
  - Introducing the open-source mfront code generator : Application to mechanical behaviours and material knowledge management within the PLEIADES fuel element modelling platform. Computers & Mathematics with Applications. Available from : `http://www.sciencedirect.com/science/article/pii/S0898122115003132`
  - The `MFrontGenericInterfaceSupport` project. Journal of Open Source Software. 2020. Available from : `https://doi.org/10.21105/joss.02003`
  - `https://www.youtube.com/watch?v=nldf7IEtnpM`

The Pleiades platform



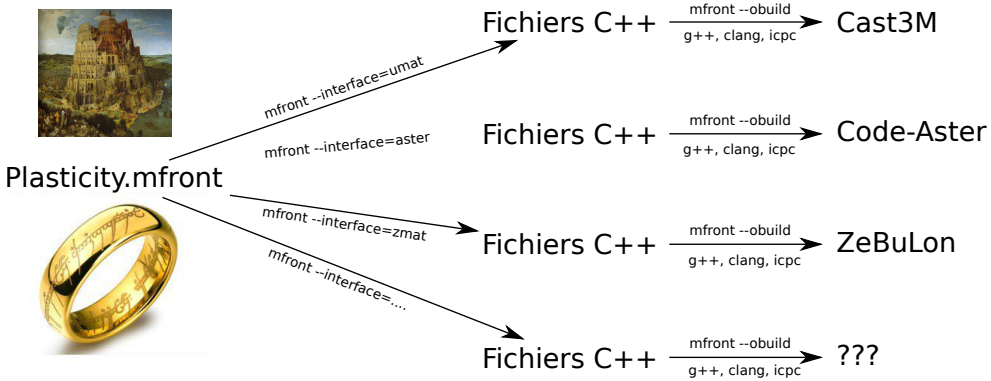
- A wide range of materials (ceramics, metals, composites).
- A wide range of mechanical phenomena and behaviours.
  - Creep, swelling, irradiation effects, phase transitions, etc..
- A wide range of mechanical loadings.

Material knowledge management



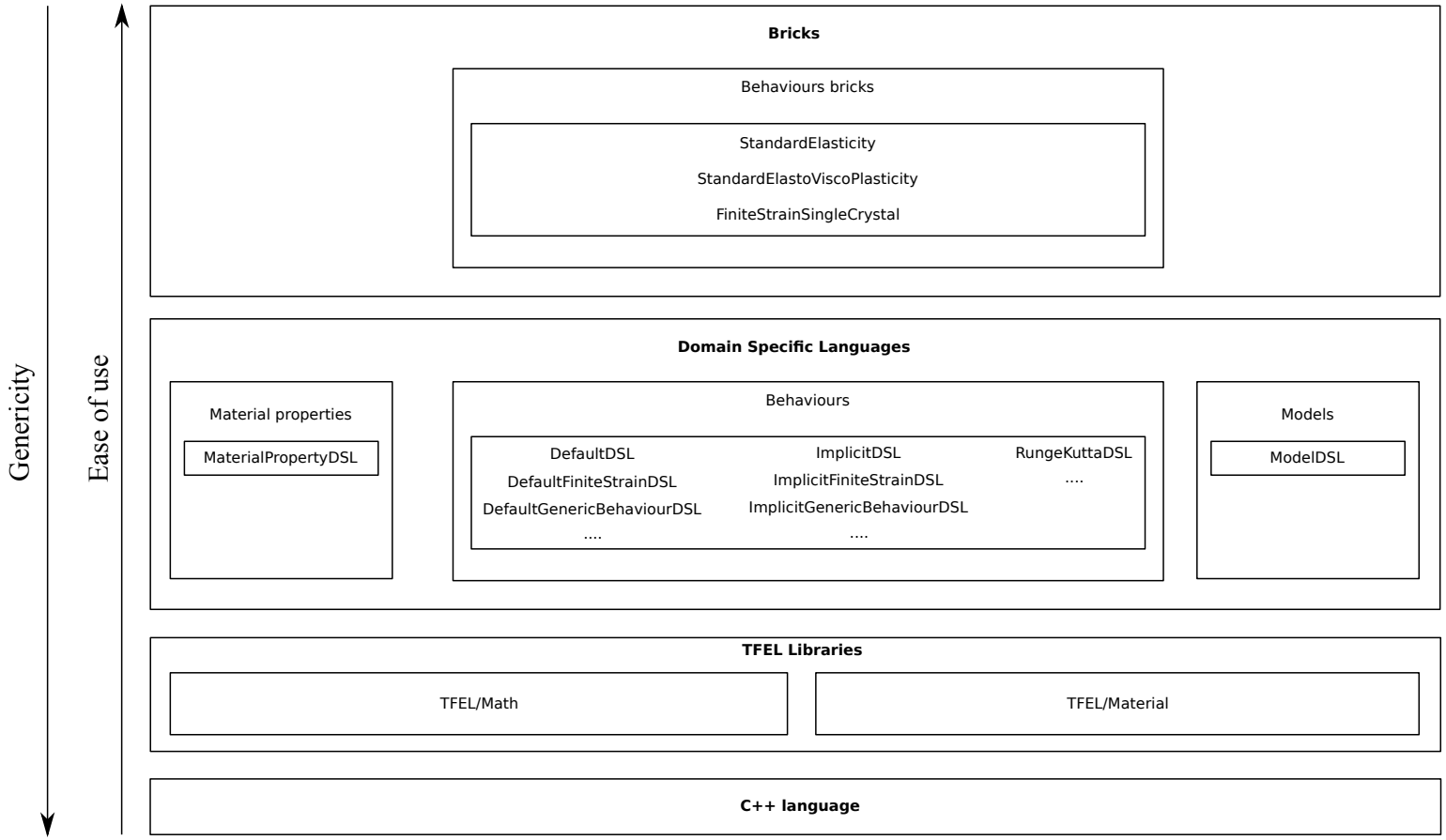
- The need to guarantee the quality of engineering studies has never been so high and is constantly growing.
- Every part of a study must be covered by strict AQ procedures :
  - The finite element solver on the one hand (see the `code_aster` documentation and unit tests).
  - The material knowledge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

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).

Layered appraoch



- Obviously, this may be disturbing for new users. Don't forget the forum and the gallery `http://tfel.sourceforge.net/gallery.html`

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 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).**

Overview of the `MFrontGenericInterfaceSupport` project (MGIS)

- The generic interface in `MFront` allows to create shared libraries usable in solvers that do not provide an existing way of adding user defined material behaviours and want to provide a more complete, efficient way to add such behaviours.

Overview of the `MFrontGenericInterfaceSupport` project (MGIS)

- The generic interface in `MFront` allows to create shared libraries usable in solvers that do not provide an existing way of adding user defined material behaviours and want to provide a more complete, efficient way to add such behaviours.
- 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.

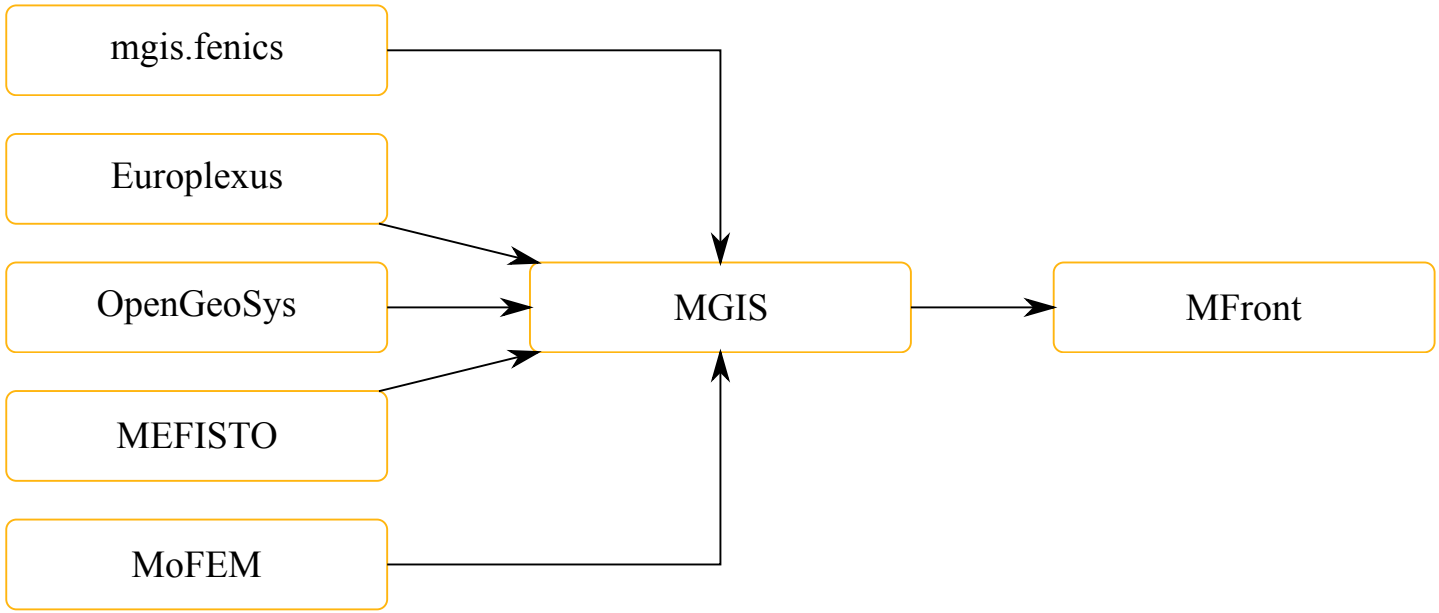
Overview of the `MFrontGenericInterfaceSupport` project (MGIS)

- The generic interface in `MFront` allows to create shared libraries usable in solvers that do not provide an existing way of adding user defined material behaviours and want to provide a more complete, efficient way to add such behaviours.
- 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.
- Two levels of integration :
  - around a single integration point
  - around a group of integration points (material)

Overview of the MFrontGenericInterfaceSupport project (MGIS)

- The generic interface in MFront allows to create shared libraries usable in solvers that do not provide an existing way of adding user defined material behaviours and want to provide a more complete, efficient way to add such behaviours.
- 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.
- Two levels of integration :
  - around a single integration point
  - around a group of integration points (material)
- Written in 'C++'. Bindings exists for 'C', 'Fortran2003', 'python', 'Julia'

Solvers using MGIS, in which MGIS has been tested

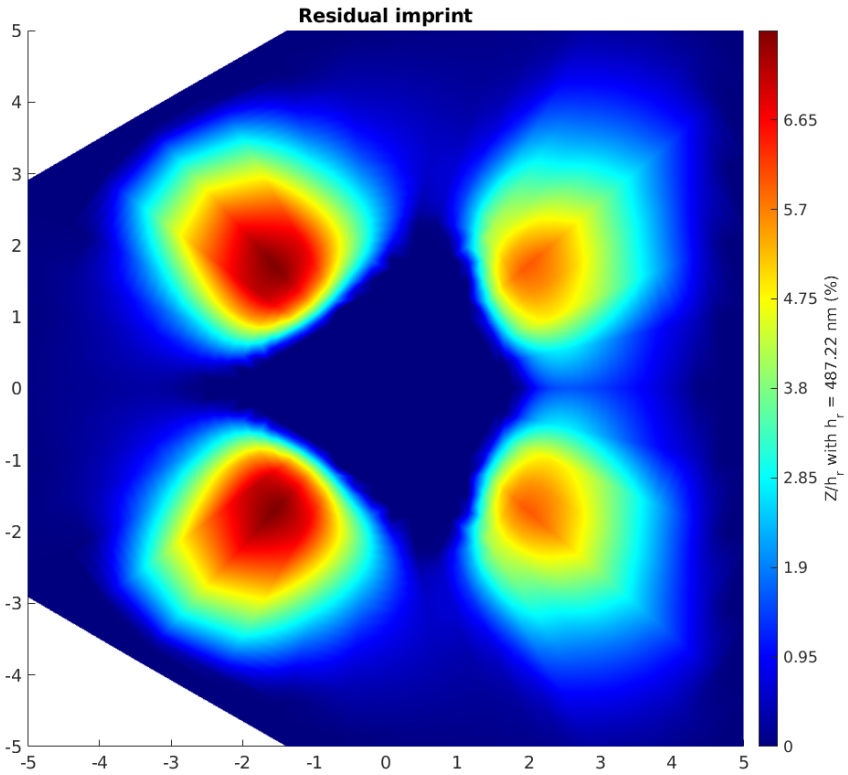


- And also XPer, Kratos Multiphysics, JuliaFEM, NairMPM, esys.escript, DUNE, HELIX (based on MFEM).
- Others are planned.

Some noticeable applications of MFront in 2020

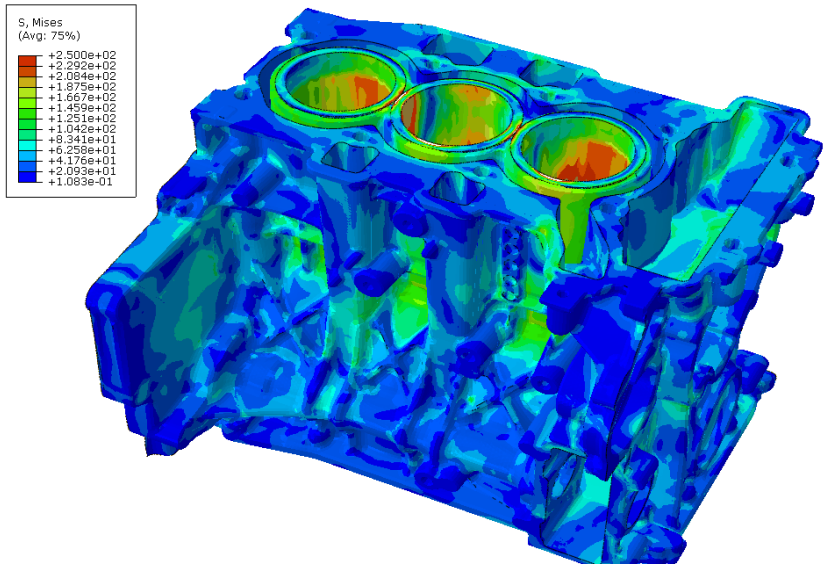
Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

Berkovich indentation on a single crystal



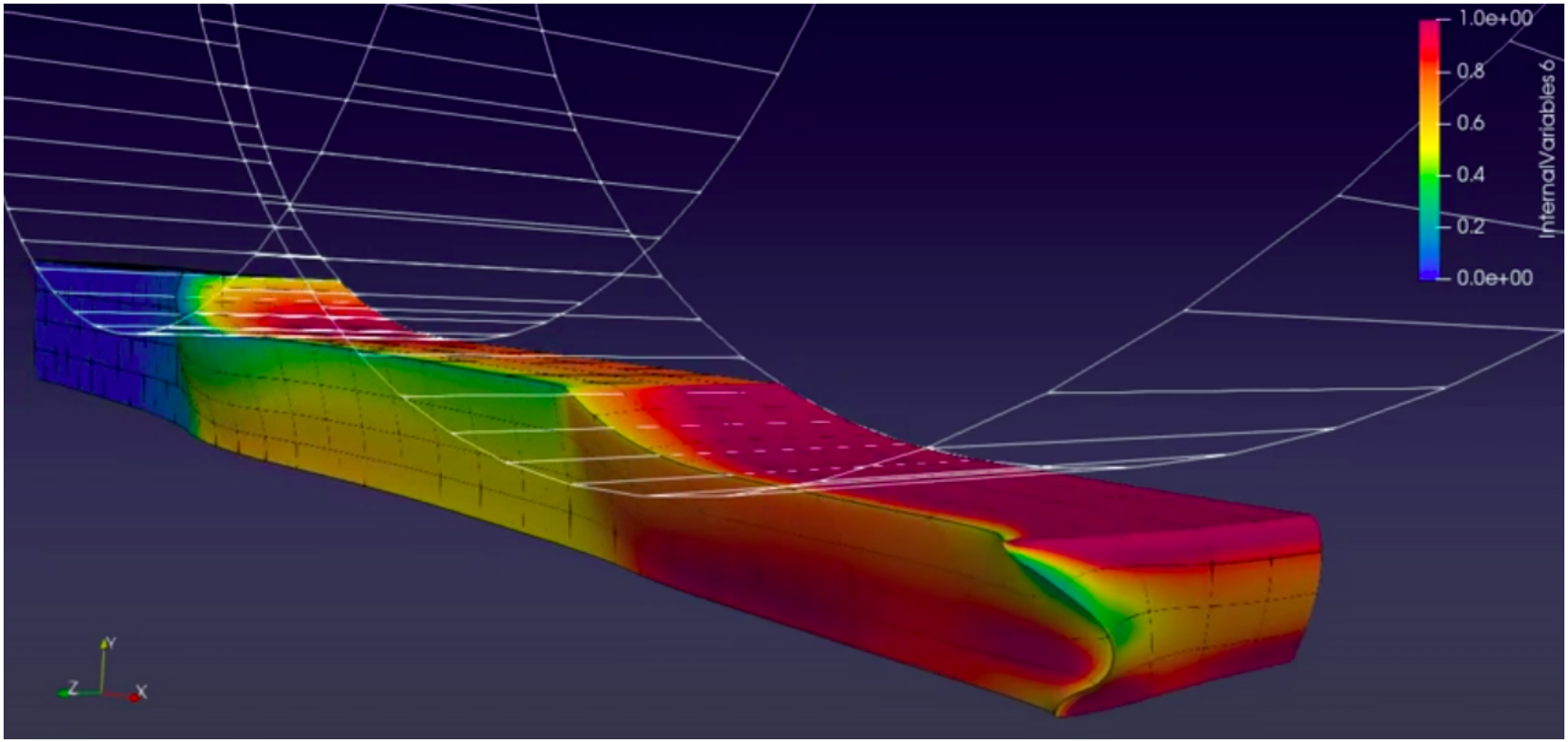
- Normalised residual topography after an indentation test on a single crsytal of copper using `Ansys`
- Courtesy of A. Bourceret, FEMTO
- The implementation of the behaviour is described here : `http://tfel.sourceforge.net/MericCailletaudSingleCrystalPlasticity.html`

Design of a cylinder block



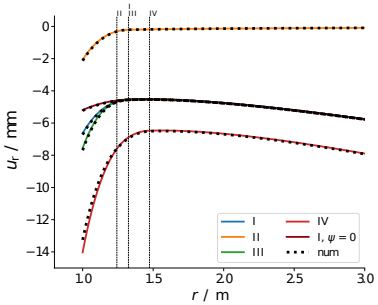
- Industrial thermomechanical design of a cylinder block with an with `MFront` and `Abaqus` at Groupe PSA.
- This study is one result of the PhD thesis of L. Jacquinot which provides a continuous modelling of the AlSi9Cu3Mg aluminium alloy behaviour from manufacturing to final usage (see the attached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

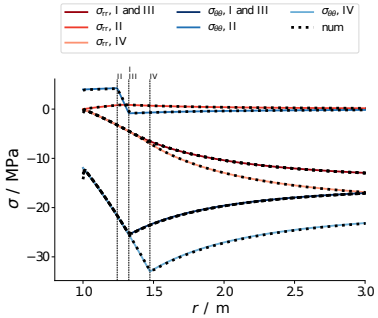
Simulation of rolling



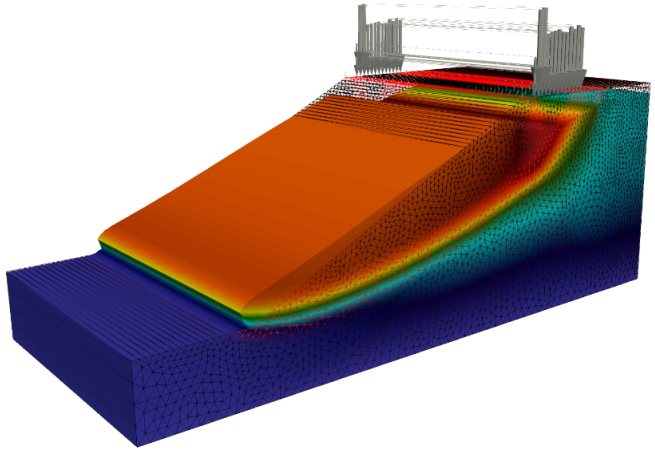
- Simulation of rolling using the innovative CEA' proto-application `MEFISTO` (implicit/explicit solver)
- Courtesy of O. Jamond, CEA

Slope failure analysis



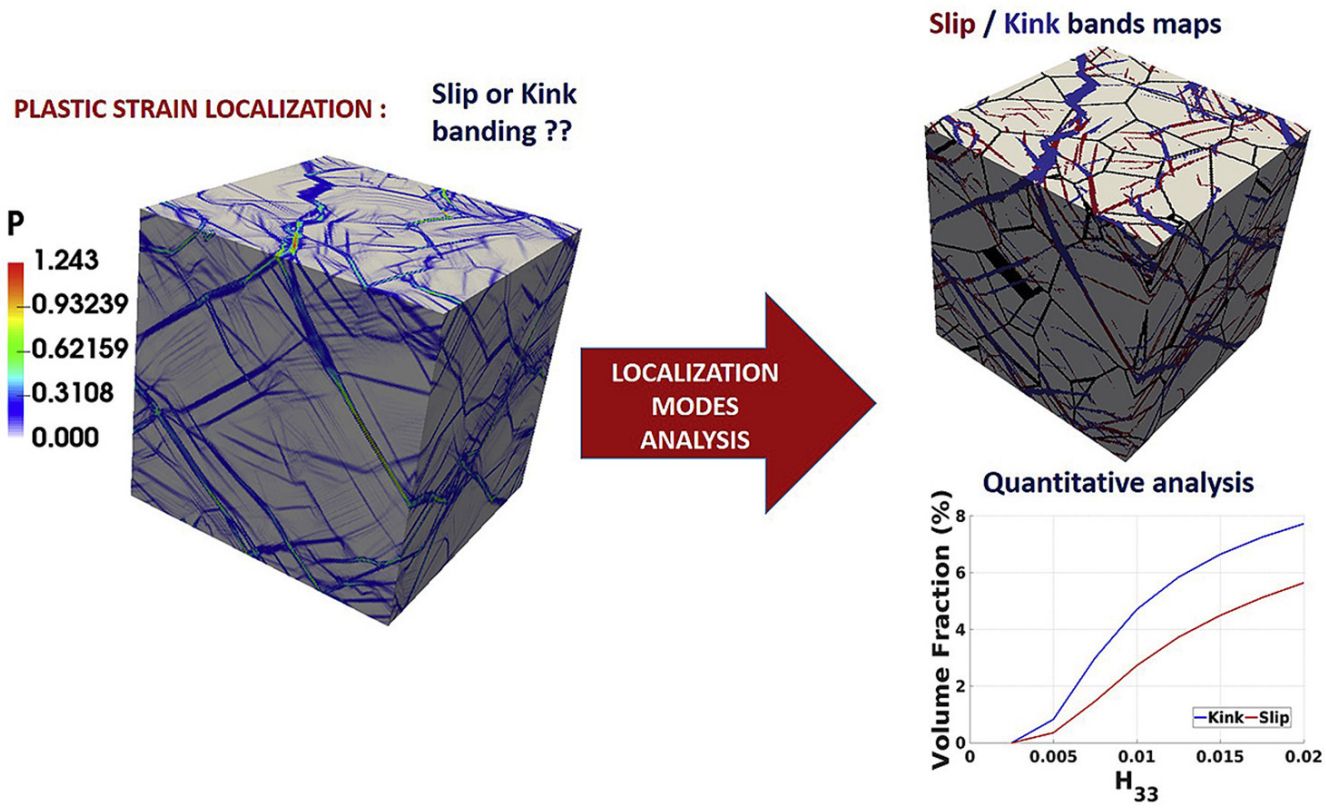


Slope failure analysis with phi-c-reduction in OpenGeoSys



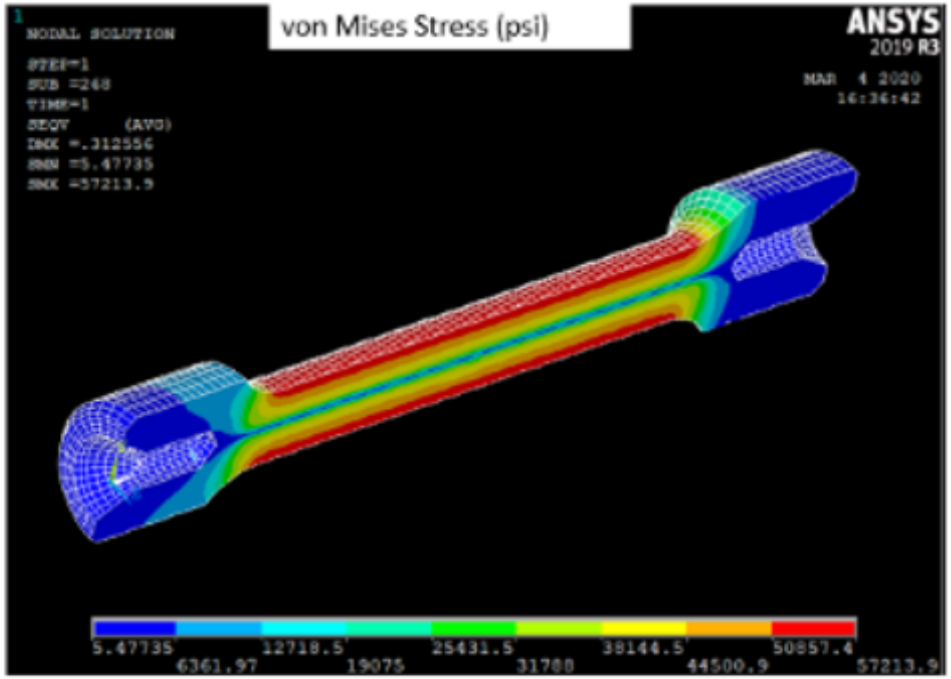
- Analytical verification (left)
- Slope failure analysis with strength reduction in `OpenGeoSys` by T. Deng and T. Nagel (Technische Universität Bergakademie Freiberg)
- For details, see `https://opengeosys.org/docs/benchmarks/small-deformations/slope_stability.pdf`.
- The implementation of the behaviour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

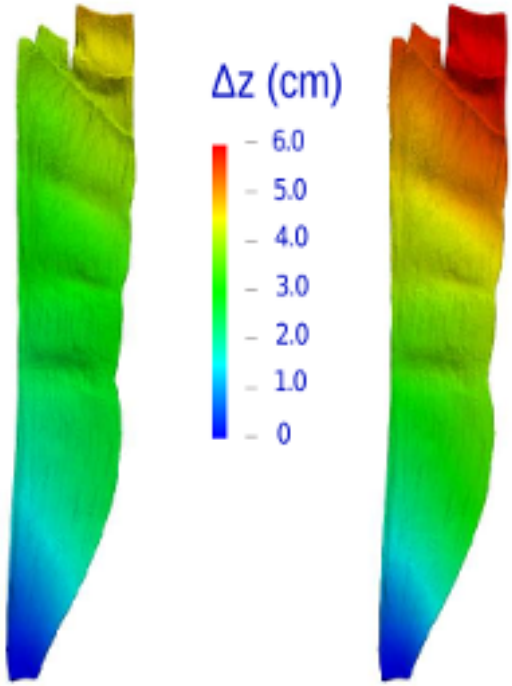
Torsional twist of a bar



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`

New features in TFEL-3.4 and MGIS

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

New features of TFEL-3.4

- http://tfel.sourceforge.net/release-notes-3.4.html
  - Exhaustive list of new features and bug fixes.
- Extended features for the computations of the consistent tangent operator in implicit scheme.
- Extensions of the StandardElastoViscoPlasticity brick.
- Support of the Madnex file format :
  - https://github.com/thelfer/madnex
- Integration of MTest in an ICOCO component :
  - Development prior to the use of MFront behaviours in CATHARE3.

Extended features for the computations of the consistent tangent operator in implicit scheme

```
@TangentOperatorBlock{
  ∂fε^el/∂ΔT = ...;
  ∂fp/∂ΔT = ...;
  auto ∂Δε^el/∂ΔT = Stensor{};
  getIntegrationVariablesDerivatives_T(∂Δε^el/∂ΔT);
  const auto λ =...;
  const auto μ =...;
  const auto ∂λ/∂T =...;
  const auto ∂μ/∂T =...;
  const auto De = λ  · (I₂ ⊗ I₂) + 2 · μ · I₄;
  const auto ∂De/∂T = ∂λ/∂T  · (I₂ ⊗ I₂) + 2 · ∂μ/∂T · I₄;
  ∂σ/∂ΔT = De · ∂Δε^el/∂ΔT + ∂De/∂T · ε^el;
}
```

```
@TangentOperatorBlock{
  const auto λ =...;
  const auto μ =...;
  const auto De = λ  · (I₂ ⊗ I₂) + 2 · μ · I₄;
  // computation of ∂σ/∂ΔT
  ∂fε^el/∂ΔT = ...;
  ∂fp/∂ΔT = ...;
  const auto ∂Δε^el/∂ΔT = -(iJ_fε^el_ε^el · ∂fε^el/∂ΔT + iJ_fε^el_p · ∂fp/∂ΔT);
  const auto ∂λ/∂T =...;
  const auto ∂μ/∂T =...;
  const auto ∂De/∂T = ∂λ/∂T  · (I₂ ⊗ I₂) + 2 · ∂μ/∂T · I₄;
  ∂σ/∂ΔT = De · ∂Δε^el/∂ΔT + ∂De/∂T · ε^el;
}
```

- $\dfrac{\partial \vec{\Sigma}}{\partial \Delta g} = \dfrac{\partial \vec{\Sigma}}{\partial \vec{Y}\Big|_{t+\Delta t}} \cdot \dfrac{\partial \vec{Y}\Big|_{t+\Delta t}}{\partial \Delta g}$
- $F\left(\vec{Y}\Big|_{t+\Delta t}\right) = 0 \Rightarrow \dfrac{\partial \vec{Y}\Big|_{t+\Delta t}}{\partial \Delta g} = J^{-1} \cdot \dfrac{\partial F}{\partial \Delta g}$
- Two new methods :
  - Using the `getIntegrationVariablesDerivatives_*` functions.
  - Using the blocks of the invert of the jacobian matrix

Extensions of the `StandardElastoViscoPlasticity` brick

- Extension to porous plasticity (see the talk of M. Shokeir et al.)
- New inelastic flow `HarmonicSumOfNortonHoffViscoplasticFlows`. The equivalent viscoplastic strain rate $\dot{p}$ is defined as :

$$\frac{1}{\dot{p}} = \sum_{i=1}^{N} \frac{1}{\dot{p}_i}$$

  where $\dot{p}_i$ has an expression similar to the the Norton-Hoff viscoplastic flow :

$$\dot{p}_i = A_i \left(\frac{\sigma_{eq}}{K_i} \right)^{n_i}$$

- New tutorial on how to introduce a new stress criterion : `https://www.researchgate.net/publication/340280305_Extending_the_StandardElastoViscoPlasticity_brick_with_a_new_stress_criterion`
- A second tutorial on how to introduce a new porous stress criterion is to be published soon.

New features of MGIS

- https://thelfer.github.io/mgis/web/release-notes-1.2.html
- Support of orthotropic behaviours (based on new developments of the generic behaviour).
- The mgis.fenics python module by Jérémy Bleyer :
  - https://thelfer.github.io/mgis/web/mgis_fenics.html
  - Various tutorials are already written which illustrates both mgis.fenics and some advanced features of MFront (generic behaviours).
  - See Jérémy Bleyer's talk.

Conclusions and perspectives

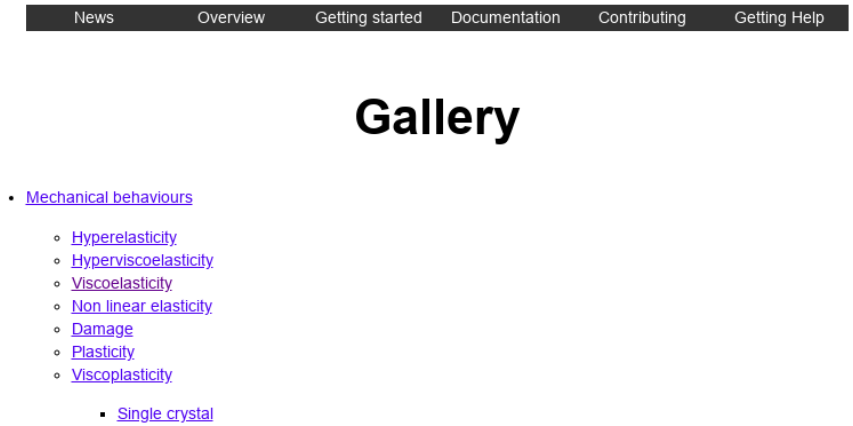
Acknowledgements

- Since its release in open-source, the community of users of `MFront` keeps increasing.
- We are in debt to all the contributors including :
  - direct contributors (new developments, new pages of tutorials, feed-backs, proposal of improvements, benchmarks, etc.)
  - indirect contributors (package managers, developers of solvers, project managers, etc..)

Current state and future works

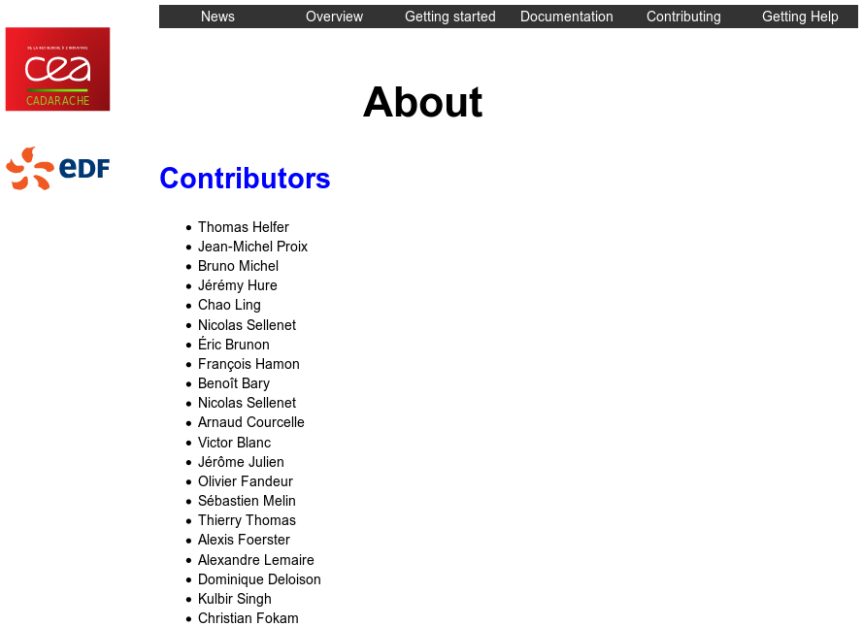
- `MFront` is an ever improving code generation tool dedicated to material knowledge with one foot in the industrial world and one foot in the academic world.
- The development of `TFEL-3.4` (this year version) has been geared around three main axes :
  - Generalised behaviours
  - Porous plasticity
  - Extension and implementation of the `Madnex` specifications for the storage of `MFront` files.
- The development of `TFEL-3.4.1` is meant to finalize some developments and requested features.
  - Versions `TFEL-3.4.x` will provide stable upgrades to `TFEL-3.4`
  - Versions `TFEL-3.4.x` will not be limited to bug fixes
- The development of `TFEL-4.0` (next year) will be driven by :
  - The port to the `C++-17` standard. `MFront` files will be backward-compatible.
  - Homogeneisation
  - Data driven simulation?

Conclusions and advices for new users



- This talk is just an *introduction* to `MFront`.
- The `MFront` gallery is good way to start :
  `http://tfel.sourceforge.net/gallery.html`
- When starting with `MFront`, your mileage may vary :
  - Some do rapidly very complex things on their own.
  - Some find the learning curve abrupt. Don't hesitate to ask for help !
  - The best way to start is to write a new entry in the `MFront` gallery.
  - From time to time, a 2-days formation is organized. Just ask !

How to contribute



- Citations and illustrations
- Feed-backs, feed-backs, and feed-backs !
  - Please use the forum.
  - Enhancement suggestions (code, documentation, algorithm, etc...)
- Submit new behaviours implementation and tests.
- Submit pages to the gallery.
- Code (for the braves)

Thank you for your attention.
Time for discussion!

https://tfel.sourceforge.net
https://www.researchgate.net/project/TFEL-MFront
https://twitter.com/TFEL_MFront
https://github.com/thelfer/

tfel-contact@cea.fr

The development of MFront is supported financially by CEA, EDF and Framatome in the framework of the PLEIADES project.