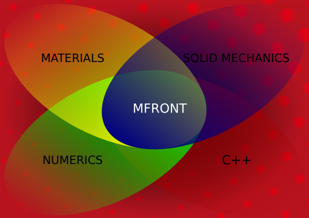
mgis.fenics: coupling MFront and FEniCS for complex solid mechanics simulations

DE LA RECHERCHE À L'INDUSTRIE



FEniCS conference 2021 - 22-26 March 2021

T. Helfer(1), J. Bleyer(2), R. Russo(3) T. Dancheva(4)
(1) CEA, DES, IRESNE, DEC, SESC, LSC, Cadarache, France
(2) Laboratoire Navier UMR 8205 (École des Ponts ParisTech-IFSTTAR-CNRS), France
(3) University of the Basque Country
(4) BCAM - Basque Center for Applied Mathematics

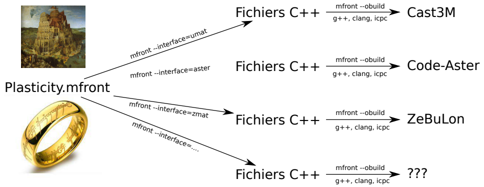
Goals of the `mgis.fenics` project

- `mgis.fenics` aims at combining the power of :
  - `FEniCS` for automatic generation of assembly of user defined weak forms in the `UFL` syntax, the HPC performances, etc...

$$R(\mathbf{v}) = \sum_{i=1}^{p} \int_{\Omega} \boldsymbol{\sigma}^{i}(\mathbf{u}) \cdot \delta \mathbf{g}^{i}(\mathbf{v}) \, \mathrm{dx} - L(\mathbf{v}) = 0 \quad \forall \mathbf{v} \in V$$

  - `MFront` for the local description of the material behaviour :
    - Complex kernels at quadrature points to compute $\boldsymbol{\sigma}^{i}(\mathbf{u})$.
    - See also the talks about the `AceGEN` and `Materiaux` projects.
  - The `MGIS` project that provides classes to :
    - Retrieve **metadata** from an `MFront` behaviour
    - Allocate memory associated with the **state variables** handled by the behaviour
    - Call the behaviour integration over a time step.
    - `https://github.com/thelfer/MFrontGenericInterfaceSupport`

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 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).**

Generalities about `mgis.fenics`

- Inside a custom `NonlinearProblem`, define $\boldsymbol{\sigma}^i$ on a `Quadrature` space and the generalized residual :

$$R(\boldsymbol{v}) = \sum_{i=1}^{p} \int_{\Omega} \boldsymbol{\sigma}^i(\boldsymbol{u}) \cdot \delta \mathbf{g}^i(\boldsymbol{v}) \,\mathrm{dx} - L(\boldsymbol{v}) = 0 \quad \forall \boldsymbol{v} \in V$$

- `MGIS` gives metadata to know on which blocks $\mathcal{B}(i)$ of gradients each flux $\boldsymbol{\sigma}_i$ depends :

$$a_{\mathrm{tangent}}(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i=1}^{p} \sum_{j \in \mathcal{B}(i)} \int_{\Omega} \delta \mathbf{g}^i(\boldsymbol{v}) \cdot \mathbb{T}^{\boldsymbol{\sigma}^i}_{\mathbf{g}^j} \cdot \delta \mathbf{g}^j(\boldsymbol{u}) \,\mathrm{dx}$$

- **This default variational problem can be overloaded by the user using `UFL`.**
- **mgis.fenics** almost automatically make the links between `FEniCS` and `MFront`.

List of available tutorials

- Documented demos have been designed to progressively illustrate the use of the interface and the versatility of the approach when implementing complex generalized behaviours, both on the MFront and FEniCS sides.
- We recommend browsing the demos in the following order :
  - Stationnary non-linear heat transfer
  - Stationnary non-linear heat transfer : 3D problem and performance comparisons
  - Transient heat equation with phase change
  - Monolithic transient thermoelasticity
  - Small-strain von Mises elastoplasticity
  - **Finite-strain elastoplasticity within the logarithmic strain framework**
  - Multiphase model for fiber-reinforced materials
  - **Phase-field approach to brittle fracture**
- Repository : a repository containing the demos sources files is available `https://gitlab.enpc.fr/navier-fenics/mgis-fenics-demos`

Finite-strain plasticity in the logarithimic space

```
@DSL Implicit;

@Behaviour LogarithmicStrainPlasticity;

@StrainMeasure Hencky;

@Brick StandardElastoViscoPlasticity{
  stress_potential : "Hooke" {
        young_modulus : 210e9,
        poisson_ratio : 0.3
        },
  inelastic_flow : "Plastic" {
    criterion : "Mises",
    isotropic_hardening : "Linear" {H : 500e6,
                                     R0 : 250e6}
  }
};
```

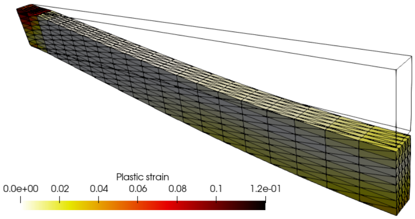
```
material = mf.MFrontNonlinearMaterial("./src/libBehaviour.so", "
        LogarithmicStrainPlasticity")
problem = mf.MFrontNonlinearProblem(u, material, bcs=bc)
problem.set_loading(dot(selfweight, u)∗dx)

prm = problem.solver.parameters
prm["absolute_tolerance"] = 1e−6
prm["relative_tolerance"] = 1e−6
prm["linear_solver"] = "mumps"

for (i, t) in enumerate(load_steps[1:]):
    selfweight.t = t
    problem.solve(u.vector())
```

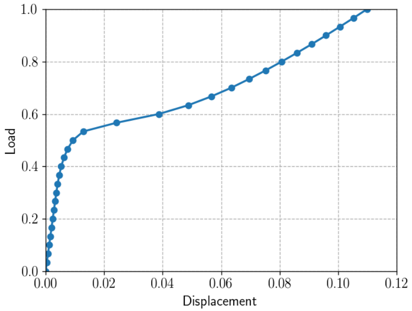
Automatic registration of
as I + (grad(Displacement)

- Residual is : $R(\boldsymbol{v}) = \int_\Omega \boldsymbol{P} : \delta\boldsymbol{R}(\boldsymbol{v})\,\mathrm{dx} - W_{ext}(\boldsymbol{v})$
- Consistent tangent bilinear form is : $a_{\text{tangent}}(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \nabla\boldsymbol{u} : \dfrac{\partial\boldsymbol{P}}{\partial\boldsymbol{F}} : \nabla\boldsymbol{v} dx$
- Logarithmic strain plasticity (Miehe, 2002) :
  - Hencky strain measure $\boldsymbol{H} = \dfrac{1}{2}\log(\boldsymbol{F}^{\mathrm{T}} \cdot \boldsymbol{F})$ ✗
  - Use a small strain constitutive relation on $\boldsymbol{H} \Rightarrow \boldsymbol{H} = \boldsymbol{H}^e + \boldsymbol{H}^p$

$$\boldsymbol{F} \longrightarrow \boldsymbol{H} \longrightarrow \boxed{\text{small strain law}} \longrightarrow \boldsymbol{T} \longrightarrow \boldsymbol{\sigma}$$

Logarithmic strain plasticity - results





- https://thelfer.github.io/mgis/web/mgis_fenics_finite_strain_elastoplasticity.html
- https://gitlab.enpc.fr/navier-fenics/mgis-fenics-demos/-/tree/master/demos/finite_strain_elastoplasticity

Phase-field approach to brittle fracture

- Bourdin/Francfort/Marigo **variational phase-field approach** :

$$\boldsymbol{u}(t), d(t) = \arg\min_{\boldsymbol{u},d} \int_\Omega (1-d)^2\psi^+(\varepsilon) + \psi^-(\varepsilon)\,\mathrm{dx} - W_{ext}(\boldsymbol{u}) + \frac{G_c}{c_w}\int_\Omega \left(\frac{w(d)}{\ell_0} + \ell_0\|\nabla d\|^2\right)\,\mathrm{dx}$$

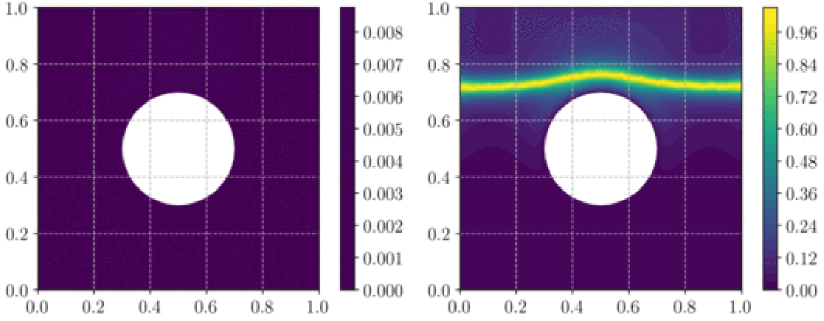
- Example of Tension/compression splitting (Miehe et al.) :

$$\psi^+(\varepsilon) = \frac{1}{2}\lambda\langle \mathrm{tr}(\varepsilon)\rangle_+^2 + \mu\sum_I \langle\varepsilon_I\rangle_+^2$$

- Implementation of the alternate minimisation algorithm :

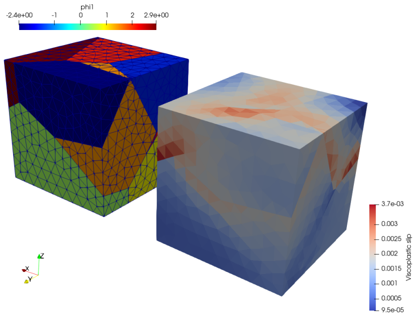
```
problem_u = mf.MFrontNonlinearProblem(u, material_u, bcs=bcu)
problem_u.register_external_state_variable("Damage", d)
psi  = problem_u.get_state_variable("PositiveEnergyDensity")

for (i, t) in enumerate(loading[1:]):
    Uimp.t = t
    while res > tol and j < Nitermax:
        problem_u.solve(u.vector()) # Solve displacement u—problem
        problem_d.solve(d.vector()) # Solve damage d—problem
```

Phase-field approach to brittle fracture



- Classical example of crack propagation (Bourdin et al.)

Polycrystal computations



- Finite strain implementation of the Méric-Cailletaud single crystal behaviour :
  - `http://tfel.sourceforge.net/MericCailletaudSingleCrystalPlasticity.html`
- Example of orthotropic behaviour support.
- Périodic boundary conditions.

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-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-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 ?
  - Support of GPUs ?

Conclusions on `MGIS` and `mgis.fenics`

- `MGIS` is a young project with many interesting perspectives :
  - Support of GPU ? Support of `Eigen` ?
- What's next in `mgis.fenics` ?
  - Tests !
  - Integration in **dolfin-x** and performances improvements.
  - Multi-materials, plates/shells
  - Other examples :
    - Cosserat elastoplasticity (see part II).
    - Micromorphic crystal plasticity.
- New users and contributions are welcomed !

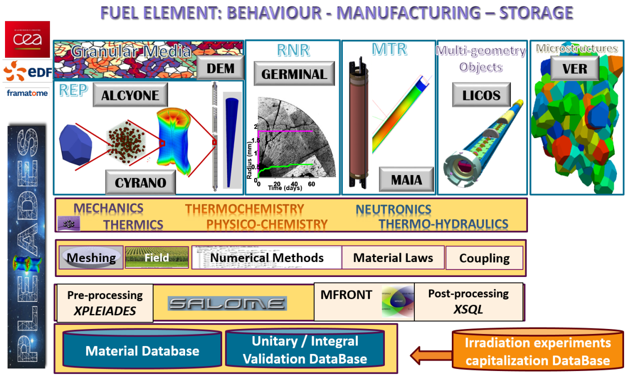
Thank you for your attention. Time for discussion !

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

tfel-contact@cea.fr

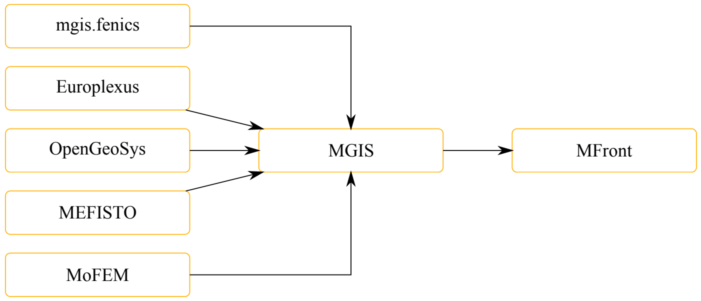
The development of `MFront` is supported financially by CEA, EDF and Framatome in the framework of the `PLEIADES` project.

The Pleiades platform



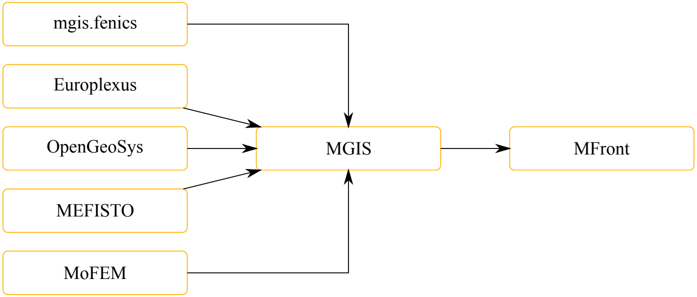
- A wide range of materials (ceramics, metals, composites).
- A wide range of mechanical phenomena and behaviours.
  - Creep, swelling, irradiation effects, phase transitions, etc..
- A wide range of mechanical loadings.

Overview of the MFrontGenericInterfaceSupport project (MGIS)



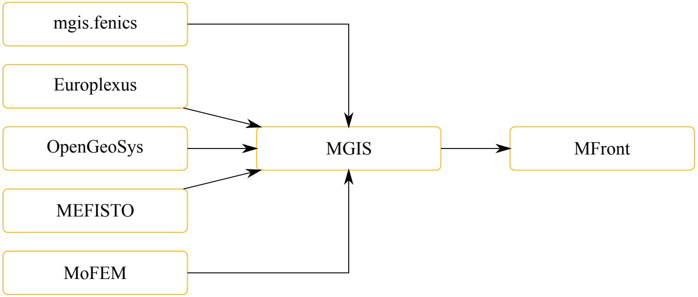
- The MGIS project provides classes on the solver side to retrieve **metadata** from an MFront behaviour and call the behaviour integration over a time step.

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).