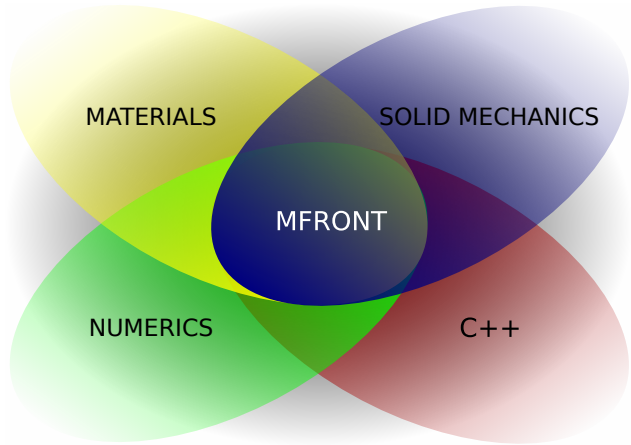
DE LA RECHERCHE À L'INDUSTRIE

cea

Fast, Robust and Portable Implementations of Complex Mechanical Behaviours with the MFront Code Generator



13th World Congress in Computational Mechanics | THOMAS HELFER

JULY 2018

www.cea.fr

Context

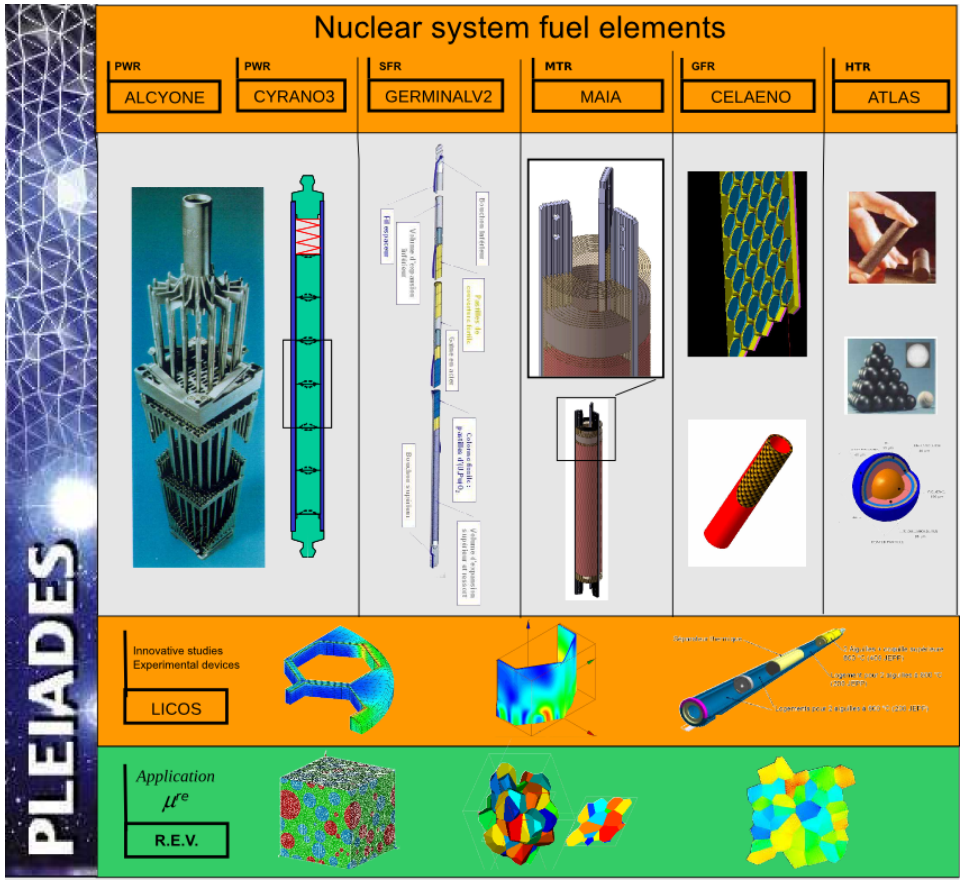
DE LA RECHERCHE À L'INDUSTRIE
cea

Outline

- Context
- Material knowledge management in nuclear simulations
- The `TFEL` project and the `MFront` code generator
- Mechanical behaviours
- Performances
- The StandardElastoViscoPlasticity brick
- Conclusions

Material knowledge management in nuclear simulations

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**

The TFEL project and the MFront code generator

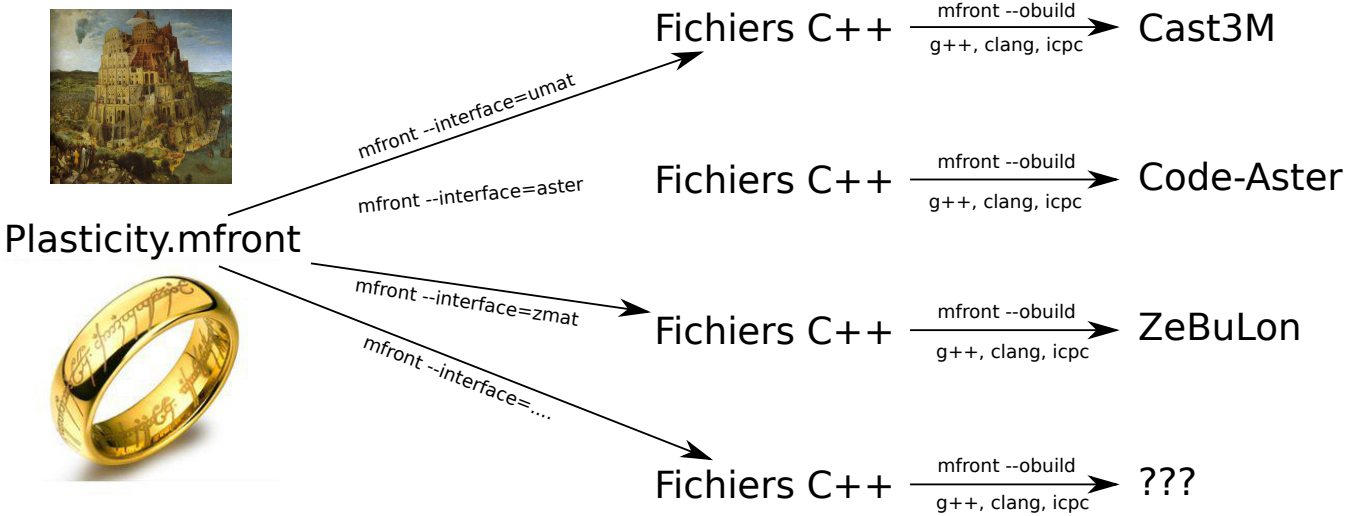
Definitions

- **State variables** : the (thermodynamical) state of a material is assumed to be fulled characterized by a set of `state variables` defined at each point of the material.
- **Model** : description of how the state variables of a set of materials evolves du to one or several physical phenomena :
  - Point-wise models
  - Equilibrium based models (heat transfer, mechanics, etc...)
- **Behaviours** : in equilibrium based models, a specific material is described by a behaviour which relates a gradient to the associated flux and makes the state variables evolves.
- **Material properties** : functions of the current state variables of the material used by some generic models or behaviours to describe a specific material.

Definitions

- **State variables** : the (thermodynamical) state of a material is assumed to be fulled characterized by a set of `state variables` defined at each point of the material.
- **Model** : description of how the state variables of a set of materials evolves du to one or several physical phenomena :
  - Point-wise models
  - Equilibrium based models (heat transfer, mechanics, etc...)
- **Behaviours** : in equilibrium based models, a specific material is described by a behaviour which relates a gradient to the associated flux and makes the state variables evolves.
- **Material properties** : functions of the current state variables of the material used by some generic models or behaviours to describe a specific material.

MFront goals



- `TFEL` is a project various libraries/softwares, including `MFront`.
- `MFront` is a code generation tool dedicated to material knowledge (material properties, mechanical behaviours, point-wise models) with focus on :
  - Numerical efficiency (see various benchmarks).
  - 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).

Timeline : more than 10 years of development

- 2006 : first line of codes :
  - Mainly focused on the `Cast3M` finite solver.
- 2009 : officially part of the PLEIADES project :
  - Development of the `Cyrano` interface.
- 2013 : first contact with the `Code_Aster` team.
- 2014 : TFEL-2.0 is released as an open-source project :
  - Officially co-developped by CEA and ÉDF.
  - Implementing an interface to `ZMAT`.
- 2015 : Officially part of `Code_Aster/Salomé-Méca`.
- 2016 : TFEL-3.0 is released :
  - Support of `Europlexus`, `Abaqus/Standard`, `Abaqus/Explicit`
  - `StandardElasticity` brick
  - $\approx$ 250 000 lines of codes
- 2017 : TFEL-3.1 is released :
  - `CalculiX` native interface, experimental support of `Ansys APDL`
- 2018 : TFEL-3.2 will be released :
  - Full support of `Ansys` and `LS-DYNA`, generic 'MFront' behaviour.
  - The `StandardElastoViscoplasticity` brick

Licences : Open-source (again) !

- To meet CEA and EDF needs, TFEL 2.0 is released under a multi-licensing scheme :
  - Open-source licences :
    - GNU Public License : This licence is used by the Code_Aster finite element solver.
    - CECILL-A : License developped by CEA, EDF and INRIA, compatible with the GNU Public License and designed for conformity with the French law.
  - CEA and EDF are free to distribute TFEL under custom licences : Mandatory for the PLEIADES plateform.

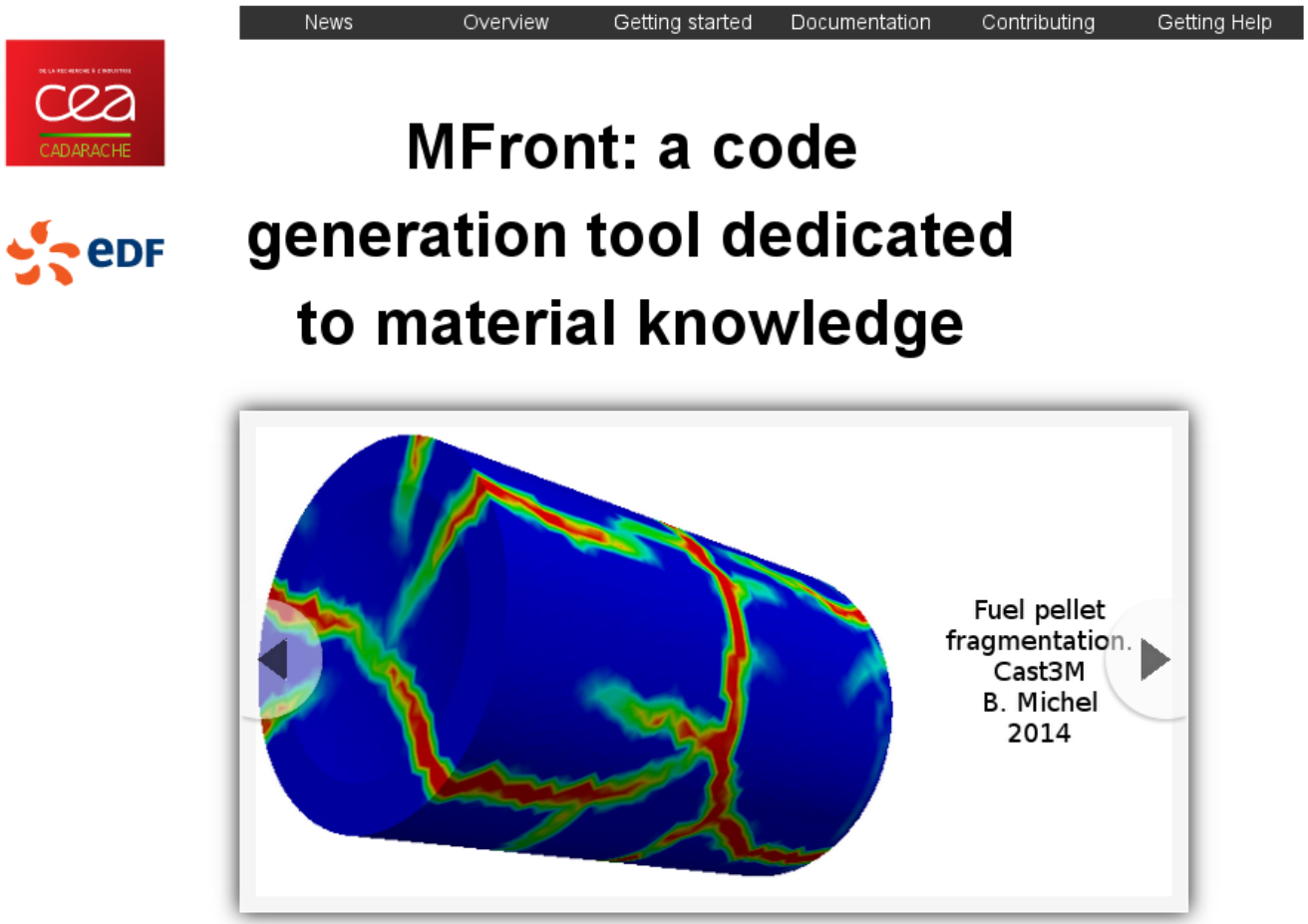
Software Portability

- TFEL 3.*x* is based on the C++ 11 standard.
- TFEL 3.*x* can be compiled with various C++ compilers :
  - gcc, from version 4.7 to version 8.1
  - clang, from version 3.5 to version 5.0
  - icc, version 2016, 2017 and 2018.
  - Visual Studio, version 15 and 17.
- TFEL is mainly developed on LiNuX.
- ports have been made to various POSIX systems, Mac Os, FreeBSD, OpenSolaris, cygwin, Windows Subsystem for LiNuX, Haiku etc... and Windows !

Software quality : an industrial strength software

- Very stringent compilers warnings :
  - `g++ -Wall -W -Wextra -pedantic -Wdisabled-optimization -Wlong-long -Winline -Wswitch -Wsequence-point -Wignored-qualifiers -Wzero-as-null-pointer-constant -Wvector-operation-performance -Wtrampolines -Wstrict-null-sentinel -Wsign-promo -Wsign-conversion -Wold-style-cast -Wnoexcept -Wmissing-include-dirs -Wmissing-declarations -Wlogical-op -Winit-self ...`
- Documentation.
- Continuous integration based on `Jenkins`
- More than 10 000 tests :
  - Most of them are based on `mtest`
- More tests inside `PLEIADES` applications.

The website



http://tfel.sourceforge.net

http://tfel.sourceforge.net/about.html

http://tfel.sourceforge.net/gallery.html

Mechanical behaviours

An very simple example

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

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

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

@Integrator{
  const real  eps = 1.e-12;
  const auto mu  = computeMu(young,nu);
  const auto seq = sigmaeq(sig);
  const auto tmp = A*pow(seq,E-1.);
  const auto df_dseq = E*tmp;
  const auto iseq  = 1/max(seq,eps*young);
  const Stensor n = 3*deviator(sig)*iseq/2;
  // implicit system
  // the StandardElasticity already set feel to deel-deto
  feel += dp*n;
  // by default, fp is initialized to dp
  fp    -= tmp*seq*dt;
  // jacobian
  dfeel_ddeel += 2*mu*theta*dp*iseq*(Stensor4::M()-(n^n));
  dfeel_ddp    = n;
  dfp_ddeel    = -2*mu*theta*df_dseq*dt*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 :

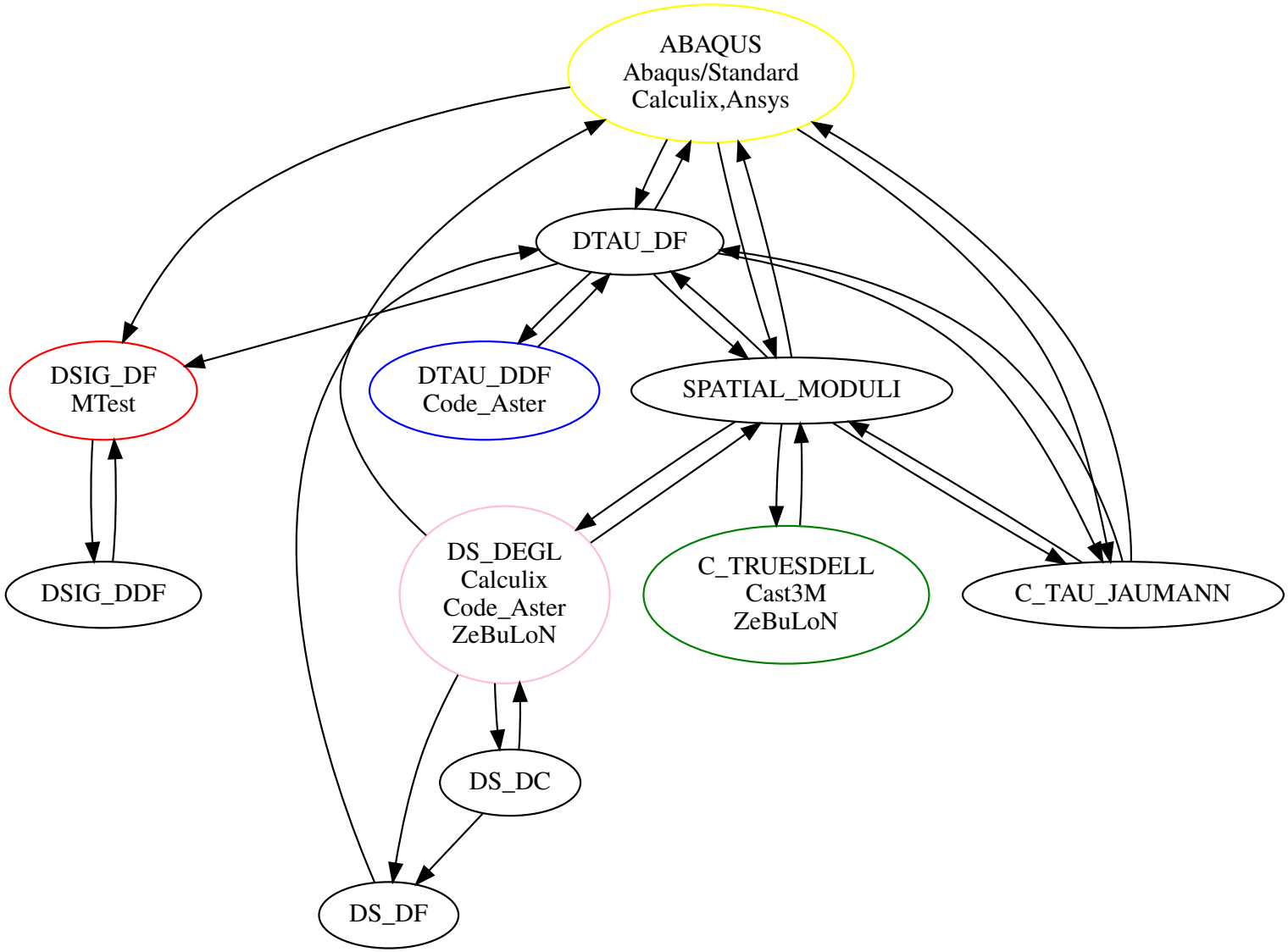
$$\begin{cases} \dfrac{\partial f_{\underline{\epsilon}^{el}}}{\partial \Delta\,\underline{\epsilon}^{el}} = \underline{\underline{\mathbf{I}}} + \dfrac{2\,\mu\,\theta\,\Delta\,p}{\sigma_{eq}}\left(\underline{\underline{\mathbf{M}}} - \underline{\mathbf{n}} \otimes \underline{\mathbf{n}}\right) \\ \dfrac{\partial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

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

Other functions of the mechanical behaviour

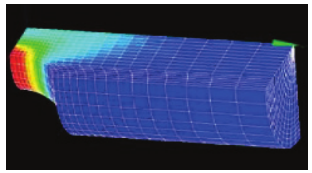
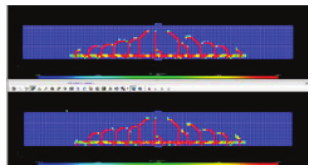
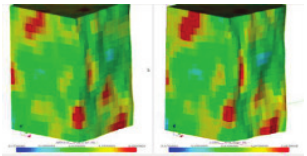
- Provide a estimation of the next time step for time step automatic adaptation
- Check bounds :
  - Physical bounds
  - Standard bounds
- Clear error messages
- Parameters
  - It is all about AQ !
  - Parametric studies, identification, etc. . .
- Generate ‘MTest‘ files on integration failures
- Generated 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. . .

Example of portability issue : tangent operator for finite strain behaviours



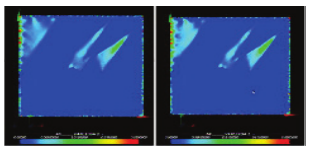
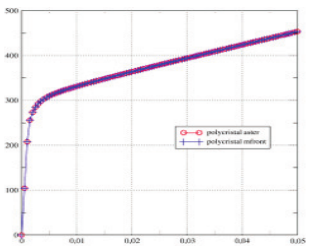
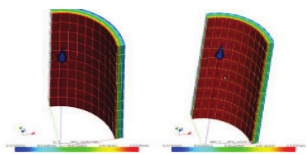
Performances

Benchmarks with Code_Aster (Jean-Michel Proix, EDF)

Description	Algorithme	Temps CPU total (Aster vs MFront)	Illustration
Visco-plastic and damaging for steel (Mustata and Hayhurst 2005; EDF 2012)	Implicit	17*mn*43*s* vs 7*mn*58*s*	
Damaging for concrete (Mazars and Hamon; EDF 2013a)	Default	45*mn* vs 63*mn*	
Generic Single crystal viscoplasticity (Méric and Cailletaud 1991; EDF 2013b)	Implicit	28*mn* vs 24*mn*	

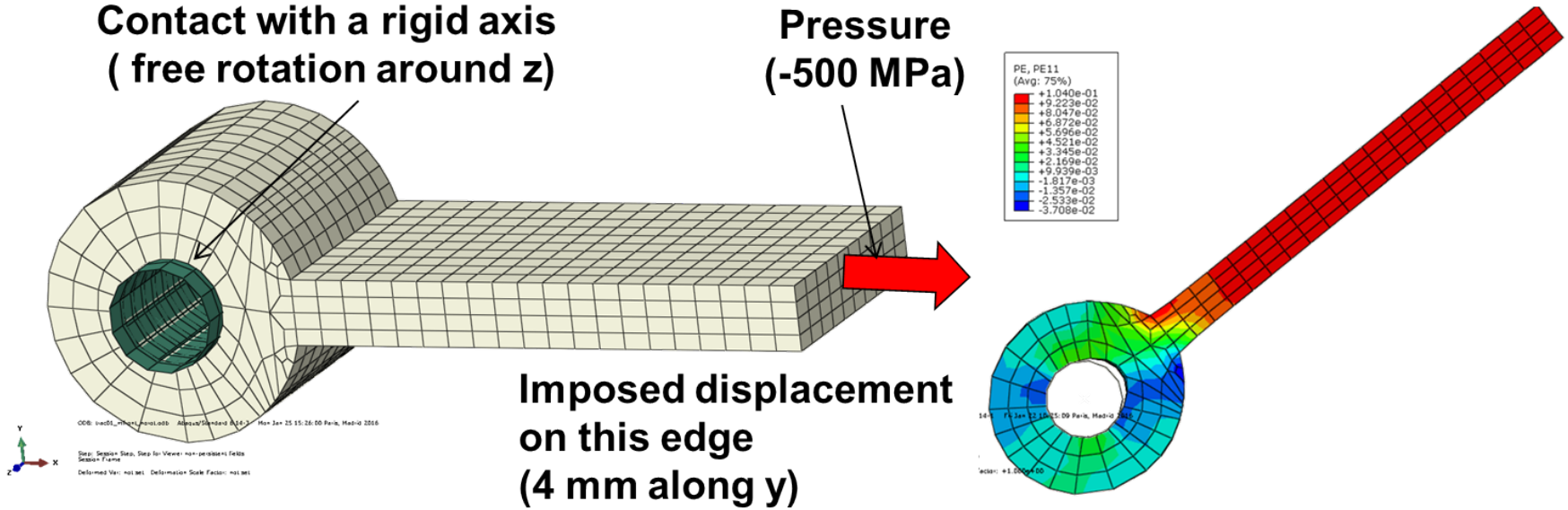
- Benchmarks made in 2013 by the Code_Aster developpers when evaluating MFront.
- Calling external behaviours in Code_Aster has improved...

Benchmarks with Code_Aster (Jean-Michel Proix, EDF)

Description	Algorithme	Temps CPU	Illustration
FCC single crystal viscoplasticity (Monnet, Naamane, and Devincre 2011 EDF (2013b))	Implicit	33*m*54*s* vs 29*m*30*s*	
FCC homogeneized polycrystals 30 grains (Berveiller and Zaoui 1978; EDF 2013b)	Runge-Kutta 4/5	9*s*67 vs 8*s*22	
Anisotropic creep with phase transformation (EDF 2013c)	Implicit	180*s* vs 171*s*	

- Benchmarks made in 2013 by the Code_Aster developpers when evaluating MFront.
- Calling external behaviours in Code_Aster has improved...

Benchmarks with Abaqus/Standard (D. Deloison, ArianeGroup)



	CPU Time (s)	Difference	Number of increments	Difference
ABAQUS	192	-	52	-
ABAQUS/UMAT(*)	1200	525%	295	467%
ABAQUS/MFRONT	201	5%	52	0%
(*) Numerical Jacobian				

The StandardElasto-ViscoPlasticity brick

Simple plastic behaviour based on the Hosford criterion

```
@DSL Implicit;
@Behaviour Hosford;

@ModellingHypotheses {".+"};
@Epsilon 1.e-16;
@IterMax 100;
@Theta 1;

@Brick "StandardElastoViscoPlasticity" {
  stress_potential : "Hooke" {
    young_modulus : 160e9,
    poisson_ratio : 0.3
  },
  inelastic_flow : "Plastic" {
    criterion : "Hosford" {a : 6},
    isotropic_hardening : "Linear" {R0 : 120e6}
  }
};
```

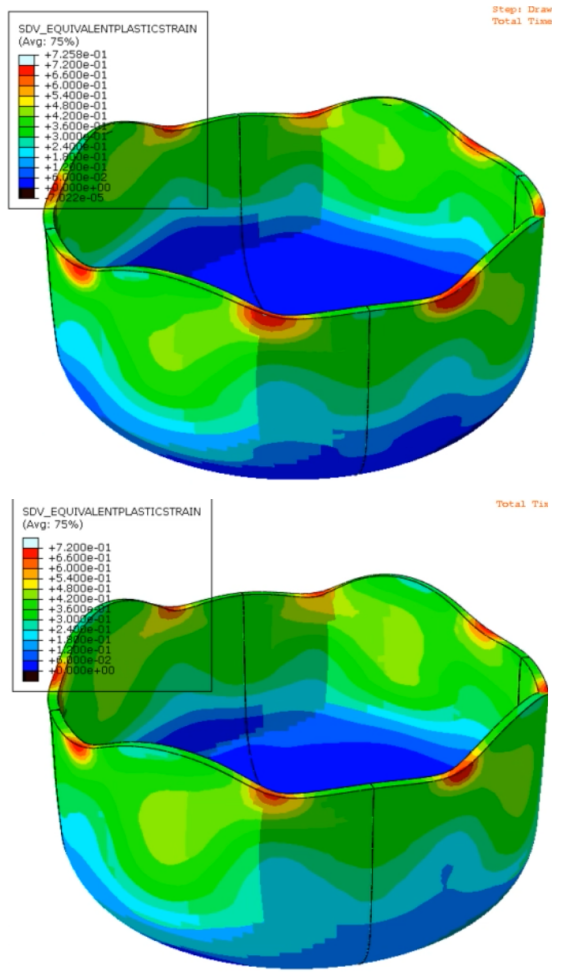
- http://tfel.sourceforge.net/StandardElastoViscoPlasticityBrick.html
- This brick provides a way to implement complex elasto-viscoplastic behaviours in a `declarative` way.
- Plane stress support and computation of the consistent tangent operator are automatically provided.

Deep drawing simulation (Computations by D. Deloison from ArianeGroup)

```
@DSL       Implicit ;
@Behaviour Barlat2090T3;

@AbaqusFiniteStrainStrategy[abaqus,abaqusexplicit] Native;
@ModellingHypotheses {".+"};
@OrthotropicBehaviour<Plate>;
@Epsilon 1.e-16;
@Theta   1;

@Brick StandardElastoViscoPlasticity{
  stress_potential  :  "Hooke" {
    young_modulus : 150e9,
    poisson_ratio  :  0.3},
  inelastic_flow  :  "Plastic"  {
    criterion  :  "Barlat"  {
      a  :  8,
      l1  :  {-0.069888, 0.079143, 0.936408, 0.524741,
              1.00306, 1.36318, 0.954322, 1.06906, 1.02377},
      l2  :  {0.981171, 0.575316, 0.476741, 1.14501,
              0.866827, -0.079294, 1.40462, 1.1471, 1.05166}
    },
    isotropic_hardening :  "Linear"  {R0 : 150e6}
  }
};
```



- Prediction of six or eight ears in a drawn cup... Yoon et Al.
- **Same results in** `Abaqus/Standard` **and** `Abaqus/Explicit`.

Conclusions

On going efforts

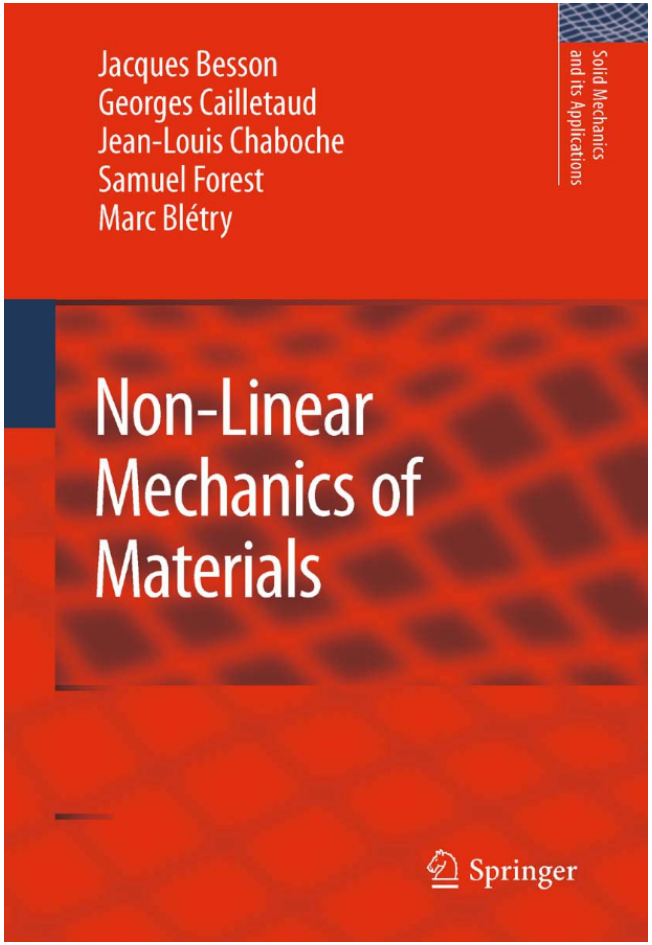
- `MFront` is a building block for the material knowledge management strategy of `EDF` and the `PLEIADES` plateform :
  - Focused of software quality and numerical performances.
- There are ongoing efforts on defining open tools to ensure a fully-consistent strategy encompassing all the steps up to engineering studies :
  - Experimental data.
  - Identification procedures.
  - Behaviour implementations.
  - Unit testing.
  - Documentation.

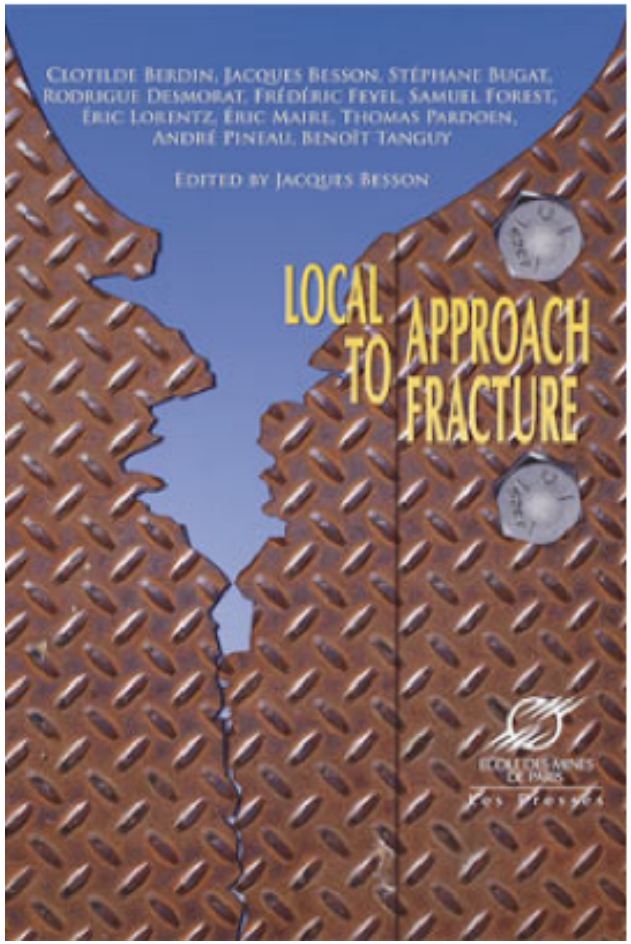
DE LA RECHERCHE À L'INDUSTRIE
cea

Future works

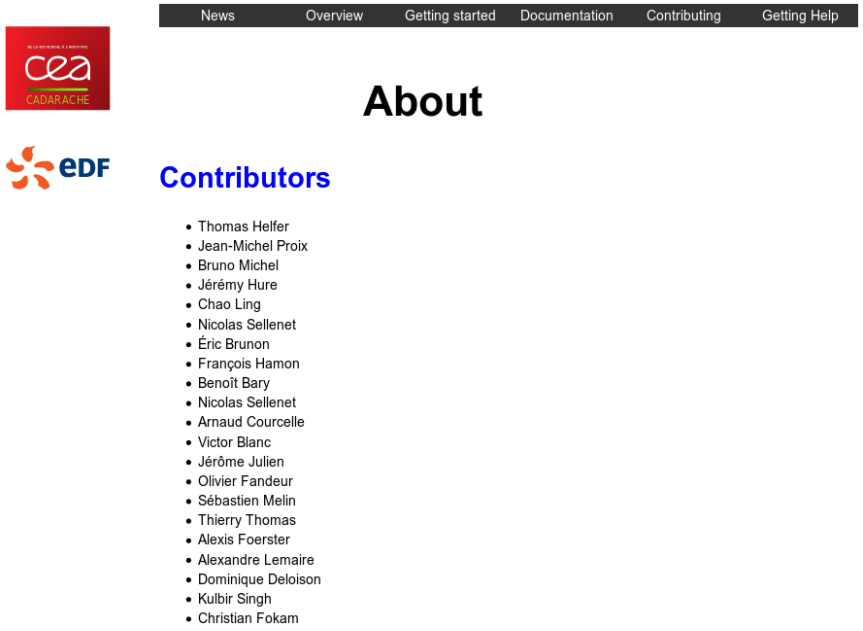
- Documentation :
  - Better code documentation.
  - New entries in the gallery.
- Numerical stability tests based on the `CADNA` library :
  - `www.lip6.fr/cadna`
  - `https://github.com/thelfer/cadna`
- Support of units :
  - Compile-time checks without any runtime overhead.
- More work on implicit algorithms :
  - Trust-region algorithms.
  - Homotopy methods.
- Generalised behaviours.
- New interfaces (ANSYS, LSDYNA).
- Generic `MFront` interface
- An official Debian/Ubuntu package.
- **The fourth MFront user days !**

A huge technical depth





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://twitter.com/TFEL_MFront
https://www.openhub.net/p/tfel
https://github.com/thelfer/

tfel-contact@cea.fr

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Cadarache | DEN/DEC/SESC b151 - 13108 Saint-Paul-Lez-Durance
T. +33 (0)4.42.25.23.66 | F. +33 (0)4.42.25.47.47
Établissement public à caractère industriel et commercial | RCS Paris B 775 685 019

Direction de l'Énergie Nucléaire
Département d'Études des Combustibles
Service d'Études et de Simulation du comportement des Combustibles