CSMA 2019
14ème Colloque National en Calcul des Structures
13-17 Mai 2019, Presqu'île de Giens (Var)

New functionalities of Versions 3.1 and 3.2 of TFEL/MFront

Thomas Helfer[1], Olivier Fandeur[2,3], Dominique Geoffroy[4], Charles Toulemonde[5], Jérémy Hure[6], Laurent Dupuy[7], Agathe Forré[8], Dominique Deloison[9], Frédéric Péralès[10], Arnaud Lejeune[11], Sébastien Thibault[12], Fabrice Richard[13], Yves Gaillard[14], Jürgen Almanstötter[15], Alexandre Gangnant[16], Jefry Draup[17], Anthony Kececioglu[18], Christophe Garnier[19], Christophe Garnier[20], Jérôme Roland[21]

[1] *CEA, DEN/DEC/SESC, Département d'Études des Combustibles, thomas.helfer@cea.fr*
[2] *CEA, DEN/DM2S/SEMT, Département de Modélisation des Systèmes et des Structures, olivier.fandeur@cea.fr*
[3] *IMSIA, UMR 8193, CNRS-EDF-CEA-ENSTA*
[4] *EDF R&D, Département ERMES, dominique.geoffroy@edf.fr*
[5] *EDF R&D, Département Matériaux et Mécanique des Composants (MMC), charles.toulemonde@edf.fr*
[6] *CEA, DEN/DMN/SEMI, Département des Matériaux pour le Nucléaire, jeremy.hure@cea.fr*
[7] *CEA, DEN/DMN/SRMA, Département des Matériaux pour le Nucléaire, laurent.dupuy@cea.fr*
[8] *Groupe PSA, Science and Future Technologies Department, agathe.forre@mpsa.com*
[9] *ArianeGroup, dominique.deloison@ariane.group*
[10] *IRSN, PSN-RES/SEMIA/LPTM, frederic.perales@irsn.fr*
[11] *COMUE UBFC, UFC, Institut FEMTO-ST (UMR CNRS 6174), Dpt Mécanique Appliquée/Pole Calcul Scientifique, arnaud.lejeune@univ-fcomte.fr*
[12] *COMUE UBFC, ENSMM, Institut FEMTO-ST (UMR CNRS 6174), Dpt Mécanique Appliquée, sebastien.thibaud@univ-fcomte.fr*
[13] *COMUE UBFC, UFC, Institut FEMTO-ST (UMR 6174), Dpt Mécanique Appliquée, fabrice.richard@univ-fcomte.fr*
[14] *COMUE UBFC, UFC, Institut FEMTO-ST (UMR 6174), Dpt Mécanique Appliquée, ygaillar@univ-fcomte.fr*
[15] *Osram GmbH, SP MFPD PRE-PLM DMET, j.almanstoetter@osram.de*
[16] *EDF Energy R&D UK Centre, alexandre.gangnant@manchester.ac.uk*
[17] *EDF Energy R&D UK Centre, jefri.draup@edfenergy.com*
[18] *EDF R&D, Département Matériaux et Mécanique des Composants (MMC), anthony.kececioglu@edf.fr*
[19] *CEA, DEN/DTN/STCP, christophe.garnier@cea.fr*
[20] *Framatome, Fuel, FDM-F Materials & Thermal-Mechanics Department, christophe.garnier@framatome.com*
[21] *Framatome, Fuel, FDM-F Materials & Thermal-Mechanics Department, jerome.roland@framatome.com*

Abstract — `MFront` is a tool which allows easy implementation of arbitrary complex mechanical behaviours in an efficient way. Those implementations are portable between various finite element solvers and solvers based on FFT. `MFront` is part of the open-source `TFEL` project. The purpose of this paper is to highlight a selected set of features introduced in Versions 3.1 and 3.2 of `TFEL`.

1 Overview of `TFEL`, `MFront` and `MTest`

The `TFEL` project is an open-source collaborative development of the French Alternative Energies and Atomic Energy Commission (CEA) and Électricité de France (EDF) in the framework of the `PLEIADES` plateform (1). `TFEL` provides mathematical libraries which are the basis of the `MFront` code generator and the `MTest` solver (2, 3).

`MFront` translates a set of closely related domain specific languages into plain C++ on top of the `TFEL` library. Those languages are meant to be easy to use and learn by researchers and engineers and cover three kinds of material knowledge: material properties (Young modulus, thermal conductivity, etc.), mechanical behaviours[1] and simple point-wise models (such as material swelling

[1] Among the many projects aiming at easing the implementation of mechanical behaviours (see for example (4–6)), `MFront` can be compared to the `ZebFront` code generator which is part of `ZMat` library (7). A comprehensive

used in fuel performance codes).

Authors of `MFront` paid particular attention to the robustness, reliability and numerical efficiency of the generated code, in particular for mechanical behaviours: various benchmarks show that `MFront` implementations are competitive with native implementations available in the `Cast3M` (9), `Code_Aster` (10), `Europlexus` (11), `Abaqus/Standard` , `Abaqus/Explicit` (12, 13), `Cyrano3` (14) and `Galileo` solvers.

Portability is also a very important issue: a behaviour written in `MFront` shall be usable in any solver for which an interface exists. In addition to the aforementioned solvers, interfaces exist for: `Ansys` (15), `ZMat` (7), `CalculiX` (16, 17). This paper will also describe the `generic` interface and the `MFrontGenericInterfaceSupport` project which allow developers of home-brew solvers to add support for external behaviours in their code.

2 Some improvements of the `TFEL/Material` library

The `TFEL/Material` provides functions to handle advanced yield criteria, such as:

- The Hosford 1978 yield criterion (18), which is defined as follows:

$$\sigma_{\text{eq}}^{H} = \sqrt[a]{\frac{1}{2}\left(|\sigma_1-\sigma_2|^a+|\sigma_1-\sigma_3|^a+|\sigma_2-\sigma_3|^a\right)}$$

  where σ_1, σ_2 and σ_3 are the eigenvalues of the stress[2].
- The Barlat 2004 yield criterion (19), defined as follows:

$$\sigma_{\text{eq}}^{B} = \sqrt[a]{\frac{1}{4}\left(\sum_{i=0}^{3}\sum_{j=0}^{3}\left|s_i'-s_j''\right|^a\right)} \quad \text{with} \quad \begin{cases} \underline{s}' = \underline{\underline{\mathbf{L}}}' : \underline{\sigma} \\ \underline{s}'' = \underline{\underline{\mathbf{L}}}'' : \underline{\sigma} \end{cases}$$

  where s_i' and s_i'' are the eigenvalues of two transformed stresses $\underline{s}'$ and $\underline{s}''$. The linear transformations $\underline{\underline{\mathbf{L}}}'$ and $\underline{\underline{\mathbf{L}}}''$ describe the orthotropy of the yield surface.

Those two yield criteria are based on the eigenvalues of the stress.[3] The computation of the second derivative, required to build the jacobian of the implicit system, is thus quite involved. The implementations of those two criteria closely follows the work of Scherzinger (21)[4].

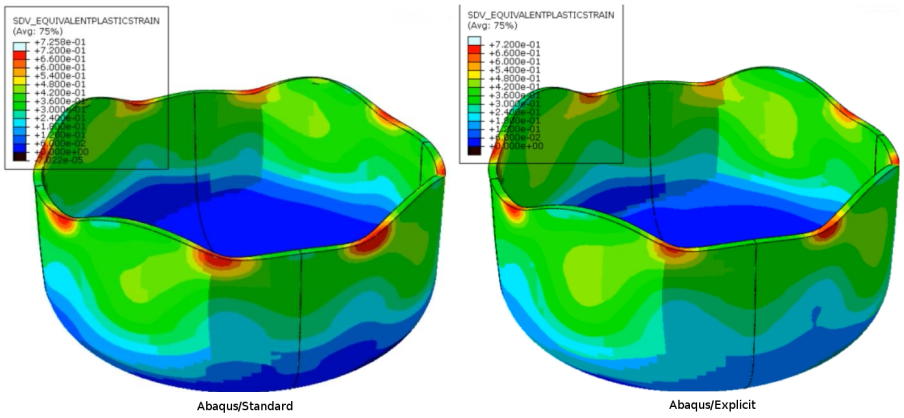

Figure 1: Simulation of the deep drawing of a highly anisotropic 2090-T3 aluminum alloy sheet

comparison between these two solutions has been presented at the ZSet User Days (8).

[2]When a goes to infinity, the Hosford stress reduces to the Tresca stress criterion. When $n = 2$ the Hosford stress reduces to the von Mises stress criterion.

[3]When a is an integer, Soare and Barlat (20) proposed another approach, not yet implemented in `TFEL`.

[4]In particular, special care has been taken to avoid overflow in the evaluation of those criteria.

Figure 1 shows an application of this development to the simulation of the deep drawing of a highly anisotropic 2090-T3 aluminum alloy sheet as discussed by Yoon et al. (22). This figure shows that the results obtained by `Abaqus/Standard` and `Abaqus/Explicit` are close, illustrating the portability of the implementation from one solver to another.

3 Mechanical behaviours

3.1 Bricks

Bricks are a new feature introduced in Version `3.0`. A brick is meant to simplify the implementation of a certain class of behaviours and hide tedious details for the user to focus on the physics. The following bricks are available:

- the `StandardElasticity` brick (Version 3.0), described in a previous paper (23).
- the `DDIF2` brick (Version 3.1), which describes a damage behaviour used in fuel performance codes (24).
- the `FiniteStrainSingleCrystal` brick (Version 3.0), described in Section 3.2.2.
- the `StandardElastoViscoPlasticity` brick, which is now described.

This `StandardElastoViscoPlasticity` brick is used to describe a specific class of strain based behaviours based on an additive split of the total strain $\underline{\varepsilon}^{\mathrm{to}}$ into an elastic part $\underline{\varepsilon}^{\mathrm{el}}$ and an one or several inelastic strains describing plastic (time-independent) flows and/or viscoplastic (time-dependent) flows:

$$\underline{\varepsilon}^{\mathrm{to}} = \underline{\varepsilon}^{\mathrm{el}} + \sum_{i_{\mathrm{p}}=0}^{n_{\mathrm{p}}} \underline{\varepsilon}^{\mathrm{p}}_{i_{\mathrm{p}}} + \sum_{i_{\mathrm{vp}}=0}^{n_{\mathrm{vp}}} \underline{\varepsilon}^{\mathrm{vp}}_{i_{\mathrm{vp}}}$$

The brick decomposes the behaviour into two components:

- The stress potential which defines the relation between the elastic strain $\underline{\varepsilon}^{\mathrm{el}}$ and possibly some damage variables and the stress $\underline{\sigma}$. As the definition of the elastic properties can be part of the definition of the stress potential, the thermal expansion coefficients can also be defined in the block corresponding to the stress potential.
- A list of inelastic flows. Inelastic flows are defined by:
  - A stress criterion. The following criteria are available: `von Mises`, `Hill 1948`, `Drucker 1949`, `Hosford 1972`, `Barlat 2004`, etc…
  - A flow criterion. By default the flow criterion is the same as the stress criterion.
  - A set of isotropic hardening rules. The following rules are available: `Linear`, `Voce` and `Swift`.
  - A set of kinematic hardening rules. The following rules are avaible: `Prager`, `Armstrong-Frederick`, etc…

A complete description of the features provided by the `StandardElastoViscoPlasticity` brick can be found in the dedicated web page.[5]

Figure 2 shows how to use this brick to implement a perfect plastic behaviour based on the Hosford criterion (see Section 2).

The brick handles many aspects of the behaviour integration. For example, if several yield surfaces are involved, a set of boolean values is introduced associated with the activation or desactivation of each yield surface: after each convergence of the implicit system, consistency of the activation/desactivation of each yield surface is checked. If one system' status changes, the Newton iterations are restarted until a state satisfying the implicit equations and the consistency checks is found.

[5] http://tfel.sourceforge.net/StandardElastoViscoPlasticityBrick.html.

```
@DSL Implicit;
@Behaviour StandardElastoViscoPlasticityPlasticityTest9;
@Author Thomas Helfer;
@Date 23/04/2018;

@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}
  }
};
```

Figure 2: Example of the `StandardElastoViscoPlasticity` brick usage.

This brick shows a deep move in `MFront`, as no more bare `C++` code is required to implement a new behaviour. The syntax used is only declarative and strongly inspired by the `JSON` format,[6] which is meant to be simple and readable. The underlying implementation has been designed so that graphical user interfaces can easily be build which would allow an even simpler usage in future versions.

However, it is important to note that the user still can add additional state variables and associated implicit equations.

3.2 Single crystal behaviours in `MFront`

Single crystal behaviours are an important use case of `MFront` (25–27). To simplify their implementations, a set of keywords to describe the crystal structure, slip systems and interaction matrices (Section 3.2.1), and a dedicated brick (Section 3.2.2), have been introduced.

3.2.1 The `TFELNUMODIS` library

Support for writting single crystal behaviours have been greatly improved thanks to the `TFELNUMODIS` library, which borrows code for the `NUMODIS` project (28).

The crystal structure can be defined using the `@CrystalStructure` keyword. The following crystal structures are supported: `Cubic` (cubic structure), `BCC` (body centered cubic structure), `FCC` (face centered cubic structure) and `HCP` (hexagonal closed-packed structures).

A single slip systems family can be defined by one of the following synonymous keywords: `@SlidingSystem`, `@GlidingSystem` or `@SlipSystem`. Several slip systems families can be defined by `@SlidingSystems`, `@GlidingSystems` or `@SlipSystems`.

Two kind of interaction matrices can be defined:

- The first interaction matrix is defined through the `@InteractionMatrix` keyword and is meant to describe the effect of dislocations on hardening.
- The second interaction matrix is defined through the `@DislocationsMeanFreePath InteractionMatrix` keyword and is meant to evaluate the effect of all the dislocations on the mean free path of dislocations of a specific system.

[6] JavaScript Object Notation. See https://en.wikipedia.org/wiki/JSON.

Those keywords are fully documented on the dedicated page.[7]

3.2.2 Description of the `FiniteStrainSingleCrystal` brick

The `FiniteStrainSingleCrystal` brick is based on a thermodynamic framework which describes the (visco)-plasticity of single crystal (see (25, 29–31) for further details). A multiplicative decomposition $\mathbf{F}_e \cdot \mathbf{F}_p$ of the deformation gradient $\mathbf{F}$ is assumed. The driving force for plasticity is the Mandel stress tensor defined in the intermediate configuration.

The `FiniteStrainSingleCrystal` brick handles all the kinematic aspects related to the multiplicative decomposition $\mathbf{F}_e \cdot \mathbf{F}_p$, the definition of the Mandel stress tensor and its derivative (if required) and the computation of the consistent tangent operator.

The user can thus focus on the definition on the constitutive equations describing the plastic evolutions of each slip systems.

The hidden details of the implementation closely follows the one described in the `Code_Aster` documentation (31).

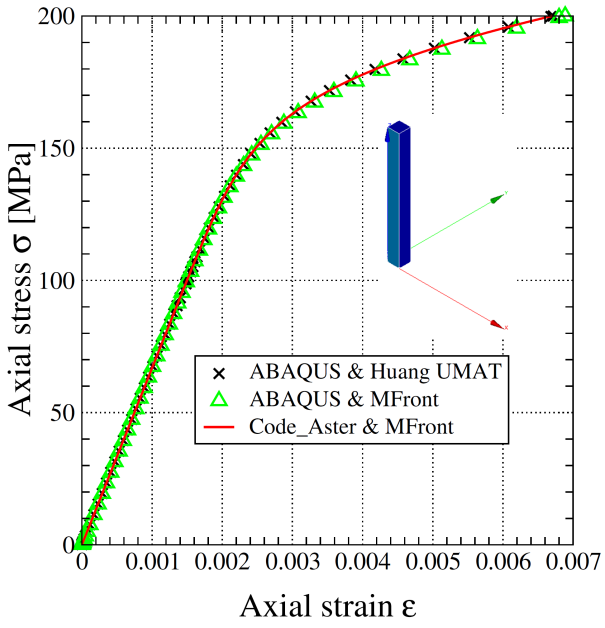

Figure 3: Benchmark of ABAQUS UMAT Fortran subroutine given by Huang (32) and MFront implementation. A single crystalline copper bar (FCC crystal structure, slip system $\{111\} < 110 >$) is subjected to a tensile load of $200\,MPa$ in $1\,s$. More details of the parameters used for the test case can be found in Reference (32).

Figure 3 shows how this brick has been used to implement a phenomenological crystal plasticity model proposed by Huang (32). This figure also shows that the same implementation works equally in `Abaqus/Standard` and `Code/Aster`.

4 Solver interfaces

4.1 The generic behaviour interface

Whereas other interfaces target a specific solver and thus are restricted by choices made by this specific solver, the `generic` interface has been created for developers of homebrew solvers who are able to modify their code to take full advantage of `MFront` behaviours.

[7] http://tfel.sourceforge.net/singlecrystal.html.

This interface is tightly linked with the `MFrontGenericInterfaceSupport` project which is available on `github`.[8] This project has a more liberal licence than `TFEL` which allows it to be included in both commercial and open-source solvers/library. This licensing choice explains why this project is not part of the `TFEL`. While the core of the project is written in `C++`, bindings are provided for the following languages: `C`, `fortran`, `python`. The following solvers already use this project to create an interface with `MFront`: `XPer` (33), `OpenGeoSys` (34). An interface to `MoFEM` is currently being discussed (35).

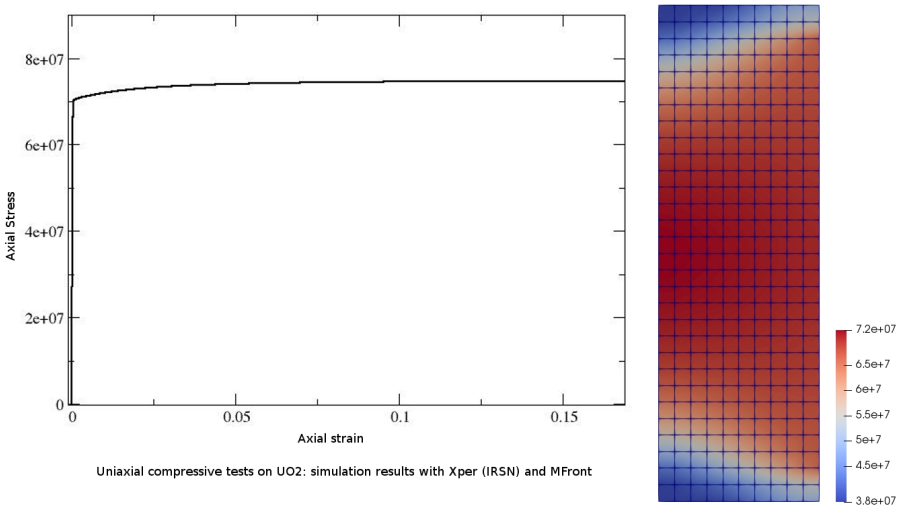

Figure 4: First results on the coupling of `MFront` and `XPer` through the `MFrontGenericInterfaceSupport` project

Figure 4 illustrates the coupling between `MFront` and `XPer` to the simulation the uniaxial compression tests on Uranium dioxyde (see (36, 37) for details about the behaviour used).

4.2 New interfaces

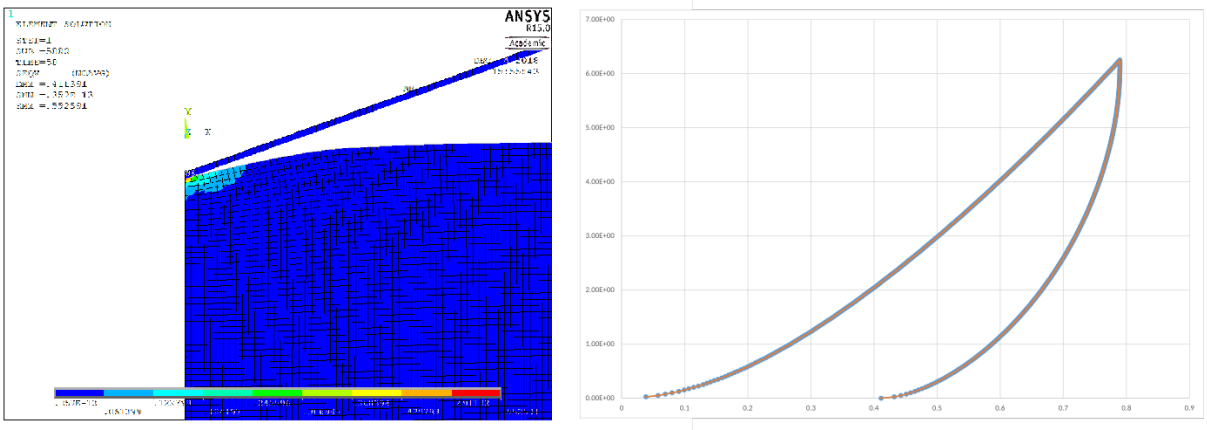

Figure 5: Qualification of the `Ansys` interface on the axisymmetrical simulation of the indentation of a viscoelastic material. Cauchy Stress value for the maximal imposed displacement (left). Comparison of the force-displacement curves obtained with a reference implementation (38) and a `MFront`' implementation: results are undistinguishable (right)

Two new interfaces are now available:

- An interface for the `Ansys` APDL solver. Figure 5 shows an example of MFront behaviour usage in `Ansys` (`Ansys` 15 under `Windows` using the `Visual Studio 15` compiler suite).

[8] https://github.com/thelfer/MFrontGenericInterfaceSupport.

- An interface to the open-source `CalculiX` solver (16, 17).

5 Numerical improvements

5.1 Robustness of the implicit scheme

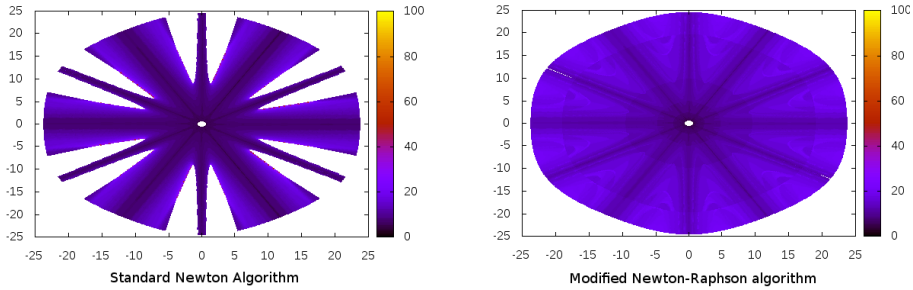

Figure 6: Comparison of the robustness of two implicit scheme algorithms for a perfect plastic behaviour based on the Hosford criterion using the test proposed by Scherzinger (21)

To illustrate some of the work made to increase the robustness of the implicit scheme, let us consider the perfect plastic behaviour based on the Hosford equivalent stress whose implementation has been described in Section 3.1.

For such a behaviour, Scherzinger proposed a test of the robustness of the integration scheme (21). For a given direction the π plane, a reference strain increment is chosen which would lead to an elastic prediction of the strain whose equivalent stress (using Hosford criterion) is equal to the yield stress. A set of tests is then performed by imposing, in one step, this reference strain increment amplified by an given factor. This amplification factor starts from 1 and is increased up to a value corresponding to an integration failure or a maximal value of 30 if no integration failure occurs. The number of iterations required to reach convergence is saved.

Figure 6 reports the results of this test in the case an Hosford exponent a egal to 6. Two algorithms are compared: the standard Newton-Raphson algorithm and for a slightly modified version of this algorithm. The modification consists in testing the value of the Hosford equivalent stress during the Newton iterations: if this value exceeds the yield criterion by a factor 1.5, the Newton-step amplitude is divided by a factor 2. The figure shows that this slight modification strongly increases the robustness of the algorithm. This modification has been used on the computation of the deep drawing of a cup depicted on Figure 1.

The results of this test for various values of the Hosford exponent are reported on a dedicated web page[9]. For very high values of a, the standard Newton-Raphson algorithm and the modified version perform poorly. One has to combine an accurate algorithm for the computation of the eigenvalues and eigen vectors (the Jacobi algorithm)[10] and the Levenberg-Marquardt algorithm to ensure convergence.

5.2 Overall work on the numerical stability

[9] http://tfel.sourceforge.net/hosford.net

[10] By default, `TFEL` will use an explicit algorithm based on Cardano formulae which is faster but less accurate than the Jacobi algorithm. A comparison of the various algorithms available available in TFEL can be found in the release note of versions 3.1 available here: http://tfel.sourceforge.net/release-notes-3.1.html. The implementation of the Jacobi algorithm has been adapted from Joachim Kopp' work (39, 40).

Table 1: Test on 10^6 random symmetric tensors $\underline{s}$ in double precision $\Delta_\infty = \max_{\underline{s}} \max_{i\in[1,2,3]} \|\underline{s}\cdot\vec{v}_i - \lambda_i\vec{v}_i\|$.

Algorithm	Δ_∞	Time ratio
TFELEIGENSOLVER	7.75e-14	1
FSESJACOBIEIGENSOLVER	1.05e-15	1.94
FSESQLEIGENSOLVER	3.30e-15	1.45
FSESHYBRIDEIGENSOLVER	3.53e-10	0.61
FSESANALYTICALEIGENSOLVER	1.09e-09	0.62

Authors put much emphasis on working on the overall numerical stability of the algorithms provided in the `TFEL/Math` library. To illustrate this, Table 1 compares the accurary of various eigen solvers, comparing the default solver of `TFEL/Math` (`TFELEIGENSOLVER`), based on the Cardano formulae, to various solvers proposed by Joachim Kopp which have been introduced in `TFEL/Math` (39, 40). While the default solver provides a good compromise in terms of accuracy and efficiency, the Jacobi solver is the most accurate and seems fairly insensitive to the optimisation flags used.

This work now allows to safely use more aggressive optimisation flags which allow the compilers to ignore some of the rules imposed by the IEEE754 standard[11].

6 Documentation: The gallery and the `MFrontGallery` project

The `MFront` gallery is meant to present well-written implementation of behaviours that will be updated to follow `MFront` latest evolutions. In each case, the integration algorithm is fully described.

The `MFrontGallery` project is a `cmake` project which builds material libraries for all the codes and/or languages supported by `MFront` based on the implementation described in the gallery. The purpose of this project is twofold:

- It delivers ready-to-use shared libraries for a wide variety of phenomena.
- It provides an example of how to build a compilation project for `MFront` files, including lots of useful `cmake` macros and recipes to build shared libraries and add tests.

The `MFrontGallery` project is available as a `github` repository[12].

7 Conclusions

This paper has highlighted some improvements made in the versions 3.1 and 3.2 of `TFEL` and `MFront`. Mechanical behaviours can be written even more easily than in previous versions and performances are competitive with built-in behaviour implementations of most mechanical solvers.

The `MFront` users' community is steadily increasing outside the nuclear industry and the french mechanical community: its use now encompasses a wide range of materials and applications (see Figure 7 for example).

[11] Namely, the `-ffast-math` option with the GNU compiler `g++`.
[12] https://github.com/thelfer/MFrontGallery

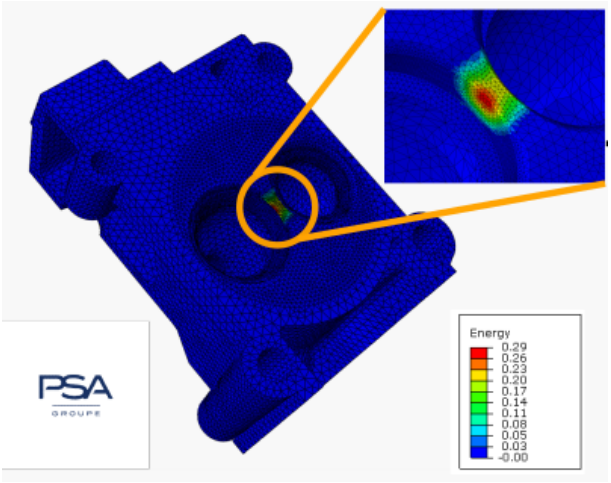

Figure 7: An example of MFront' usage at Groupe PSA: thermo-mechanical design of an automotive single cylinder head

7.1 Interfaces under developement

The following interfaces are under development:

- an interface for the DIANA FEA solver (41).
- an interface for the LS-DYNA solver (42).

Acknowledgements This research was conducted in the framework of the PLEIADES project, which was supported financially by the CEA (Commissariat à l'Énergie Atomique et aux Énergies Alternatives), EDF (Électricité de France) and Framatome.

The authors are also thankful to Jamel Tirari and Khouloud Derouiche for the contributions they made during their respective internships.

8 References

1. Marelle, Vincent, Goldbronn, Patrick, Bernaud, Stéphane, Castelier, Étienne, Julien, Jérôme, Nkonga, Katherine, Noirot, Laurence and Ramière, Isabelle. New developments in ALCYONE 2.0 fuel performance code. In : *Top fuel*. Boise, USA, 2016.

2. Helfer, Thomas, Proix, Jean-Michel and Fandeur, Olivier. Implantation de lois de comportement mécanique à l'aide de MFront : Simplicité, efficacité, robustesse et portabilité. In : *12ème colloque national en calcul des structures*. Giens, France : CSMA, June 2015.

3. Helfer, Thomas, Michel, Bruno, Proix, Jean-Michel, Salvo, Maxime, Sercombe, Jérôme and Casella, Michel. 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*. September 2015. Vol. 70, no. 5, p. 994–1023. DOI 10.1016/j.camwa.2015.06.027. Available from: http://www.sciencedirect.com/science/article/pii/S0898122115003132

4. Stainier, Laurent. Introduction to MatLib: Founding concepts. 2010.

5. Nonlinear CAE, Japan Association for. Unified material model driver for plasticity.. 2018. Available from: https://www.jancae.org/annex/annexUMMDe/index.html

6. Portillo, David, Pozo, Daniel del, Rodríguez-Galán, Daniel, Segurado, Javier and Romero, Ignacio. MUESLI - a material UnivErSal LIbrary. *Advances in Engineering Software*.

March 2017. Vol. 105, p. 1–8. DOI 10.1016/j.advengsoft.2017.01.007. Available from: http://www.sciencedirect.com/science/article/pii/S0965997816301430

7. Mines Paristech and Onera. ZMat: A material model library.. 2018. Available from: http://www.zset-software.com/products/z-mat/

8. Helfer, Thomas, Fandeur, Olivier, De Soza, Thomas, Deloison, Dominique and Toulemonde, Charles. Material knowledge management with the MFront code generator. Description of the ZMAT interface. Centre de l'Onéra, Chatillon, 2017. Available from: https://github.com/thelfer/tfel-doc/blob/master/Talks/ZSetUserDays2017/mfront-zset.pdfClub des Utilisateurs du code Z-set

9. CEA. Cast3M website.. 2018. Available from: http://www-cast3m.cea.fr/

10. EDF. Code_Aster website.. 2018. Available from: http://www.code-aster.org

11. CEA and JRC. Europlexus web site.. 2018. Available from: http://www-epx.cea.fr/index.php/what-is-epx

12. Dassault Systèmes. Abaqus web site.. 2018. Available from: http://www.3ds.com/products-services/simulia/products/abaqus/

13. Deloison, Dominique and Congourdeau, Fabrice. Testing and validation of the MFRONT interface for ABAQUS. EDF Lab Sacaly, 2016. Available from: https://github.com/thelfer/tfel-doc/blob/master/MFrontUserDays/SecondUserDay/deloison-abaqus.pdfSecond MFront users day

14. Petry, Charles and Helfer, Thomas. Advanced mechanical resolution in CYRANO3 fuel performance code using MFront generation tool. In : *LWR fuel performance meeting/TopFuel/WRFPM*. Zurich, Switzerland, July 2015.

15. ANSYS Inc. Engineering simulation & 3D design software.. 2018. Available from: https://www.ansys.com/

16. Dhondt, Guido. The finite element method for three-dimensional thermomechanical applications. 1 edition. Hoboken, NJ : Wiley, 2004. ISBN 978-0-470-85752-6.

17. Dhondt, Guido and Wittig, Klaus. Calculix: A free software three-dimensional structural finite element program.. 2018. Available from: http://www.calculix.de/

18. Hosford, W. F. A generalized isotropic yield criterion. *Journal of Applied Mechanics*. 1972. Vol. 39, no. 2, p. 607–609.

19. Barlat, F., Aretz, H., Yoon, J. W., Karabin, M. E., Brem, J. C. and Dick, R. E. Linear transfomation-based anisotropic yield functions. *International Journal of Plasticity*. 1 May 2005. Vol. 21, no. 5, p. 1009–1039. DOI 10.1016/j.ijplas.2004.06.004. Available from: http://www.sciencedirect.com/science/article/pii/S0749641904001160

20. Soare, Stefan C. and Barlat, Frédéric. A study of the yld2004 yield function and one extension in polynomial form: A new implementation algorithm, modeling range, and earing predictions for aluminum alloy sheets. *European Journal of Mechanics - A/Solids*. 1 November 2011. Vol. 30, no. 6, p. 807–819. DOI 10.1016/j.euromechsol.2011.05.006. Available from: http://www.sciencedirect.com/science/article/pii/S0997753811000726

21. Scherzinger, W. M. A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method. *Computer Methods in Applied Mechanics and Engineering*. 15 April 2017. Vol. 317, p. 526–553. DOI 10.1016/j.cma.2016.11.026. Available from: http://www.sciencedirect.com/science/article/pii/S004578251630370X

22. Yoon, J. W., Barlat, F., Dick, R. E. and Karabin, M. E. Prediction of six or eight ears in a drawn cup based on a new anisotropic yield function. *International Journal of Plasticity*.

1 January 2006. Vol. 22, no. 1, p. 174–193. DOI 10.1016/j.ijplas.2005.03.013. Available from: http://www.sciencedirect.com/science/article/pii/S074964190500063X

23. Helfer, Thomas, Fandeur, Olivier, Haboussa, David, Deloison, Dominque, Jamond, Olivier, Munier, Rémi, Berthon, Lucie, Castelier, Étienne and Isabelle, Ramière. New functionalities of the 3.0 version of TFEL, MFront and MTest. In :. Giens, France : CSMA, June 2017.

24. Michel, B., Helfer, T., Ramière, I. and Esnoul, C. A new numerical methodology for simulation of unstable crack growth in time independent brittle materials. *Engineering Fracture Mechanics*. 12 August 2017. DOI 10.1016/j.engfracmech.2017.08.009. Available from: http://www.sciencedirect.com/science/article/pii/S001379441630412X

25. Ling, Chao. Modeling the intragranular ductile fracture of irradiated steels. Effects of crystal anisotropy and strain gradient. PhD thesis. PSL Research University, 2017. Available from: https://pastel.archives-ouvertes.fr/tel-01699226/document

26. Shi, Qiwei. Experimental and numerical studies on the micromechanical crystal plasticity behavior of an RPV steel. Paris Saclay, 2018. Available from: http://www.theses.fr/2018SACLN009

27. Portelette, Luc, Amodeo, Jonathan, Madec, Ronan, Soulacroix, Julian, Helfer, Thomas and Michel, Bruno. Crystal viscoplastic modeling of UO2 single crystal. *Journal of Nuclear Materials*. 22 June 2018. DOI 10.1016/j.jnucmat.2018.06.035. Available from: http://www.sciencedirect.com/science/article/pii/S0022311518305981

28. CEA and CNRS. PROJECT numodis: Numerical modelling of dislocations.. 2018. Available from: http://www.numodis.fr/

29. Han, Xu. Modélisation de la fragilisation due au gonflement dans les aciers inoxydables austénitiques irradiés. PhD thesis. 2012. Available from: http://www.theses.fr/2012ENMP0066/document2012ENMP0066

30. Ling, Chao, Tanguy, Benoît, Besson, Jacques, Forest, Samuel and Latourte, Felix. Void growth and coalescence in triaxial stress fields in irradiated FCC single crystals. *Journal of Nuclear Materials*. 15 August 2017. Vol. 492, p. 157–170. DOI 10.1016/j.jnucmat.2017.04.013. Available from: http://www.sciencedirect.com/science/article/pii/S0022311516310728

31. EDF. R5.03.11 révision : 10623: Comportements élastoviscoplastiques mono et polycristallins. Référence du Code Aster. EDF-R&D/AMA, 2018. Available from: http://www.code-aster.org

32. Huang, Yonggang. Report Mech-178: A user-material subroutine incorporating single crystal plasticity in the ABAQUS finite element program. Division of Engineering and Applied Science. Cambridge, Massachusetts : Division of Engineering; Applied Science, Harvard University, 1991. Available from: http://www.columbia.edu/~jk2079/fem/umat_documentation.pdf

33. Perales, Frederic, Dubois, Frederic, Monerie, Yann, Piar, Bruno and Stainier, Laurent. A NonSmooth contact dynamics-based multi-domain solver. *European Journal of Computational Mechanics*. 1 January 2010. Vol. 19, no. 4, p. 389–417. DOI 10.3166/ejcm.19.389-417. Available from: https://doi.org/10.3166/ejcm.19.389-417

34. Kolditz, O., Bauer, S., Bilke, L., Böttcher, N., Delfs, J. O., Fischer, T., Görke, U. J., Kalbacher, T., Kosakowski, G., McDermott, C. I., Park, C. H., Radu, F., Rink, K., Shao, H., Shao, H. B., Sun, F., Sun, Y. Y., Singh, A. K., Taron, J., Walther, M., Wang, W., Watanabe, N., Wu, Y., Xie, M., Xu, W. and Zehner, B. OpenGeoSys: An open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/c) processes in porous media. *Environmental Earth Sciences*. 1 September 2012. Vol. 67, no. 2, p. 589–599. DOI 10.1007/s12665-012-1546-x. Available from: https://doi.org/10.1007/s12665-012-1546-x

35. L. Kaczmarczyk, K. Lewandowski, Z. Ullah. MoFEM-v0.5.42.. March 2017. Available from: https://doi.org/10.5281/zenodo.438712http://mofem.eng.gla.ac.uk/mofem/html/

36. Salvo, Maxime, Sercombe, Jérôme, Ménard, Jean-Claude, Julien, Jérôme, Helfer, Thomas and Désoyer, Thierry. Experimental characterization and modelling of UO2 behavior at high temperatures and high strain rates. *Journal of Nuclear Materials*. January 2015. Vol. 456, p. 54–67. DOI 10.1016/j.jnucmat.2014.09.024. Available from: http://www.sciencedirect.com/science/article/pii/S002231151400614X

37. Salvo, Maxime, Sercombe, Jérôme, Helfer, Thomas, Sornay, Philippe and Désoyer, Thierry. Experimental characterization and modeling of UO2 grain boundary cracking at high temperatures and high strain rates. *Journal of Nuclear Materials*. May 2015. Vol. 460, p. 184–199. DOI 10.1016/j.jnucmat.2015.02.018. Available from: http://www.sciencedirect.com/science/article/pii/S0022311515001130

38. Poilâne, C., Cherif, Z. E., Richard, F., Vivet, A., Ben Doudou, B. and Chen, J. Polymer reinforced by flax fibres as a viscoelastoplastic material. *Composite Structures*. 1 June 2014. Vol. 112, p. 100–112. DOI 10.1016/j.compstruct.2014.01.043. Available from: http://www.sciencedirect.com/science/article/pii/S0263822314000567

39. Kopp, Joachim. Efficient numerical diagonalization of hermitian 3x3 matrices. *International Journal of Modern Physics C*. March 2008. Vol. 19, no. 3, p. 523–548. DOI 10.1142/S0129183108012303. Available from: http://arxiv.org/abs/physics/0610206

40. Kopp, Joachim. Numerical diagonalization of 3x3 matrices.. 2017. Available from: https://www.mpi-hd.mpg.de/personalhomes/globes/3x3/

41. DIANA FEA BV. Diana fea solver.. 2018. Available from: https://dianafea.com/

42. Livermore Software Technology Corporation. LS-dyna.. 2018. Available from: http://www.lstc.com/products/ls-dyna