MFrontInterface.jl MFront material models in JuliaFEM

T. Frondelius, T. Helfer, I. Yashchuk, J. Vaara, A. Laukkanen

Proceedings of the 32nd Nordic Seminar on Computational Mechanics
H. Koivurova and A.H. Niemi (Eds.)

MFrontInterface.jl: MFront material models in JuliaFEM

Tero Frondelius[1,3], Thomas Helfer[4], Ivan Yashchuk[2], Joona Vaara[3], and Anssi Laukkanen[2]

(1)University of Oulu, firstname.lastname@oulu.fi
(2)VTT Technical Research Centre of Finland, firstname.lastname@vtt.fi
(3)Wärtsilä Research & Development and Engineering, firstname.lastname@wartsila.com
(4)CEA, DEN, DEC, Cadarache F-13108 Saint-Paul-Lez-Durance, France, thomas.helfer@cea.fr

Summary. This article describes the `MFrontInterface.jl` package which aims at allowing mechanical behaviours generated by the `MFront` code generator to be used in `JuliaFEM`. `MFrontInterface.jl` is build on top of the `MFrontGenericInterfaceSupport` library.

Introduction

This article describes the `MFrontInterface.jl` package which aims at allowing mechanical behaviours generated by the `MFront` code generator to be used in `JuliaFEM`. `MFrontInterface.jl` is build on top of the `MFrontGenericInterfaceSupport` library. See Figure 1 for calrification.

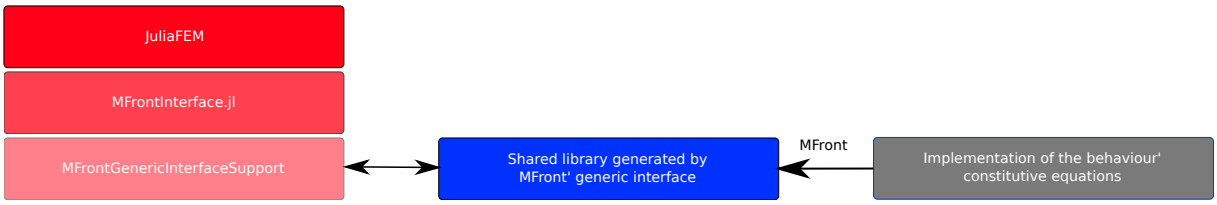

Figure 1: Sofware layers involved

A short introduction to `MFront` and `MGIS`

Overview of the `MFront` *code generator*

The behaviour of solid materials is modelled using so-called constitutive equations which describe how the internal state of the material evolves. Those state variables can describe many microstructural aspects of the material (i.e. grain size, dislocation density, hardening state) or be considered as purely phenomenological.

More precisely, after discretization in time of the problem, the solver provides an estimate of the local change of loading of the material (for example the strain increment for small strain behaviours) to the constitutive equations which allow:

- The computation of the values of the internal state variables at the end of a time step knowing their values at the beginning of the time step.

- The local thermodynamic forces (for example the stress for small strain behaviours) which affects the material equilibrium at the structural scale.

This step is called behaviour integration. In the following, the term behaviour becomes a synonym for constitutive equations. `MFront` is an open-source and cross-platform code generator dedicated to various material knowledge, such as material properties, behaviours and simple physical models, see [1, 2] for details. `MFront` is developed in the framework of the `PLEIADES` project which is co-developed by CEA, EDF and Framatome.

`MFront` provides a set of domain-specific languages on top of `C++` which allows the code of the constitutive equations to be fairly close to the mathematical expressions. Concerning mechanical behaviours, which is the most exciting aspect of `MFront` regarding this paper, `MFront` allows the user to implement, small and finite strain behaviours, as well as cohesive zone models. Isotropic and orthotropic behaviours are supported.

Thanks to the notion of `interfaces`, code specific to various mechanical solvers can be generated. As of version 3.2, `MFront` provides interfaces for e.g. `Cast3M`, `code_aster`, `Europlexus`, `Cyrano`, `CalculiX`, `Abaqus/Standard` and `Abaqus/Explcit`, `AMITEX_FFTP`, `Ansys`. The generated code is then compiled a shared library which can be plugged in the targeted solver.

An interface named `generic` has been introduced in version 3.2. Behaviours generated with the `generic` interface are meant to be used through the `MFrontGenericInterfaceSupport` library, which is described hereafter. `MFront` has been carefully designed to meet the high-quality standards of the nuclear industry. In particular, an extensive set of unit tests has been set up to guarantee its reliability.

Overview of the `MGIS` *library*

This `MFrontGenericInterfaceSupport` library (`MGIS`) aims at proving tools (functions, classes, bindings, etc...) to handle behaviours written using `MFront` generic interface, see [3–5] for details. Those tools are meant to be used by (`FEM`, `FFT`, etc.) solver developers. Permissive licences have been chosen to allow integration in open-source and proprietary codes. In particular, the `MGIS` library provides:

- functions to load behaviour from a shared library. Along with a pointer to the function implementing the behaviour, various metadata are also retrieved, such as the number of state variables, their nature (scalar, tensorial), their name. Those metadata can be used to properly allocate the storage of the internal state variable, check that the user input file is consistent (i.e. does not try to initialize a non-existing variable), provides additional information on which variable can be post-processed.
- data structures to store the material states at the beginning of the time and at the end of the time step for one integration points or a set of integration points. Views data structure are available to use externally allocated memory.
- functions to perform behaviour integration over one time step. If a set of integration points is handled, parallelization of the behaviour, integration can be handled by the `MGIS` library.

`MGIS` is written in `C++` and provides bindings for `C`, `Fortran 2003`, `python` and `Julia` [6] thanks the `CxxWrap.jl` library. The `Julia` bindings are written in pure `C++` and are barely usable per se.

The `MFrontInterface.jl` package

The `MFrontInterface.jl` package wraps `MGIS`' `Julia` bindings to make its usage much more convenient and consistent with the `Julia` language, in other words make `MFront` material models

available in `JuliaFEM` [7–11]. `BinaryBuilder.jl` package is used for binary dependencies, which makes the package installation easy and convenient, like shown below:

```
(v1.2) pkg> add MFrontInterface

julia> using MFrontInterface
```

After adding and using, the following example shows how to load a behaviour in a shared library and how to retrieve some of the metadata:

```
julia> b = load("data/libBehaviour.so","Norton", mbv.Tridimensional)
behaviour Norton in shared library data/libBehaviour.so for modelling hypothesis
Tridimensional generated from Norton.mfront using TFEL version: 3.3.0-dev.

julia> get_parameters(b)
11-element StringsVector
 epsilon
 YoungModulus
 PoissonRatio
 ....
julia> get_external_state_variables(b)
1-element VariablesVector
 Temperature
```

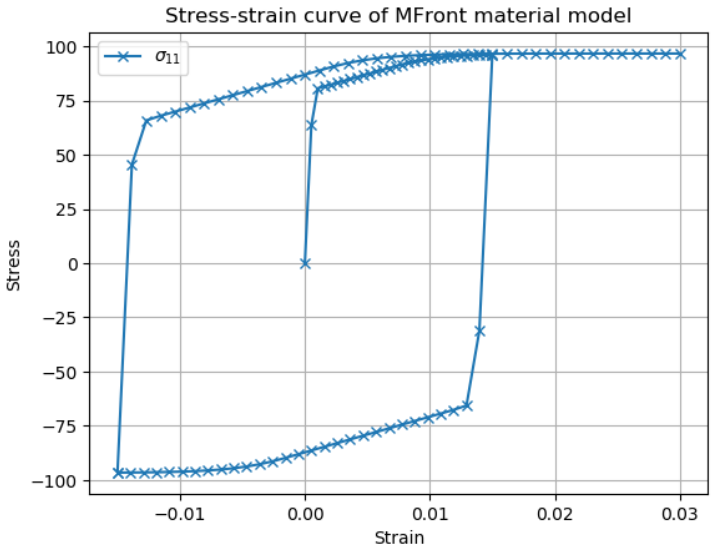

Figure 2: Stress-strain curve of MFront material model calculated with `MFrontInterface.jl`

The `MFrontInterface.jl` package already allows calling the behaviour on one time step. This has been used to build the stress-strain curve depicted in Figure 2 for a simple plastic behaviour following a kinematic hardening rule introduced by Chaboche et al. [12].

Conclusions

This short extended abstract shows the potential of `MFrontInterface.jl` package. Development has started but like all open source projects we would appreciate any kind of contributions gracefully. Next step is to get all modified and created packages registered. This will guarantee a nice user experience.

Acknowledgements

The authors would like to acknowledge the financial support of Business Finland in the form of a research projects ISA Wärtsilä Dnro 7734/31/2018, and ISA VTT Dnro 7980/31/2018. In addition, this research was supported financially by the CEA in the framework of the 'PLEIADES' project.

References

[1] CEA and EDF. MFront Web Site, 2019.

[2] Thomas Helfer, Bruno Michel, Jean-Michel Proix, Maxime Salvo, Jérôme Sercombe, and Michel Casella. 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*, 2015.

[3] Thomas Helfer. The MFrontGenericInterfaceSupport project.

[4] Thomas Helfer. MGIS. original-date: 2018-06-20T12:41:49Z.

[5] Thomas Helfer. A brief introduction to the MGIS C++ library for mechanical behaviours.

[6] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[7] Tero Frondelius and Jukka Aho. JuliaFEM —open source solver for both industrial and academia usage. *Rakenteiden Mekaniikka*, 50(3):229–233, 2017. open access.

[8] Marja Rapo, Jukka Aho, and Tero Frondelius. Natural frequency calculations with JuliaFEM. *Rakenteiden Mekaniikka*, 50(3):300–303, 2017. open access.

[9] Marja Rapo, Jukka Aho, Hannu Koivurova, and Tero Frondelius. Implementing model reduction to the JuliaFEM platform. *Rakenteiden Mekaniikka*, 51(1):36–54, 2018. open access.

[10] Marja Rapo, Joona Vaara, Jukka Aho, Teemu Kuivaniemi, Niclas Liljenfeldt, Antti Vuohijoki, and Tero Frondelius. Pipe routing optimization to avoid vibration problems by using juliaFEM. *Rakenteiden Mekaniikka*. open access. ACCEPTED MANUSCRIPT.

[11] Jukka Aho, Ville Jämsä, Teemu Kuivaniemi, Niclas Liljenfeldt, and Tero Frondelius. JuliaFEM 2D and 3D beam element implementation. *Rakenteiden Mekaniikka*. open access. ACCEPTED MANUSCRIPT.

[12] J. L. Chaboche, P. Kanouté, and F. Azzouz. Cyclic inelastic constitutive equations and their impact on the fatigue life predictions. *International Journal of Plasticity*, 35:44–66, 2012.