




The MFrontGallery project

Thomas Helfer ¹, Maxence Wangermez ¹, Eric Simo ^{2,3}, Thomas Nagel ^{3,4}, Christian B. Silbermann ³, and Lorenzo Riparbelli ⁵

1 CEA, DES, IRESNE, DEC, Cadarache F-13108 Saint-Paul-Lez-Durance, France **2** BGE TECHNOLOGY, Eschenstrasse 55, 31224 Peine, Germany **3** Geotechnical Institute, Technische Universität Bergakademie Freiberg, Gustav-Zeuner-Str. 1, 09599 Freiberg, Germany **4** Department of Environmental Informatics, Helmholtz Centre for Environmental Research – UFZ, Permoserstr. 15, 04318 Leipzig, Germany **5** University of Florence, Dagri Dept., Florence, Italy.

DOI: [10.21105/joss.07742](https://doi.org/10.21105/joss.07742)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Sarath Menon](#)  

Reviewers:

- [@mrBuché](#)
- [@likask](#)

Submitted: 07 August 2024

Published: 23 May 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Introduction

MFront is an open-source code generator focused on material knowledge ([CEA, EDF, 2022](#); [Helfer et al., 2015](#)) developed collaboratively by the French Alternative Energies and Atomic Energy Commission (CEA) and Électricité de France (EDF). The open-source status of MFront has led to its adoption in a wide range of applications¹ covering a variety of materials (ceramics, metals, concrete, woods, etc.) and physical phenomena (viscoplasticity, plasticity, damage, etc.). It is also part of the PLEIADES numerical platform ([Bernaud et al., 2024](#)), which is devoted to multi-physics nuclear fuel simulations and is developed by CEA and its industrial partners EDF and Framatome.

MFront provides so-called interfaces to ensure that material knowledge is portable and can be used in a wide range of contexts. For instance, MFront-based mechanical behaviours can be compiled for use with various commercial and academic solvers such as: [Cast3M](#), [code_aster](#), [Europlexus](#), [Abaqus/Standard](#), [Abaqus/Explicit](#), [Ansys](#), [AMITEX_FFTP](#), [CalculiX](#), [ZSet](#), [DIANA FEA](#).

Additionally, the generic interface extends the availability of these mechanical behaviours to all solvers using the [MFrontGenericInterfaceSupport projet](#) (MGIS) ([Helfer et al., 2020](#)), including: [OpenGeoSys](#) ([Bilke et al., 2019](#)), [MFEM-MGIS](#), [MANTA](#) ([Jamond et al., 2024](#)), [mgis.fenics](#), [MoFEM](#), [XPER](#) ([Perales et al., 2022](#)), etc.

The MFrontGallery project addresses the management of MFront implementations including their compilation, unit testing and deployment.

The [MFrontGallery project](#) has two main, almost orthogonal, objectives:

1. Show how solver developers may provide their users a set of ready-to-use (mechanical) behaviours that can be parametrized to meet specific needs.
2. Describe how to set up a high-quality material knowledge management project based on [MFront](#), capable of meeting the requirements of safety-critical studies.

While the first objective is common to all (mechanical) solvers, the originality of the MFrontGallery project is to address the second goal which is discussed in Section “Statement of need”.

In summary, the project provides:

- A [CMake](#) infrastructure that can be replicated in (academic or industrial) derived projects, which allows for:

¹For a comprehensive list of publications utilizing MFront, please visit: <https://thelfer.github.io/tfel/web/publications.html>

- compiling MFront sources using all supported interfaces.
- executing unit tests based on MTest which generate XML result files conforming to the JUnit standard, compatible with continuous integration platforms such as [jenkins](#).
- generating documentation associated with the stored implementations.
- A documentation of best practices for handling material knowledge implemented with MFront, such as use of consistent unit systems, bound-aware physical quantities, consistent tangent operators, and others.
- A set of high-quality MFront implementations. Those implementations are not discussed in this paper which is thus focused on the two previous points.

This paper aims to describe the MFrontGallery project and is organized as follows:

Section “Statement Of Need” discusses the necessity for a new approach to material knowledge management, particularly in the context of safety-critical studies.

Section “CMake Infrastructure provides an overview of the CMake infrastructure of the project, and details the process for creating derivative projects using the same CMake framework as the MFrontGallery.

Statement of need : material knowledge management for safety - critical studies

Role of material knowledge in numerical simulations of solids

Numerical simulations of solids are based on the description of the evolution of the thermodynamic state of materials. In the context of the MFrontGallery project, this thermodynamical state is represented at each point in space by a set of internal state variables that evolve over time due to various physical phenomena.

In MFront, material knowledge can be categorized as follows:

- **Material properties** are defined as functions of the current state of the material such as the Young’s modulus or Poisson’s ratio.
- **Behaviours** describe how a material evolves and reacts locally due to internal gradients. The material reaction is associated with fluxes (or forces) thermodynamically conjugated to gradients. For instance, Fourier’s law relates the heat flux to the temperature gradient. Mechanical behaviour in infinitesimal strain theory relates the stress and the strain and may describe (visco)elasticity, (visco)plasticity, or damage.
- **Point-wise models** describe the evolution of some internal state variables without considering gradients (i.e. with the evolution of other state variables), such as phase transition, swelling under irradiation or shrinkage due to dessication.

Requirements related to safety-critical studies

The MFrontGallery project has been developed to address various issues related to material knowledge management in safety-critical studies:

- **Intellectual property:** Material knowledge often embodies industrial know-how that must be kept confidential. This includes highly valuable mechanical behaviours derived from extensive experimental testing in dedicated facilities. MFrontGallery supports creating private derived projects to protect such valuable knowledge, as detailed in Section “Creation of a derived project”.
- **Portability:** safety-critical studies may involve several partners which use different solvers for independent assessment and review. From a single MFront source file, MFrontGallery can generate shared libraries compatible with all the solvers of interest. Moreover, the

project uses best practices guidelines² to ensure that a given MFront implementation can be shared among several teams while assuring quality.

- **Maintainability over decades:** Long-term projects require that both solvers and material knowledge evolve while ensuring past studies remain accessible and reproducible. In the authors' experience, having a dedicated material knowledge project based on *self-contained* implementations, facilitate maintainability as discussed in Section "Implementations and classification".
- **Progression of the state of the art:** safety-critical studies must reflect current scientific and engineering advancements. Thus, material knowledge, numerical methods, and software engineering need to evolve while guaranteeing the quality assurance of past, present and future simulations.
- **Continuous integration and unit testing:** Each implementation includes unit tests to prevent regression during the MFront development.
- **Documentation:** the project can automatically generate the documentation associated with the various implementations. Implementations of material knowledge can be associated to essential meta data.

Implementations and classification

MFront implementations can be classified into two main categories:

- **self-contained implementations** that contain all necessary physical information (e.g., model equations and parameters).
- **generic implementations** for which the solver is required to provide additional physical information to the material treated, e.g. the values of certain parameters. Those "generic" implementations are usually provided with solvers as ready-to-use behaviours.

Thus, self-contained implementations describe both constitutive equations **and** material coefficients identified on a well-defined set of experiments for a particular material, while generic implementations describe only the constitutive equations.

In practice, the physical information described by self-contained implementations may be more complex than a set of material coefficients. For example, the Young's modulus of a material may be defined by an analytical formula and cannot thus be reduced to a set of constants. This analytical formula shall be part of a self-contained mechanical behaviour implementation. Of course, this analytical formula could be included in the set of constitutive equations and parametrized to retrieve a certain degree of generality. In our experience, such a hybrid approach is fragile, less readable and cumbersome. Moreover, it does not address the main issue of generic behaviours which is the management of the physical information in a reliable and robust manner.

In the authors' experience, self-contained behaviours allows to **decouple the material knowledge management from the development (source code) of the solvers of interest** and thus allow a proper material knowledge management strategy suitable to meet the requirements depicted on Section "Requirements related to safety-critical studies".

State of the field

The project is focused on Quality Assurance issues related to material knowledge management which is, in the authors' experience, seldom a major concern in most open-source thermo-mechanical solvers. Several libraries providing high quality implementations of constitutive equations are available, but generally dedicated to one specific solver. The implementations are generally generic (as opposed to self-contained). The MFrontGallery project thus provide a unique approach.

²<https://thelfer.github.io/MFrontGallery/web/best-practices.html>

The CMake infrastructure

This section provides an overview of the [CMake](#) infrastructure of the MFrontGallery project.

This infrastructure is fully contained in the `cmake/modules` directory, the file `cmake/modules/mfm.cmake` being the main entry point.

Section “Main functions” describes the main CMake functions provided by the project.

Section “Creation of a derived project” shows how to create a derived project.

Main functions

The [CMake](#) infrastructure provides:

- functions used to compile MFront files related to material properties, behaviours and models.
- functions related to unit testing. Those functions will not be described in this paper.
- functions related to documentation and website generation. Those functions will not be described in this paper.

The following snippet shows how the `mfront_properties_library` introduces a set of libraries describing the material properties of Uranium dioxide:

```
mfront_properties_library(U02
    U02_YoungModulus_Martin1989
)
```

The `mfront_behaviours_library` and `mfront_models_library` are available for behaviours and point-wise models respectively.

Creation of a derived project

This section describes the process for setting up a new project based on the [CMake infrastructure](#) of the MFrontGallery project.

Fetching `cmake/modules` directory

To create a material management project derived from MFrontGallery, just copy the contents of the `cmake/modules` directory in your local directory.

If you intend to use `git` for version control, one easy way is to add the MFrontGallery as a remote ressource and check out the CMake repository from it as follows:

```
$ git remote add MFrontGallery https://github.com/thelfer/MFrontGallery
$ git fetch MFrontGallery master
$ git checkout MFrontGallery/master --cmake
```

Of course, the user may also want to follow another branch rather than the master branch.

Top-level `CMakeLists.txt` file

The next step consist of creating a top level `CMakeLists.txt` file. Here is a minimal example:

```
project(NewMaterialManagementProject)
set(PACKAGE new-material-management-project)
cmake_minimum_required(VERSION 3.0.2)

include(cmake/modules/mfm.cmake)

# testing
```

```
set(CTEST_CONFIGURATION_TYPE "${JOB_BUILD_CONFIGURATION}")  
enable_testing()
```

```
# add subdirectories here  
add_subdirectory(materials)
```

Now you are ready to create the subdirectory materials containing your MFront files.

Conclusions

The MFrontGallery project is dedicated to material knowledge management in safety-critical studies and is the result of long-standing experience gathered in the PLEIADES project. Key concepts built upon are portability, maintainability and reproducibility over long time periods, continuous integration and unit testing, documentation and the safeguarding of intellectual property as well as attribution. Based on the technical infrastructure described in this article, it becomes possible to set up derived projects in similar contexts where these concepts are considered relevant.

Acknowledgements

This research was conducted in the framework of the PLEIADES project, which is supported financially by the French Alternative Energies and Atomic Energy Commission (CEA), Électricité de France (EDF) and Framatome.

TN acknowledges funding provided by the European Joint Programme on Radioactive Waste Management EURAD “European Partnership on Radioactive Waste Management” (Grant Agreement No 101166718), the Federal Ministry for Education and Research (BMBF) under grant 03G0927B for the DigBen project and thanks Projektträger Jülich (PtJ) for support. CBS acknowledges the German Federal Ministry for Economic Affairs and Energy and the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection for funding the projects Sandwich-HP (02E11799C) and Sandwich-HP2 (02E12163C) and thanks the project management agency of Karlsruhe.

References

- Bernaud, S., Ramière, I., Latu, G., & Michel, B. (2024). PLEIADES: A numerical framework dedicated to the multiphysics and multiscale nuclear fuel behavior simulation. *Annals of Nuclear Energy*, 205, 110577. <https://doi.org/10.1016/j.anucene.2024.110577>
- Bilke, L., Flemisch, B., Kalbacher, T., Kolditz, O., Helmig, R., & Nagel, T. (2019). Development of Open-Source Porous Media Simulators: Principles and Experiences. *Transport in Porous Media*, 130(1), 337–361. <https://doi.org/10.1007/s11242-019-01310-1>
- CEA, EDF. (2022). *TFEL/MFront website*. <https://thelfer.github.io/tfel/web/index.html>
- Helfer, T., Bleyer, J., Frondelius, T., Yashchuk, I., Nagel, T., & Naumov, D. (2020). The ‘MFrontGenericInterfaceSupport’ project. *Journal of Open Source Software*, 5(48), 2003. <https://doi.org/10.21105/joss.02003>
- Helfer, T., Michel, B., Proix, J.-M., Salvo, M., Sercombe, J., & Casella, M. (2015). 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*, 70(5), 994–1023. <https://doi.org/10.1016/j.camwa.2015.06.027>

Jamond, O., Lelong, N., Brooking, G., Helfer, T., Prabel, B., Prat, R., & Jaccon, A. (2024, May). MANTA: An industrial-strength open-source high performance explicit and implicit multi-physics solver. *16ème Colloque National En Calcul de Structures*. <https://hal.science/hal-04610968>

Perales, F., Socie, A., Nkoumbou Kaptchouang, N. B., Dubois, F., Monerie, Y., Mozul, R., Vincent, P. G., & Babik, F. (2022). *XPER : Une plateforme pour la simulation numérique distribuée d'interactions multiphysiques entre corps*. <https://hal.science/hal-03704427>