# MFEM-MGIS-MFRONT, a HPC mini-application targeting nonlinear thermo-mechanical simulations of nuclear fuels at mesoscale

G. LATU1, T HELFER1, S. BERNAUD1, G. FOLZAN2

1CEA DES/IRESNE/DEC/SESC/LSC, St-Paul-Lez-Durance, France

2CEA DES/ISAS/DM2S/SEMT/LM2S, Gif sur Yvette, France

Email contact of corresponding author: guillaume.latu@cea.fr

Objectives of MFEM-MGIS-MFRONT mini-application

The MFEM-MGIS-MFRONT mini-application (abbreviated version MMM), aims at efficiently use supercomputers in the field of implicit nonlinear thermomechanics. This open-source library is based on several components as prerequisites. The first component, MFEM [1,6], is a finite element library designed for current supercomputers but also for the upcoming exascale supercomputers. It provides many useful features for carrying out realistic simulations: support for curvilinear meshes, high order approximation spaces and different families of finite elements, interfaces to several types of parallel solvers (including matrix-free ones), preconditioners, and native support for adaptive non-conforming mesh refinement (AMR).

Originating from the applied mathematics and parallel computing communities, MFEM offers both performance and a large panel of advanced mathematical features. In particular, one can easily switch from one linear solver to another (direct or iterative), which is essential for the targeted application: microstructure and mesoscale modelling for nuclear fuel. However, applications to solid mechanics in MFEM are mostly limited to simple constitutive equations such as elasticity and hyperelasticity, which is insufficient.

The aim of the MMM project is to combine MFEM with the MFrontGenericInterfaceSupport (MGIS) project [4,8], an open-source C++ library that provides convenient data structures to support arbitrarily complex nonlinear constitutive equations generated by the open-source MFront code generator [2,5]. This library handles all the kinds of behaviour supported by MFront. In the field of nonlinear mechanics, this encompasses arbitrary complex behaviours that can describe damage, plasticity, viscoplasticity, phase change in small and finite strain analyses. Generalised behaviours such as strongly coupled thermomechanical behaviours, variational approaches to fracture, Cosserat media are supported by MGIS and will be considered in future versions of MMM.

Thanks to the coupling with MGIS and MFront and specific developments made, MMM has added the following mechanical features compared to pure MFEM approach:

– Ability to handle several materials which distinct constitutive equations.

– Support for internal state variables (defined at quadrature points).

– Support for complex boundary conditions specific to nonlinear mechanics.

– Support for post-processing specific to nonlinear mechanics.

– Integration with the Salomé platform, support of the MED mesh [12] file format on input and export of results in the same format on output.

The implementation of high order meshes or finite elements is easy. The library tackles some peculiarities of nonlinear mechanics. In particular, the support of complex constitutive laws and the management of advanced boundary conditions.

The MMM library [7] is written in C++-17 language and provides a very high level of abstraction based on a very declarative text-based Application Programming Interface. In the future, python bindings will be also considered to improve the user experience.

Motivations for an open-source framework

The mini-app takes full advantage of an open-source software (OSS) stack. It benefits from the increasing maturity of many communities and tools working together. Thus, within MFEM, one has many available choices to set the linear solver, such as: Hypre, PETSc, MUMPS, SuperLU, UMFPACK or other ones. Likewise, several preconditioners, partitioning libraries, or input mesh formats can be activated and used. Combinations are highly configurable and almost all external libraries are switchable. To handle the numerous accessible combinations, Spack is really a cornerstone. This package manager simplifies building, installing, customizing, and sharing HPC software stacks. It provides a simple way for installing packages with cumbersome structures and lots of dependencies. Spack is an open-source package manager developed and maintained by community of HPC developers. Our setup that combines OSS allows for a fast and cheap access to advanced features embedded in the underlying libraries. At the beginning of MMM mini-app, a preliminary study considering several potential finite element libraries leads us to the conclusion that choosing the right open-source stack was a key point. For example, to achieve both performance and keep low maintenance/development costs, one should ensure that components of the software stack can connect well with each other, and offer a high degree of controllability. One must also looks at the maturity of each library and the quality of technical support proposed. The responsiveness of the support is important and is often related to the size of the community behind each software. Coding practices, documentation and expected durability of the libraries have also to be considered. Then, pondering all these points in our analysis, assessing the risks and in estimating the potential gains, we chose the most suitable software stack serving our purpose.

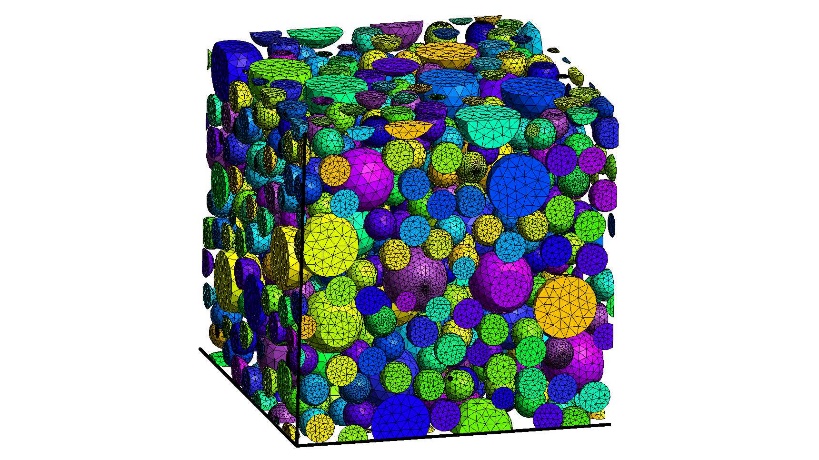
Compared to closed source development that we might have considered, OSS brings regular bug fixing, peer review of code by many developers, a community that tests and improves the software frequently. The proposed services such as error reporting, bug tracker, forum for user requests and offered support help a lot for speeding up the development process and shortening maintenance actions.

High-performance simulations suffer from difficulties for ensuring exact numerical reproducibility. Numerical results that are not reproducible make it difficult to assess the methods and novelties a given software produces. In addition to that, the available publications typically lack the level of detail that is required to reproduce simulations. Also, with prototype realizations often remaining private, other people are unable to track the code. Then, one should find ways to improve this issue. Several reasons explains why reproducibility is difficult: floating-point arithmetic peculiarities, non-determinism of parallel calculations, complex software stacks, closed source software. The two first points are well beyond the scope of this contribution. However, the last two points can be partially solved with open-source solutions. Hence, assembling software, connecting tools together into pipelines, and specifying parameters can be handled. Existing tools can automate a series of processes. For MMM, we choose to rely on Spack to completely organize and setup the versions of the different libraries composing the stack. Furthermore, the CMake tool helped us to achieve reproducible builds across multiple computers and systems. A set of reference solutions has also been designed. Then, we can check on any new target machine if MMM gives correct results using Spack and CMake. Verification through defined standard examples ensure quite fast deployment, these examples also serve as documentation for users of the code. It is also foreseen that we supplement the current strategy with a continuous integration platform to further improve the robustness of the mini-app.

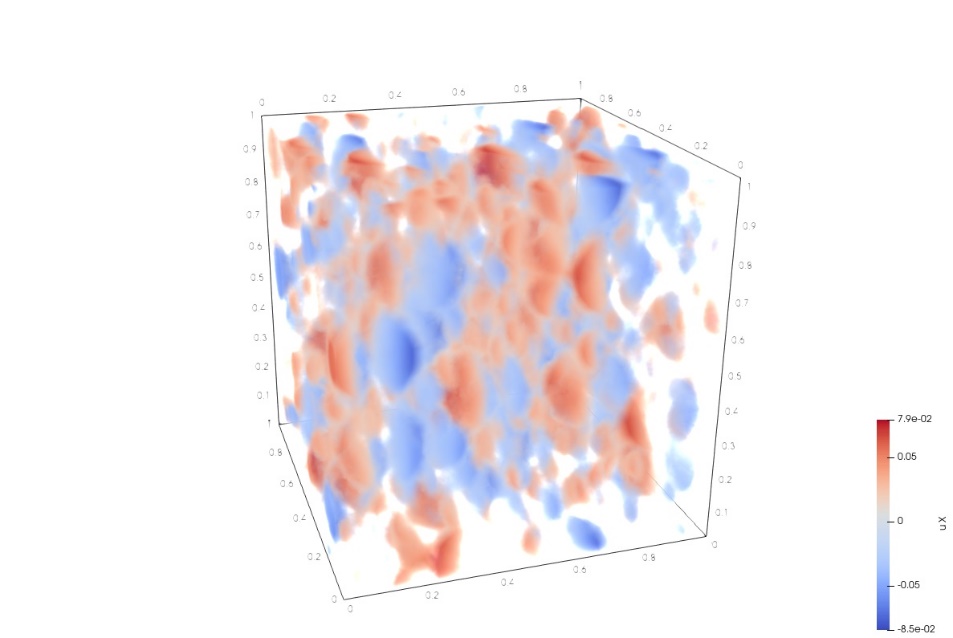
Open-source approach enable transparency, accessibility, and replicability among contributors of a single code. Modelling effort can include participants with different domain expertise. This diversity of contributions really helps building up software that combine state-of-the-art numerical methods. Our goal is to establish international collaborations with privileged partners and with academics around MMM. In this context, open-source approach is fostering a greater adoption of the software product. We also expect that this solution can favour external contributions and promote collaborations focusing on grain and subgrain-resolved microstructure for nuclear fuel modelling.

Numerical results

Installation and deployment on desktop or large computers is based on the Spack package manager [10]. With MMM, we were able to carry out a multi-material elastic modelling on computing clusters. Scalability performance is good on a few thousands of CPU cores. Despite the very high level of abstraction and the genericness of MMM (multi-material and arbitrary behaviours), the overhead appears reasonably limited, roughly 30% compared to a pure MFEM version which provides very optimised and specialised kernels (this was tested on an elastic setting with 2 materials). The presentation will show some of the current capabilities of the software.



*FIG. 1. Periodic mesh generated by the GMSH mesher, model of 544 inclusions (polydisperse) within a periodic Representative Volume Element.*



*FIG. 2. Elastic modelling of two materials with different characteristics: a large matrix containing many inclusions. Displacements in x direction (from left to right direction) are shown.*

For several use cases, MMM has good strong scaling [performance.W](http://performance.Up)e have canducted benchmarks on two testbeds: Marseille supercomputing center (up to 512 cores) and CEA/CCRT center (up to 2000 cores). Up to now we have been considering calculations of periodic micro-structures dedicated to the study of the behavior of nuclear fuels [3, 9]. Presently, grain and subgrain-resolved microstructure models are the main targets of the code, see Fig. 1 and 2 for a result of a simulation conducted with MMM on a Representative Volume Element. This require to be able to handle periodic boundary conditions at reasonable cost, which is achieved thanks to MFEM that embeds a customized elimination methods (modification of the global stiffness matrix). Simulation approaches that explicitly take all relevant microstructural details into account are the only modelling approaches that have sufficient predictive capability for several applications related to the nuclear fuel field at mesoscale. For such requirements, we need computationally fast and robust numerical algorithm to handle 3D volumes. But we require also physics-based constitutive models that are able dealing with complex microstructure in full 3D. MMM has been designed recently to reach exactly this goal and to tackle large systems with several million unknowns or even a few billion.

Conclusion

The MMM mini-app is a new HPC modelling software. Based on an open-source software stack, it allows for the fine representation of microstructure in full 3D in the field of fuel modelling. On the one hand, MGIS and MFRONT bring nonlinear mechanics features such as damage, plasticity, viscoplasticity capabilities. On the other hand, MFEM provides advanced finite elements schemes and parallel performance which has been checked on several thousands of cores until now. Open-source approach was chosen mainly to: promote collaboration, improve reproducibility, and reduce costs for development and maintenance.

References

[1] ANDERSON R. & al. “MFEM : A modular finite element methods library”.   
Computers and Mathematics with Applications (2021) 81:42–74.   
URL: <http://www.sciencedirect.com/science/article/pii/S0898122120302583>

[2] CEA and EDF. TFEL/MFront.   
URL: <https://tfel.sourceforge.net/>

[3] ESNOUL C., LARGENTON R. & al. “Studying fuel failure behavior with a micromechanical approach”.   
The 12th World Congress on Computational Mechanics (Proc. WCCM 12, 2016).   
URL: <https://hal.archives-ouvertes.fr/hal-01474287>

[4] HELFER T., BLEYER J. & al.. The ‘MFrontGenericInterfaceSupport‘ project.   
The Open Journal (2003) 5-48.   
URL: <https://doi.org/10.21105/joss.02003>

[5] HELFER T., BRUNO M. & al. “Introducing the open-source MFront code generator : Application to mechanical behaviours and material knowledge management within the PLEIADES fuel element modelling platform”.  
Computers and Mathematics with Applications (2015) 70-5:994–1023.   
URL: <http://www.sciencedirect.com/science/article/pii/S0898122115003132>

[6] MFEM : Modular finite element methods [Software].   
URL: <https://mfem.org>

[7] The MFEM-MGIS project [Software].   
URL: <https://github.com/thelfer/mfem-mgis>

[8] The MFrontGenericInterfaceSupport project [Software].   
URL: <https://github.com/thelfer/MFrontGenericInterfaceSupport>

[9] PORTELETTE L. & al. “Athermal dislocation strengthening in UO2”.   
Journal of Nuclear Materials (2020).   
URL: https://hal. archives-ouvertes.fr/hal-02938715.

[10] Spack package manager [Software].   
URL: <https://spack.io/>

[11] BIGOT J., LATU G. et al. “An approach to increase reliability of HPC simulation, application to the Gysela5D code”   
Esaim: Proceedings (2016) 53:248-270.  
URL: <https://www.esaim-proc.org/articles/proc/pdf/2016/01/proc165315.pdf>

[12] CEA and EDF. Salomé platform [Software].   
URL: <https://www.salome-platform.org/user-section/about/med>