The Journal of Open Source Software

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

Thomas Helfer [1], **Guillaume Latu** [1], **Raphaël Prat** [1], **Maxence Wangermez** [1], **and Francesca Cuteri** [1]

1 CEA, DES, IRESNE, DEC, Cadarache F 13108 St Paul Lez Durance

DOI: 10.21105/joss.07719

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @iammix
- @jacobmerson

Submitted: 21 October 2024
Published: 10 April 2025

Summary

The MFEM/MGIS application aims to efficiently use supercomputers in order to describe coupled multiphysics phenomena with a particular focus on thermo-mechanics. The authors primarily aim at describing the nuclear fuels at mesoscale (see example below), but MFEM/MGIS is versatile and can address more general cases.

This open-source library is based on several components as prerequisites: the MFEM (Modular Finite Element Methods) (Anderson et al., 2021; Kolev & Dobrev, 2010) library, the MGIS (MFront Generic Interface Support) (Helfer et al., 2020) library and the MFront code generator (Helfer et al., 2015).

Thanks to the features embedded within MGIS and MFront and thanks to specific developments, MFEM/MGIS adds several mechanical features compared to a pure MFEM approach. The library tackles some peculiarities of nonlinear mechanics. In particular, the support of complex constitutive laws and the management of advanced boundary conditions.

Context and motivation

About MFEM

MFEM is a finite element library designed for current supercomputers and the upcoming exascale supercomputers. It provides many useful features for carrying out state-of-the art 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 (Bernaud et al., 2024): microstructure and mesoscale modeling for nuclear fuel.

Statement of need

The solid mechanic examples in MFEM are mostly limited to simple constitutive equations such as elasticity and hyperelasticity without internal state variables. This is insufficient to address many engineering studies and in particular complex nuclear fuel simulations.

The aim of the MFEM/MGIS project is to combine MFEM with the MFrontGenericInterfaceSupport

(`MGIS`) project, an open-source `C++` library that handles all the kinds of behaviours supported by the open-source `MFront` code generator.

In the field of nonlinear mechanics, this encompasses arbitrary complex behaviours that can describe damage, plasticity, and viscoplasticity in both small or finite strain analyses. Generalized behaviours such as variational approaches to fracture are supported by `MFEM/MGIS`.

The `MGIS` data structures are used to add support for partial quadrature functions to `MFEM`, a feature needed to store internal state variables on each material.

State of the field

Many open-source thermomechanical solvers allow handling complex mechanical behaviours. `code_aster`, `MoFEM`, `CalculiX` are examples of state of the art solvers which have an interface with `MFront`.

However, those solvers lack many features provided by `MFEM` that the authors found interesting to explore in the field of solid mechanics (see the above section for a detailed list). The authors also found interesting to take a platform designed from the start for high performance computing and adapt it to engineering needs and evaluate the resulting performances.

Overview of `MFEM/MGIS` features

The `NonLinearEvolutionProblem` class

The main class of `MFEM/MGIS` is called `NonLinearEvolutionProblem` and describes the evolution of the materials of the physical system of interest over a single time step for a given phenomenon.

Currently `MFEM/MGIS` provides built-in support for mechanics, heat transfer, and micromorphic damage.

The following snippet declares a new nonlinear evolution problem:

```
mfem_mgis::NonLinearEvolutionProblem problem(
    {{"MeshFileName", "fuel.msh"},
     {"FiniteElementFamily", "H1"},
     {"FiniteElementOrder", 6},
     {"UnknownsSize", 1},
     {"Hypothesis", "Tridimensional"},
     {"Parallel", true}});
```

As the unknown is scalar (according to the `UnknownsSize` parameter), this problem can be used to describe heat transfer or micromorphic damage, depending on the behaviour integrators declared, as explained in the next section. The `NonLinearEvolutionProblem` class supports both sequential and parallel computations and lets the user exploit a large subset of `MFEM` abilities, including the use of finite elements of arbitrary orders.

A staggered approach for multiphysics simulations can be set up by using several instances of `NonLinearEvolutionProblem`.

The `PeriodicNonLinearEvolutionProblem` class

`MFEM/MGIS` provides a specialized version for the `NonLinearEvolutionProblem` for periodic computations named `PeriodicNonLinearEvolutionProblem`.

This class allows managing the evolutions of the macroscopic gradients (strain in small strain analysis, deformation gradient in finite strain analysis, temperature gradient in heat transfer analysis) and passing them to the behaviour integrators.

Note about linear analyses

As implied by its name, the NonLinearEvolutionProblem is focused on nonlinear resolutions. Linear analyses can still be performed by using linear behaviours (generated by MFront), but with a computational overhead compared to linear analysis made with optimised kernels, such as the elastic kernels provided natively by MFEM.

In our experience, this overhead is limited and mostly comes from the extra flexibility allowed by MFEM/MGIS. For instance, MFEM elastic kernels assume that material properties (Young's modulus, Poisson's ratio) are uniform on each material.

Behaviour integrators

MFEM/MGIS allows the assignment of distinct behaviours to each material. To achieve this, a special nonlinear formulation has been implemented which delegates the computations of residual and jacobian terms on each material to so-called behaviour integrators.

Behaviour integrators are associated with a physical phenomenon and a modelling hypothesis (plane strain, plane stress).

The following snippet assigns the behaviour integrator named Mechanics to the material named beam to the mechanical non linear evolution problem named mechanics:

```
mechanics.addBehaviourIntegrator("Mechanics", "beam",
                                  "src/libBehaviour.so",
                                  "MicromorphicDamageI_SpectralSplit");
```

The behaviour MicromorphicDamageI_SpectralSplit is loaded from a library named libBehaviour.so which shall have been generated using MFront before running the simulation. The behaviour integrator Mechanics supports arbitrary small strain and finite strain behaviours.

Internally, the addBehaviourIntegrator method calls an abstract factory which instantiates a BehaviourIntegrator dedicated to the kind of behaviour selected by the user (small or finite strain) and the modelling hypothesis declared by the problem (plane strain, plane stress, tridimensional, etc.).

About the definition of material properties

Behaviours may require the user to provide properties, such as the Young's modulus, Poisson's ratio, etc.. In MFEM/MGIS, those properties can be uniform on the material or given by a partial quadrature function. The latter case allows the properties to be defined independently on each integration point, which is required if those properties depend on local material properties or the local state of the material (for instance, the local temperature).

User interface

The MFEM/MGIS library is written in C++17 language.

As the application targets mechanical engineers, it provides a high level of abstraction, focused on the physical aspects of the simulation and hiding most numerical details by default.

The API is declarative and mostly based on data structures similar to a python dictionary, hence limiting direct usage of C++. In particular, such data structures are used to instantiate non linear evolution problems, behaviour integrators, post-processings, and boundary conditions.

Post-processings

Various post-processings are available. Here are some examples of post-processings that were added to MFEM/MGIS:

- `ComputeResultantForceOnBoundary`: Compute the resultant of the inner forces on a boundary.
- `ComputeMeanThermodynamicForcesValues`: Compute the macroscopic stress and strain for each material.
- `ParaviewExportIntegrationPointResultsAtNodes`: Paraview post processing files of partial quadrature functions, like the ones associated with the internal state variables.

Tutorials

MFEM/MGIS features are described in the following tutorial: https://thelfer.github.io/mfem-mgis/tutorial.html.

Several examples can be found on the open-source GitHub repository: https://github.com/latug0/mfem-mgis-examples, including a simulation of nuclear fuel at microstructural scale using a Representative Volume Element (RVE), cf. Figure 1.

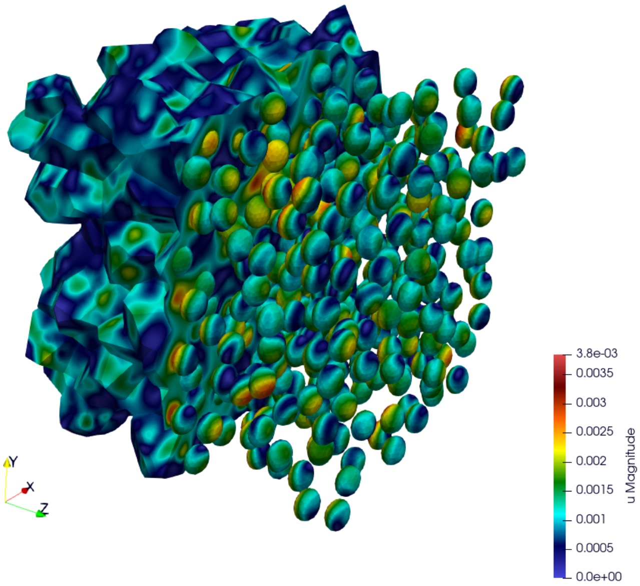

Figure 1: Simulation of a Representative Volume Element (RVE) of mixed oxide material, which contains 634 inclusions. The variable "u" represents the displacement. This simulation reproduces the results obtained by (Masson et al., 2020), who used an FFT method.

Software stack and installation process

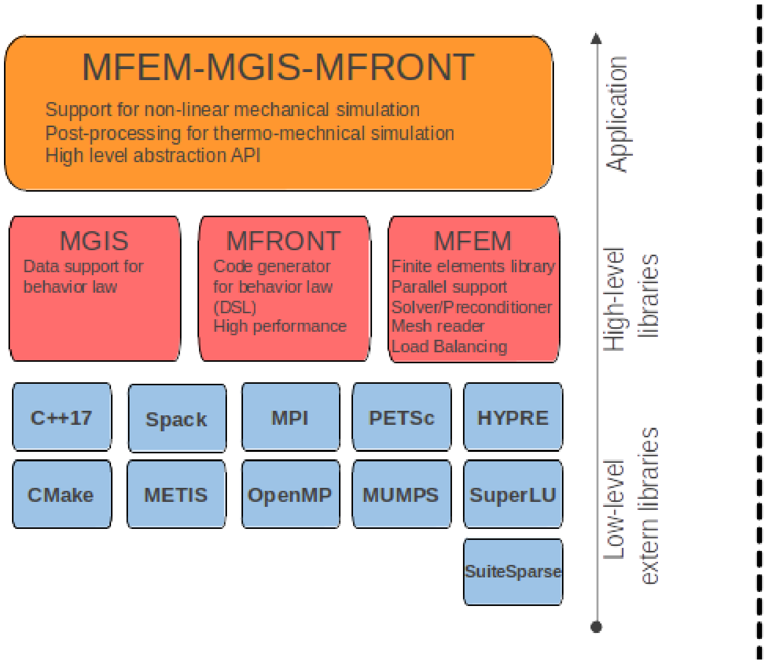

Figure 2: MFEM/MGIS Stack. Each library is open source.

The MFEM/MGIS software stack is illustrated in Figure 2. Hence, the minimal package requirements to build MFEM/MGIS on a HPC platform are typically:C++17, MFEM, MGIS, TFEL (MFront), CMake and MPI. It is important to note that MFEM includes a significant stack of several tens of packages including linear solver libraries. To handle the numerous accessible combinations, Spack (Gamblin et al., 2015) is really a cornerstone. This package manager simplifies building, installing, customizing, and sharing HPC software stacks. In the end, MFEM-MGIS offers a Spack package reusing most of the MFEM installation variants already provided in the MFEM Spack package while maintaining compatibility across package versions.

Numerical Results

Multi-material elastic modelling on computational clusters has been carried out with MFEM/MGIS. The observed scalability performance is good on a few thousands of CPU cores. Benchmarks are available in the documentation: https://thelfer.github.io/mfem-mgis/benchmark.html.

Several examples can be found on the open-source GitHub repository: https://github.com/latug0/mfem-mgis-examples.

Acknowledgement

Funded by the European Union, this work has received funding from the Open HPC thermomechanical tools for the development of EATF fuels undertaking (OperaHPC) grant agreement No 101061453.

Benchmarks and scalability tests were performed using HPC resources from CCRT funded by the CEA/DES simulation program.

References

Anderson, R., Andrej, J., Barker, A., Bramwell, J., Camier, J.-S., Cerveny, J., Dobrev, V., Dudouit, Y., Fisher, A., Kolev, Tz., Pazner, W., Stowell, M., Tomov, V., Akkerman, I., Dahm, J., Medina, D., & Zampini, S. (2021). MFEM: A modular finite element methods library. *Computers & Mathematics with Applications*, *81*, 42–74. https://doi.org/10.1016/j.camwa.2020.06.009

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

Gamblin, T., LeGendre, M., Collette, M. R., Lee, G. L., Moody, A., De Supinski, B. R., & Futral, S. (2015). The spack package manager: Bringing order to HPC software chaos. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–12. https://doi.org/10.1145/2807591.2807623

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

Kolev, T., & Dobrev, V. (2010). *Modular finite element methods (MFEM)*. https://doi.org/10.11578/dc.20171025.1248

Masson, R., Seck, M. E. B., Fauque, J., & Gărăjeu, M. (2020). A modified secant formulation to predict the overall behavior of elasto-viscoplastic particulate composites. *Journal of the Mechanics and Physics of Solids*, *137*, 103874. https://doi.org/10.1016/j.jmps.2020.103874