



MFEM-MGIS-MFRONT

**A HPC mini-application targeting  
nonlinear thermo-mechanical simu-  
lations of nuclear fuels at mesoscale**

DE LA RECHERCHE À L'INDUSTRIE

## IAEA technical meeting

22/06/2022

G. Latu<sup>1</sup>, Th. Helfer<sup>1</sup>, S. Bernaud<sup>1</sup>, G. Folzan<sup>2</sup>, R. Prat<sup>1</sup>

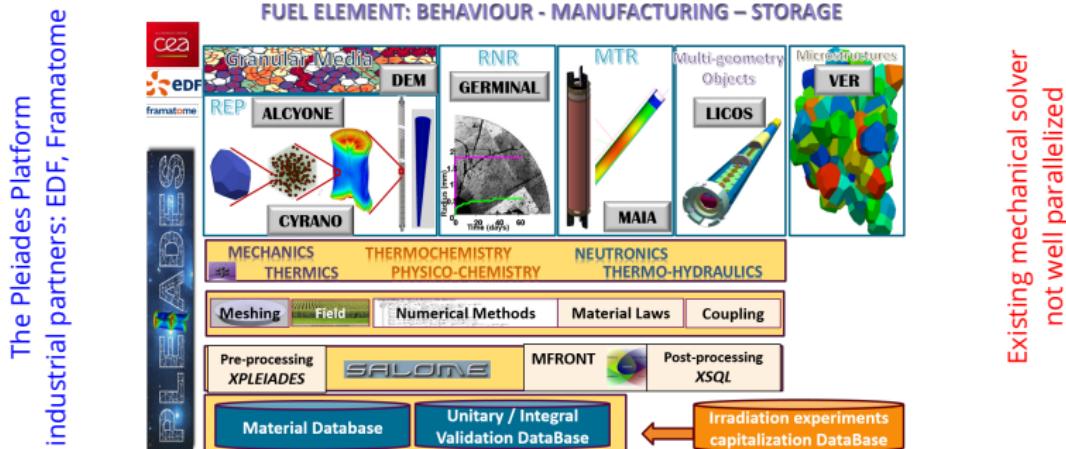
<sup>1</sup>CEA, DEC, St-Paul-Lez-Durance, France

<sup>2</sup>CEA, DM2S, Gif sur Yvette, France

- ▶ Context and goals
- ▶ A small tutorial
- ▶ Design and features of the MMM project
- ▶ Feed-backs on some issues using MFEM
- ▶ European project - OperaHPC
- ▶ Examples
- ▶ Conclusions and perspectives

## Context and goals

- ▶ French Atomic Energy Commission (CEA), public institution
- ▶ DEC department: modelling and simulation of nuclear fuel



- ▶ A wide range of materials (ceramics, metals, composites)
- ▶ A wide range of mechanical phenomena and behaviours
  - Creep, swelling, irradiation effects, phase transitions ...
- ▶ A wide range of mechanical loadings

- ▶ Build a HPC general purpose non linear multi-physics library,  
*development began end of 2020.*
  - Primary focus is **non linear solid mechanics** and **heat transfer**.
  - Expected modelling (long-term): Shells, Beams,  
Phase-field approaches of brittle fracture, micromorphic models,  
Cosserat plasticity, strongly coupled thermo-chemical-mechanical or  
thermo-hydro-mechanical phenomena.
  - Replace existing mechanical solver in Pleiades platform
  - targeted application: microstructure and mesoscale modelling for  
nuclear fuel
- ▶ A two-pillar library with *opensource* commitment (*MMM LGPL 3.0*):
  - MFEM: HPC finite element solver (*3-clause BSD*)
  - MGIS/MFront: constitutive laws, material modelling (*LGPL 3.0 / GPL 3.0*)

# Open-source orientation

## Expected benefits

### ► Exchanges and collaboration

- Facilitate sharing of reference solutions, standard benchmark problems and input data
- Foster collaboration with academics
- Allow to assess features and opportunities for integration of already developed open-source codes
- Ease the spread of the tool within industry, dodge the silos due to funding within companies

### ► Lesson learned from MFront developped at CEA opensource since 2014 (cf. poster session)

- ease the access to the tool: increase in visibility, reproducibility, credibility
- software engineering: users help to improve the software (bug report) and robustness
- other codes that use your product: opportunity for partnership, publications
- extra work, need to book time for: minimal user support, documentation, training

# A small tutorial

# What are we talking about ?

## Example of end-user API

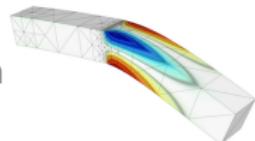
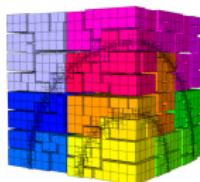
```
// loading the mesh and building the non linear problem
mfem_mgis::NonLinearEvolutionProblem problem(
    {"MeshFileName", mesh_file}, {"FiniteElementFamily", "H1"}, 
    {"FiniteElementOrder", order}, {"UnknownsSize", dim}, 
    {"Hypothesis", "PlaneStrain"}, {"Parallel", parallel}});

// associating names to element and boundary attributes (could be automated for some mesh formats)
problem.setMaterialsNames({{1, "NotchedBeam"}});
problem.setBoundariesNames({{3, "LowerBoundary"}, {4, "SymmetryAxis"}, {2, "UpperBoundary"}});
// declaring behaviour integrators
problem.addBehaviourIntegrator("Mechanics", "NotchedBeam", library, behaviour);
// setting the initial state of the materials
auto& m1 = problem.getMaterial("NotchedBeam");
mgis::behaviour::setExternalStateVariable(m1.s0, "Temperature", 293.15);
...
// defining boundary conditions, postprocessings and solver parameters
problem.addUniformDirichletBoundaryCondition({{"Boundary", "LowerBoundary"}, {"Component", 1}});
problem.addPostProcessing("ParaviewExportResults", {"OutputFileName", "ssna303-displacements"});
auto& solver = problem.getSolver();
...
// loop over time step
for (mfem_mgis::size_type i = 0; i != nsteps; ++i) {
    // updating the boundary values and resolution
    ...
    problem.solve(dt);
    problem.update();
    t += dt;
}
```

- ▶ Instantiating NonLinearEvolutionProblem Class is the main entry point
- ▶ Behaviour integrator is the main new concept of MFEM/MGIS

## **Design and features of the MMM project**

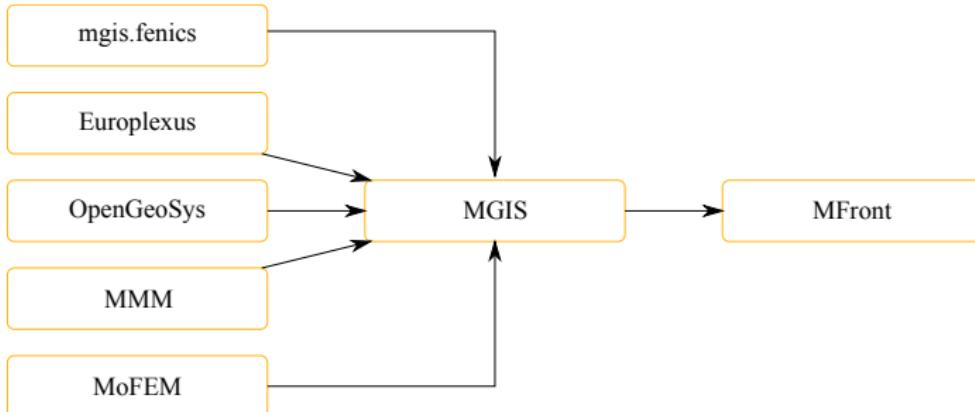
- ▶ Main features <https://mfem.org/>
  - C++, opensource FEM library. Large community
  - **High-order** finite element meshes and spaces
- ▶ **HPC** oriented
  - Scales to hundreds of thousands of cores
  - Many hardware devices are supported, such as GPUs
  - Minimal changes to switch from serial to parallel version
  - Top-notch optimizations for speeding up computations
  - Able to call many standard linear solvers (Hypre, PETSC, ...)
- ▶ Mesh features
  - Triangular, quadrilateral, tetrahedral, wedge, and hexahedral elements
  - **Periodic** meshes
  - Conforming and nonconforming adaptive mesh refinement
  - Parallel **load balancing** (several methods)
- ▶ Non linear PDE's resolution
  - handled by **non linear form integrators** which describes how to build the element contribution to the *residual* and its *jacobian*.  
Glue with MGIS is here



- ▶ Many linear solvers available in MFEM
  - direct and iterative ones
  - large set of preconditioners
  - matrix-free approach
  - GPU versions proposed
- ▶ Highly configurable through spack tool
  - ease installation and configuration of the software stack
  - helps to deal with complex settings and dependancies
  - opensource and include recent versions of software
  - bring performance and low maintenance efforts
- ▶ Software lifecycle
  - the community helps to identify bugs and to propose solutions
  - maintenance is offered
  - access to release version (every year) and master version
- ▶ Opensource
  - community of MFEM is large
    - brings extensive community support (github issues)
  - license is compatible with our usage
  - foster collaborations
    - MFEM workshop (cf s.13), EU project OperaHPC (cf s.14)



**GitHub**



- ▶ High-level API to ease the use of MFront
  - Provides classes to retrieve **metadata** from an MFront behaviour and call the behaviour integration over a time step
  - Provides way to **ease memory management**: deal with internal state variables
  - Written in C++, bindings exists for C, Fortran2003, python, Julia
- ▶ Used/tested in `mgis.fenics`, `OpenGeoSys`, `MMM`, `XPer`, `MoFEM`, `Disk++`, `Kratos Multiphysics`, `JuliaFEM`, `NairmMPM`, `esys.escript`, `DUNE`, `OOFEM` ...

## ► Statements

- Poor support of multiple materials in MFEM for solid mechanics
- No support for functions on integration points for a given material (element attribute)

## ► Improvements made by MMM

- Support for non linear behaviour integrators based on MFront
- Support for functions on integration points that depends on material identifier
- Simplified High-level API for end users (mechanics)
- Support for the MED mesh file format (needed for EDF/CEA partnership)

**TODO** Support for complex boundary conditions specific to nonlinear mechanics

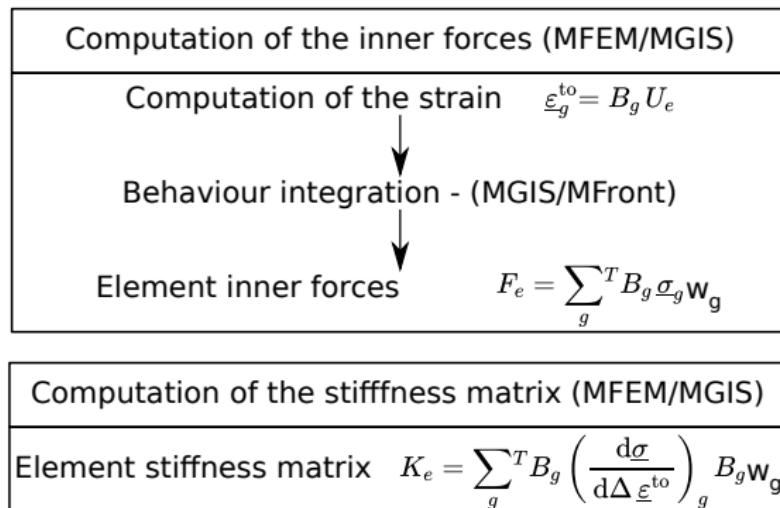
## ► In practice, MMM is a library made up of

1. Classes that inherit from MFEM
  - loops on elements, linear & non-linear solvers, mesh management...
2. Classes that inherit from MGIS
  - materials management, internal variables, fill matrix entries...
3. Procedures for the end user to interact with MFEM & MGIS

## ► MMM is a C++17 opensource library

- <https://github.com/thelfer/mfem-mgis>
- <https://github.com/latug0/mfem-mgis-examples>

# The role of behaviour integrators



- ▶ Behaviour integrators are associated with a material identifier (element attribute).
- ▶ Behaviour integrators are called in the assembly loop over the elements for:
  - the residual (contribution of the inner forces)
  - the jacobian matrix (i.e. the tangent stiffness matrix)

- ▶ Behaviour integrators are meant to
  - Compute the gradients from unknowns (using the  $B$  matrix)
  - Handle the state variables
  - Call the behaviour integration
  - Compute the inner forces from the thermodynamic forces
  - Compute the stiffness matrix from the consistent tangent operator
- ▶ All those steps depends on
  - Kind of problem described (mechanics, heat transfer)
  - Symmetry of the material (isotropic or orthotropic)
  - Modelling hypotheses: 3D, plane strain, plane stress, axisymmetry ...
- ▶ The writing of behaviour integrators is tedious and error-prone and shall be done with care
  - However, specific behaviour integrators gives access to pieces of physics
  - Code generation to the rescue !

## ► Aim of this code-generator

- Generate behaviour integrators from definition of the gradients
- Automated code factorisation/optimisation (no sparse matrix multiply)
- Avoid coding errors due to tedious formula

## ► Machinery

- Based on the GiNaC for symbolic computations in C++
- Current scope: isotropic and orthotropic, small and finite strain behaviours in plane strain, plane stress and tridimensional hypotheses

## ► Extensions (to come)

- Support of axisymmetry
- Non linear heat transfer, non linear diffusion
- Can be extended to other non-linear models, wide range of phenomena

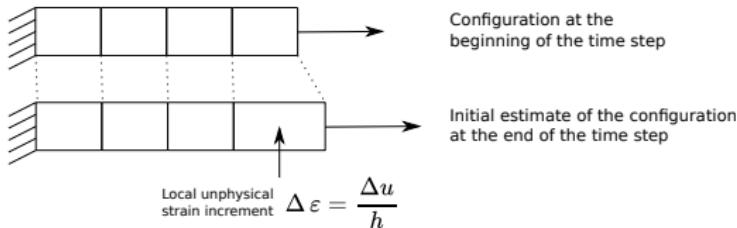
## **Feed-backs on some issues using MFEM**

► Non-linear solve, integration step

- The integration of the mechanical behaviour is a **complex** kernel solving a system of ordinary differential equations which can **fail** -> not handled in MFEM
- Work-around: derived our own Newton algorithm which separates the behaviour integration step from residual/stiffness assemblies (however, does **not work** with PETSc)
- Discussion started:
  - <https://github.com/mfem/mfem/issues/2139>
  - Also discussed during MFEM user meeting (20/10/2021)

► Dirichlet boundary conditions

- Imposing the boundary Dirichlet in MFEM leads to **unphysically strain** increments on elements near the boundary (exacerbated for refined mesh) -> divergence.
- In most **mechanical** solvers, a **prediction** of the solution based on the tangent problem is performed, not easy to do in MFEM
- Discussion started:
  - <https://github.com/mfem/mfem/issues/2174>
  - Also discussed during MFEM user meeting (20/10/2021)



## European project - OperaHPC

Commissariat à l'énergie atomique et aux énergies alternatives - [www.cea.fr](http://www.cea.fr)

## European project

- ▶ Duration nov 2022 -> 2027
- ▶ Focus: Advanced simulation tools enabling **3D representation** of fuel rod
- ▶ Consortium: France, Italy, UK, Switzerland, Czech Republic, Sweden ...

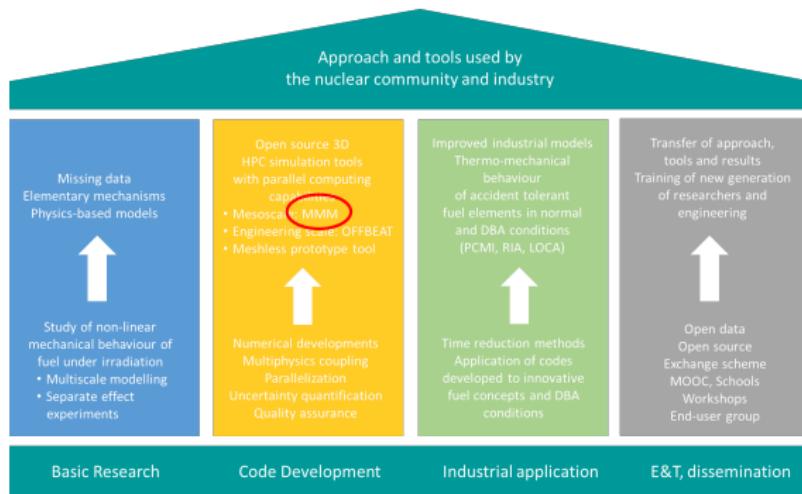
## Aims

- ▶ **Open-source** tools enabling direct or indirect use in **industrial** context
- ▶ Research & training activities, better **knowledge** of fuel rods & fuel behavior
- ▶ Support for **safe operation** of existing nuclear power plants
- ▶ **Physically-based** modelling implemented in HPC 3D simulation tools



## Open-source benefits

- ▶ Open science approach: education/training, ease publication & **dissemination** avoiding *black-box* approaches
- ▶ Use **top-notch** opensource software **stacks**
- ▶ Ease the **coupling** of multiple codes within the consortium
- ▶ Helps promoting advanced HPC tools within nuclear community
- ▶ **Societal impact:** enable european citizen to know about tools used in nuclear field



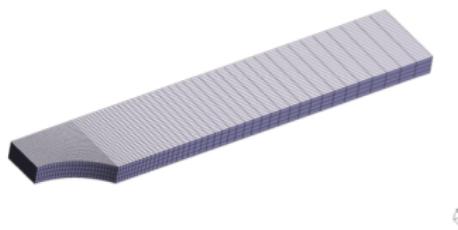
# Examples

# Non linear case

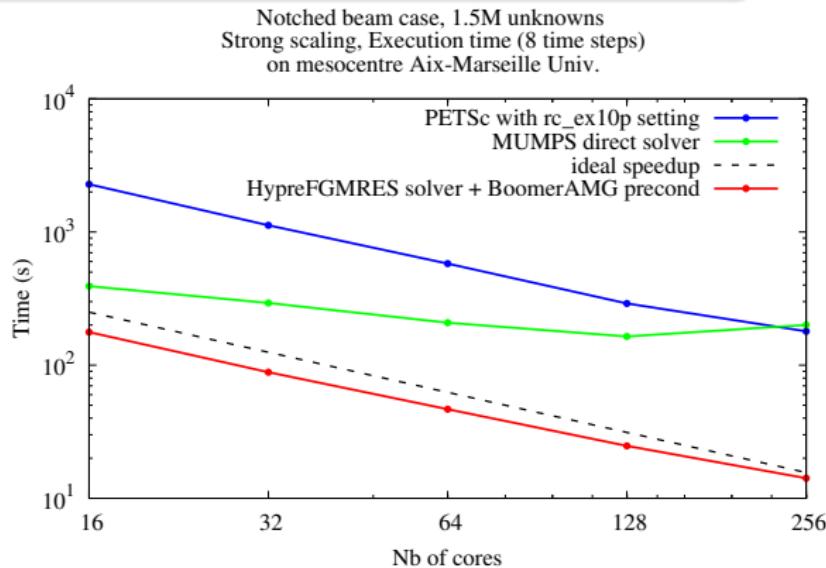
## Elastoplastic modelling, finite strain

### Case description

- ▶ Uniaxial tensile test on a 3D notched beam
- ▶ Imposed displacement: right of the beam
- ▶ Symmetry conditions at: left position, down position

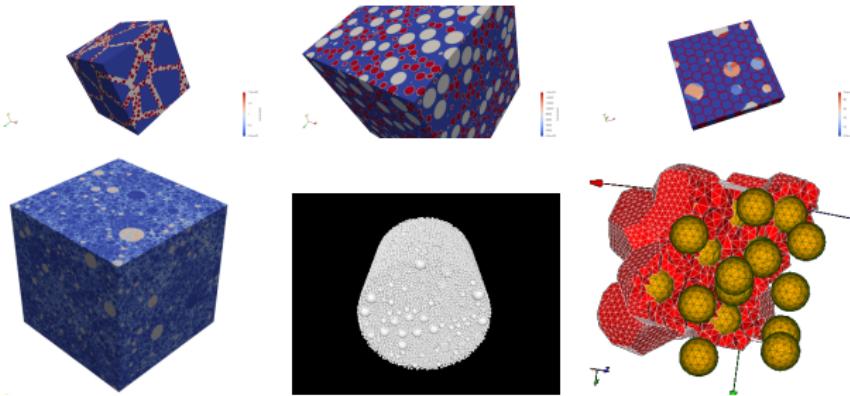


Geometry of the notched beam



### REV modelling in PLEIADES

- ▶ Existing capabilities - not in MMM but using previous Cast3m solver
  - Microstructural representation:  
inclusions, polycrystals, coated materials
  - Application:  
MOX fuel, Accident-Tolerent-Fuel (ATF), Thermal conductivity at fine scale
  - Features: Damage, creep, homogenisation, cracking, multi-material, porosity ...
  - Typical scale of computational domain:  $500 \mu\text{m}$
  - Requirements: Periodic 3D boundary conditions
- ▶ Aim: port these capabilities into MMM



## Properties

- ▶ 14 million unknowns, elastic modelling
- ▶ Ref. case on 32 cores - skylake @ mesocentre Aix-Marseille University
- ▶ Main linear solver: Conj. Gradient (iterative), no preconditioner used

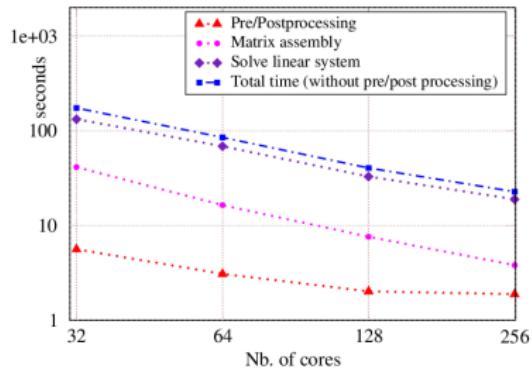
## Timings

- ▶ MFEM linear solve
  - total **166s**, matrix assembly 39s, CGsolver 120s
- ▶ MFEM non-linear solve - residual form, a single newton iteration
  - total **179s**, matrix assembly 53s, CGsolver 117s
- ▶ MMM non-linear solve - residual form, a single newton iteration
  - total **188s**, matrix assembly 42s, CGsolver 133s
- ▶ Overheads due to MMM are low
  - Due to: read/write to memory buffers, function calls for assembly

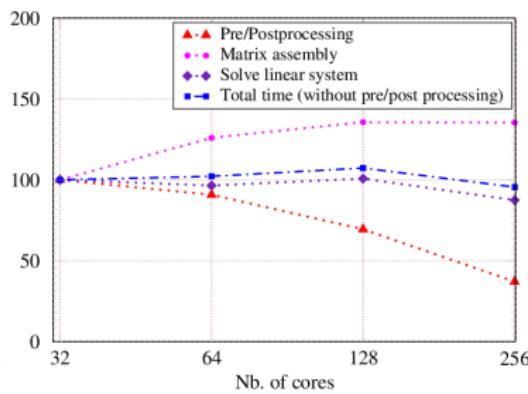
### Strong scaling

- ▶ Using MMM non-linear solve
- ▶ Strong scaling from 32 to 256 cores
- ▶ Scalable parallelism observed (mainly due to MFEM capabilities)

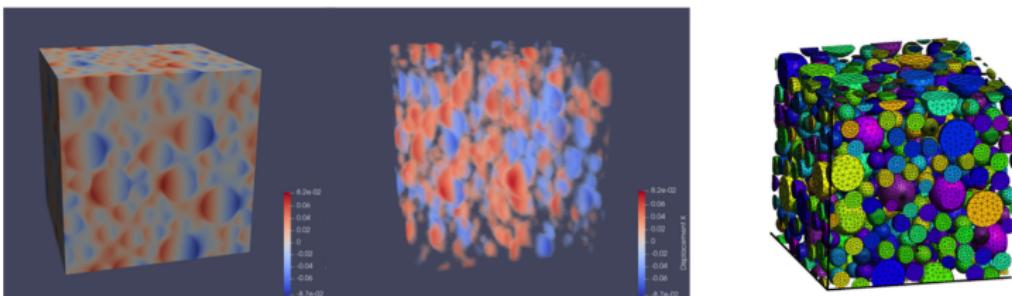
Execution time, Skylake - Mesocentre AMU



Relative efficiency, Skylake - Mesocentre AMU



## Modelling inclusions at fuel pellet's microstructure scale



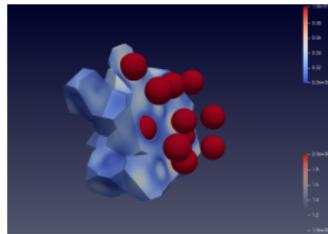
2000 inclusions (cut spheres), elastic modelling,  
displacement in x direction is shown,  
linear system with 500M unknowns

## MFEM contributions

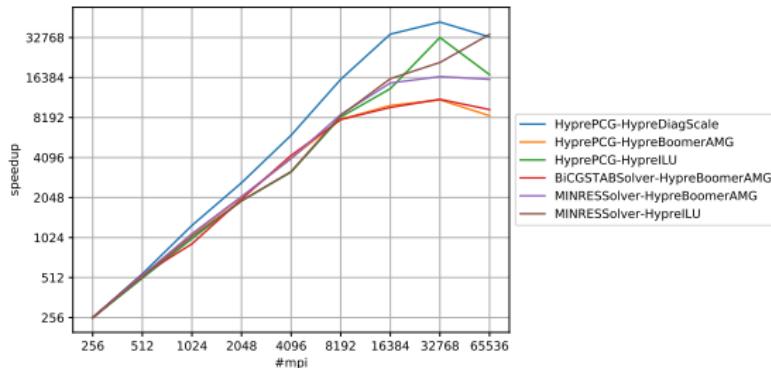
- ▶ Using as input : 3D periodic gmsh meshes  
we made a public MFEM contribution to add these capabilities
- ▶ Read/write MED (Salome platform) I/O file format  
development is not publicly disclosed  
pull request is delayed, need time to go through

## MFEM iterative solvers and preconditioners

- ▶ Investigate several solvers in MFEM  
checking which solvers is most adequate for each test case  
is proposing to branch to many MFEM solvers already
- ▶ While direct solvers represent a fallback solution  
iterative solvers deals with largest systems  
finding good preconditioners is a key
- ▶ Strong scaling benchmark (CEA/CCRT AMD Milan architecture)  
 $80 \cdot 10^6$  unknowns

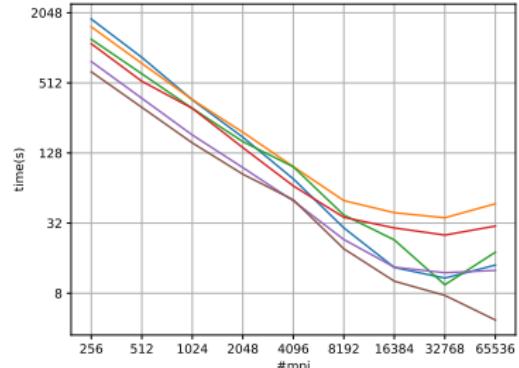


3D periodic setting



Very good scalability up to **16k** cores

Scalability is superlinear for HyperPCG-HypreDiagScale

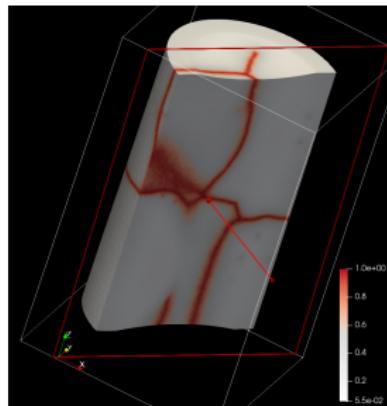
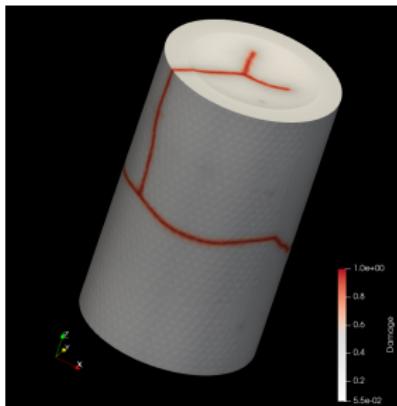


MINRES+HyperILU minimizes time to solution

## Modelling fuel pellet fragmentation during reactor start-up

- ▶ Implementing phase-field approach
  - Some phase-field models have been previously tested: AT1, AT2, Lorentz
  - Micromorphic damage approach used here (David Siedel PhD work - CEA)
  - Exact treatment of the irreversibility constraint at integration points
- ▶  $22 \cdot 10^6$  unknowns, **2500** cores (CEA/CCRT computing facility)

## Numerical results

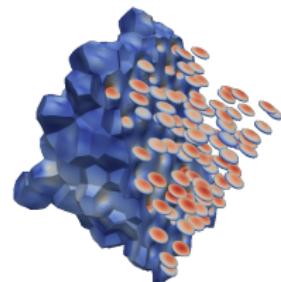


## Conclusions and perspectives

# Conclusions and perspectives

## ► What was done ?

- MMM - opensource project thermo-mechanical solver
- High level declarative API suitable for engineering studies and upcoming integration in PLEIADES ecosystem
  - MFEM API still accessible user a lower level API
- Multi-material support
- Ability to handle arbitrary complex finite strain behaviours:
  - All behaviours are set at runtime (dynamically loaded libraries)



## ► What comes next ?

- Maintain the confidence of industrial partners (EDF, Framatome) on PLEIADES keep the funding
- Overall robustness of the code, tutorials
- Setup extensive continuous integration (github actions + private)
- Extension to other physical phenomena (non linear heat-transfer, diffusion, non local mechanics). *Should be easy* using code generation of behaviour integrators
- Adaptative mesh refinements (requires new data structures in MGIS) - AMR
- Additional boundary conditions, including contact with friction
- Provide more examples
- Checkpoint/restart
- Port to GPUs (requires tremendous work on the MFront and MGIS side)

**Thank you for your attention.  
Time for discussion !**



<https://github.com/thelfer/mfem-mgis>  
<https://github.com/latug0/mfem-mgis-examples>  
[https://github.com/thelfer/  
MFrontGenericInterfaceSupport](https://github.com/thelfer/MFrontGenericInterfaceSupport)  
<https://tfel.sourceforge.net>  
<https://www.researchgate.net/project/TFEL-MFront>