# Passmo : code de calcul séismes sur Arcane

DEN/DM2S/SEMT **M-A BADRI, E. FOERSTER**, G. RASTIELLO, G. FOLZAN

DAM/DIF/DSSI C. CHEVALIER, G. GROSPELLIER, B. LELANDAIS, M.-P. OUDOT

*Rencontres Arcane, TGCC, 5 mars 2020*

## Improving Safety Assessment for nuclear sites:

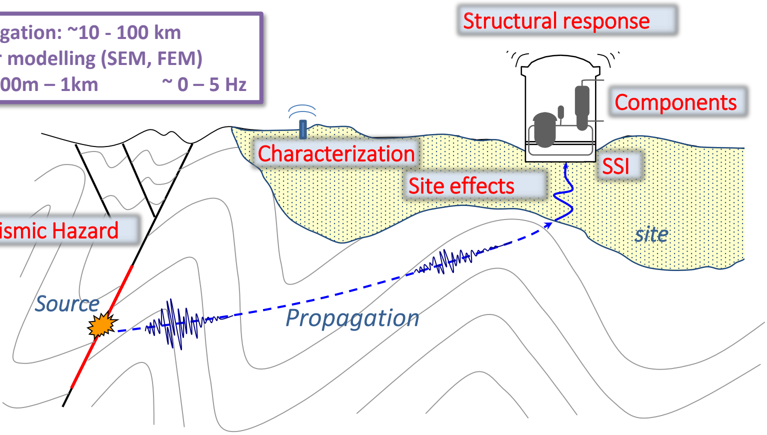
- **Some on-going collaborative research projects**:

New Approach to Reactor Safety ImprovementS
(2017-2021)

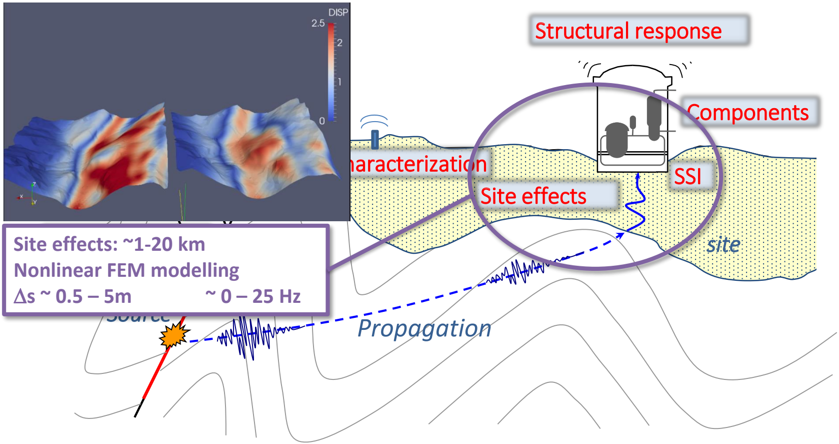
Séisme & Installations Nucléaires: Assurer et Pérenniser la Sûreté
(2013-2020)

## An integrated seismic modeling approach from fault to site:

Propagation: ~10 - 100 km
Linear modelling (SEM, FEM)
$\Delta s$ ~ 100m – 1km ~ 0 – 5 Hz

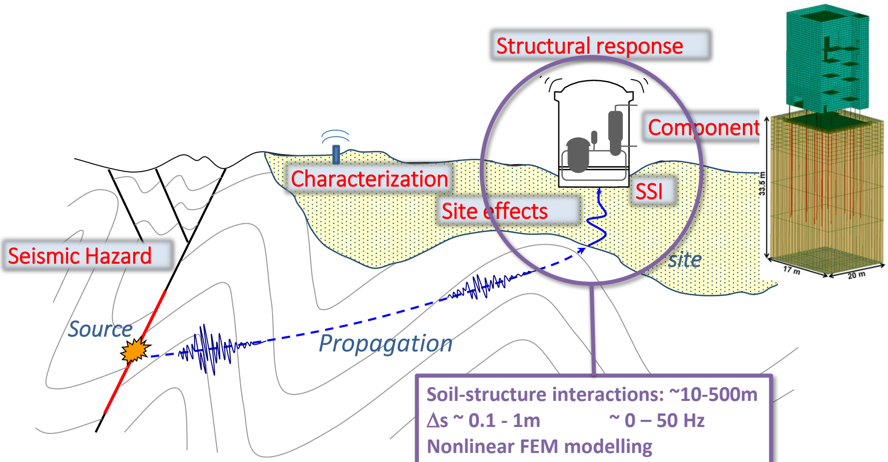


## An integrated seismic modelling approach from fault to site:




**Site effects: ~1-20 km**
**Nonlinear FEM modelling**
**Δs ~ 0.5 – 5m ~ 0 – 25 Hz**

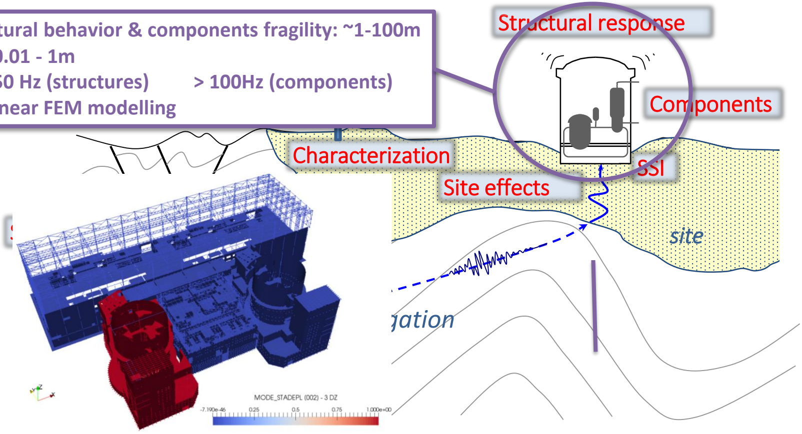
## An integrated seismic modelling approach from fault to site:




**Soil-structure interactions: ~10-500m**
**$\Delta$s ~ 0.1 - 1m ~ 0 – 50 Hz**
**Nonlinear FEM modelling**

## An integrated seismic modelling approach from fault to site:

**Structural behavior & components fragility: ~1-100m**
**Δs ~ 0.01 - 1m**
**~0 – 50 Hz (structures) > 100Hz (components)**
**Nonlinear FEM modelling**



## Our main goals:

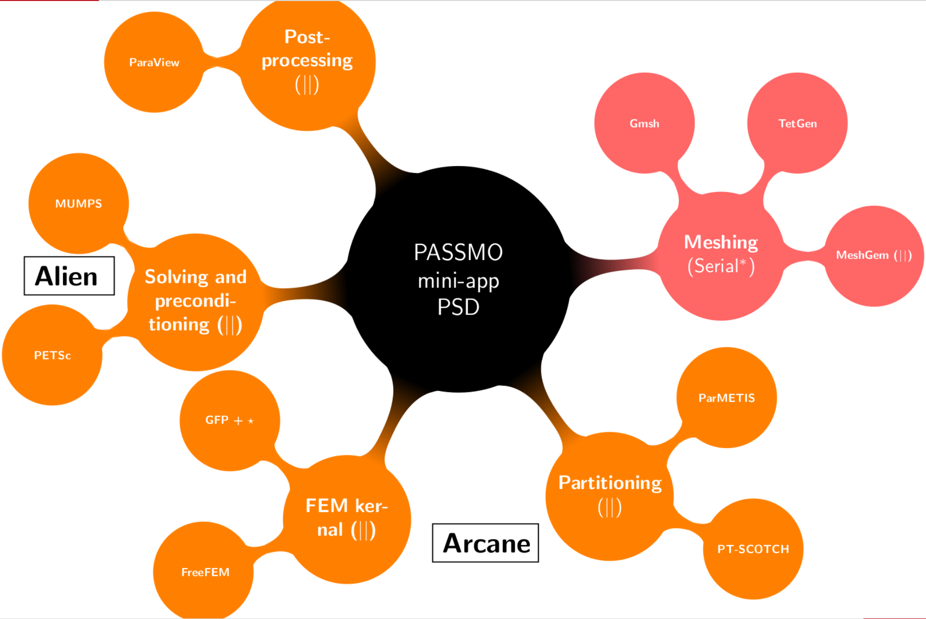
- **To achieve full FEM "best-estimate" and/or "high-fidelity" 3D modelling**
  e.g. for seismic PSA of nuclear sites including interactions (soils, structures, components) and detailed material behaviors (local damage, …), variabilities and uncertainties

- To have a **full parallel perspective** for **computing** but also for pre- & post-processing (meshing, visualization, …)

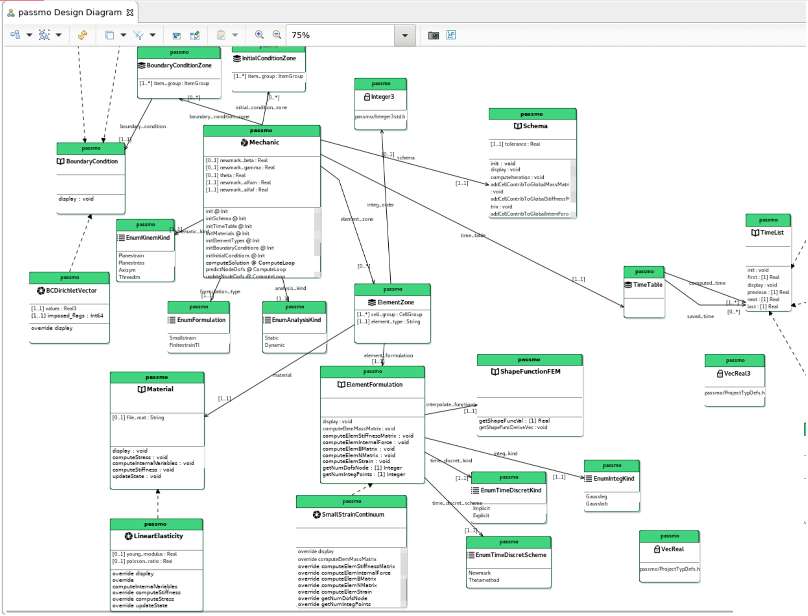
- To work either on **Exascale** parallel or **multi-core** computing **architectures** (even on the "every-day" laptops and PC's)

## On-going developments:

- **Linear implicit iterative solvers** based on domain decomposition, for **damage mechanics and dynamics**
- **Tailored Algebraic Multi-Grid preconditioner** to improve the solver performances and reach quasi-linear scaling
- **Vectorial FEM approach**
- **Fully parallel process:** unstructured meshing, partitioning, assembling, solving & post-processing
- **Arcane FEM kernel for transient nonlinear dynamics (soils & structures)**


Post-processing (||)
ParaView
MUMPS
Alien
Solving and precondi-tioning (||)
PETSc
GFP + ⋆
FEM ker-nal (||)
FreeFEM
PASSMO mini-app PSD
Gmsh
TetGen
Meshing (Serial*)
MeshGem (||)
ParMETIS
Partitioning (||)
PT-SCOTCH
Arcane

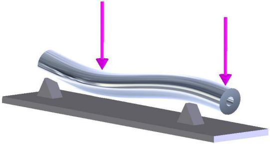
# PASSMO App (Performant Assessment for Seismic Site Modeling)

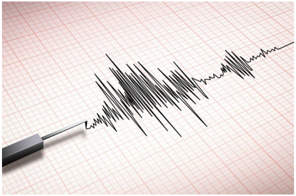


# Outline

- Context
- Introduction
- Parallelization
- VFEM for solid mechanics
- HPC analysis
- Examples
- Conclusions

**“ Main goal is to develop parallel finite element solvers for standalone solid mechanics problems ”**





- **Linear Elasticity**
  - 2D & 3D
  - Homogeneous
  - Heterogeneous
- **Damage Mechanics**
  - 2D & 3D
  - Brittle fracture
  - Quasi-brittle fracture
- **Earthquake**
  - 2D & 3D
  - Linear dynamics
  - Nonlinear dynamics

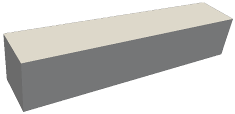
- Discretization (FEM) leads to solving

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

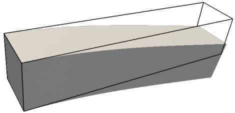
- Matrix **A** & vector **b** inherits physics
- **x** is solution (displacement) field

- Discretization (FEM) leads to solving
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
- Matrix **A** & vector **b** inherits physics
- **x** is solution (displacement) field

**Elasticity**

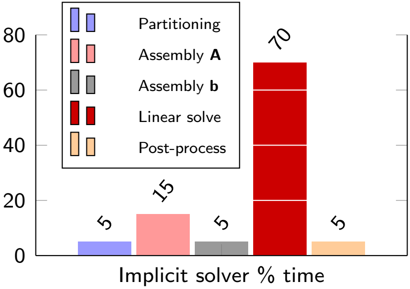


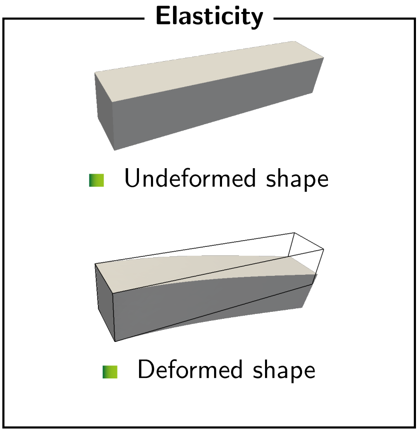
- Undeformed shape



- Deformed shape

- Discretization (FEM) leads to solving
$$\mathbf{Ax} = \mathbf{b}$$
- Matrix **A** & vector **b** inherits physics
- **x** is solution (displacement) field




**Assembly is embarrassingly parallel. Major focus reduce the linear system solving time. Critical in dynamic/non-linear as repeated solving**

# Outline

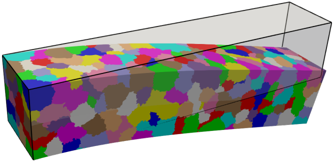
# Parallelization

## What

- Means to **faster computation**
- Means to **larger computation**

## Why

- Large scale problems
- Huge meshes
  - Complex geometry
  - Sharp solution fields
- Time/Load/Non-linear loop



| D.O.F | Overall solving time | |
|---|---|---|
| | 768 PU | 1 PU |
| 112 Million | 220 (s) | $>$ 2 (days) |

## What

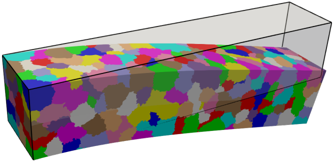
- Means to **faster computation**
- Means to **larger computation**

## Why

- Large scale problems
- Huge meshes
  - Complex geometry
  - Sharp solution fields
- Time/Load/Non-linear loop

## How

- Use multiple processing units (CPUs / cores / processes / threads,..)
  - **Assemble linear system in parallel**
  - **Solve linear system in parallel**



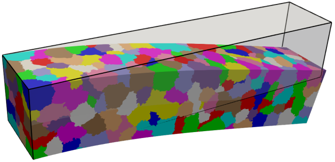
| D.O.F | Overall solving time | |
|---|---|---|
| | 768 PU | 1 PU |
| 112 Million | 220 (s) | $>2$ (days) |

## What

- Means to **faster computation**
- Means to **larger computation**

## Why

- Large scale problems
- Huge meshes
  - Complex geometry
  - Sharp solution fields
- Time/Load/Non-linear loop

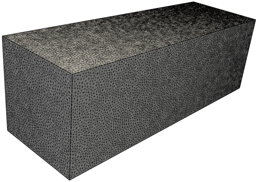


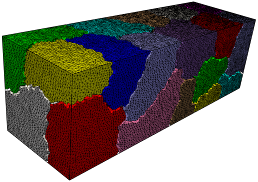
| D.O.F | Overall solving time | |
|---|---|---|
| | 768 PU | 1 PU |
| 112 Million | 220 (s) | $>2$ (days) |

## How

- Use multiple processing units (CPUs / cores / processes / threads,..)
  - **Assemble linear system in parallel**
  - **Solve linear system in parallel**

**How to break a continuous simulation into discontinuous one?**

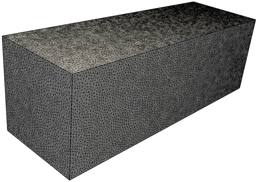




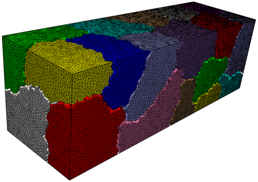
- Domain $\Omega$
- Partitioned $\{\Omega^h\}_{i=1}^{18}$

- Divide $\Omega^h$ into $\{\Omega^h\}_{i=1}^{N_p}$ (**ParMETIS**)
- Build $\{\mathbf{Ax} = \mathbf{b}\}_{i=1}^{N_p}$ (**FreeFEM + GFP**)
- Build (**MPI**) communications
  - Restriction matrix $\{R\}_{i=1}^{N_p}$
  - Partition of unity $\{D\}_{i=1}^{N_p}$

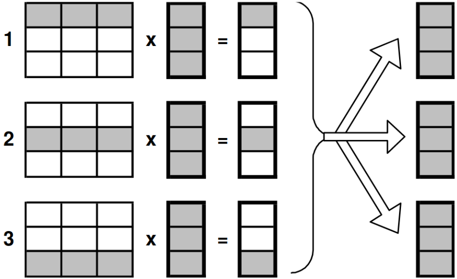
# Domain Decomposition



- Domain $\Omega$

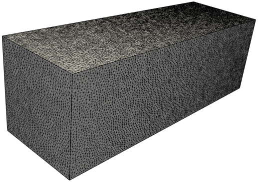


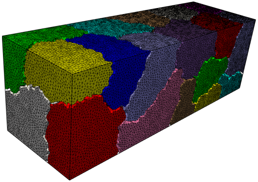
- Partitioned $\{\Omega^h\}_{i=1}^{18}$



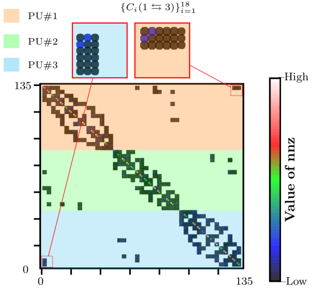

- Divide $\Omega^h$ into $\{\Omega^h\}_{i=1}^{N_p}$ (**ParMETIS**)
- Build $\{\mathbf{Ax} = \mathbf{b}\}_{i=1}^{N_p}$ (**FreeFEM + GFP**)
- Build (**MPI**) communications
  - Restriction matrix $\{R\}_{i=1}^{N_p}$
  - Partition of unity $\{D\}_{i=1}^{N_p}$

# Domain Decomposition





- Domain $\Omega$
- Partitioned $\{\Omega^h\}_{i=1}^{18}$




- Divide $\Omega^h$ into $\{\Omega^h\}_{i=1}^{N_p}$ (**ParMETIS**)
- Build $\{\mathbf{Ax} = \mathbf{b}\}_{i=1}^{N_p}$ (**FreeFEM + GFP**)
- Build (**MPI**) communications
  - Restriction matrix $\{R\}_{i=1}^{N_p}$
  - Partition of unity $\{D\}_{i=1}^{N_p}$
- Solve parallel using Krylov subspace (**PETSc**)
- **Parallel matrix–vector product**

## What

- Means to faster solution
  $$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$
- Means to decrease iteration

## Why

- Ill conditioned problems
- Strongly coupled
- Efficient parallel algorithm

1. M. Seaid et al., *J. of Computational and Applied Mathematics*, v. 170 (2004).

## What

- Means to faster solution

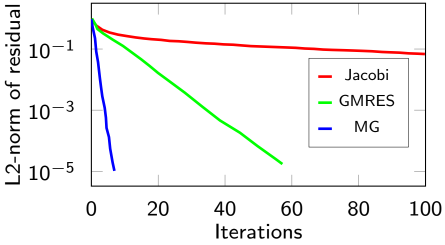
$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

- Means to decrease iteration

## Why

- Ill conditioned problems
- Strongly coupled
- Efficient parallel algorithm

**Example** [1]




*Jacobi extremely slow, GMRES and MG better"*

1. M. Seaid et al., *J. of Computational and Applied Mathematics*, v. 170 (2004).

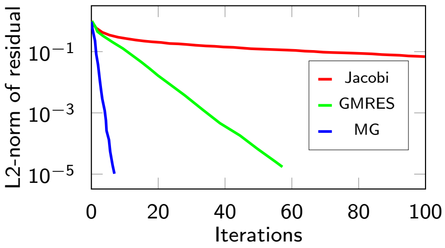
## What

- Means to faster solution
  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$
- Means to decrease iteration

## Why

- Ill conditioned problems
- Strongly coupled
- Efficient parallel algorithm

**Example** [1]




*Jacobi extremely slow, GMRES and MG better"*

## How

- Use the Krylov subspace method on modified system such as
  - Left preconditioned system $M^{-1}\mathbf{A}\mathbf{x} = M^{-1}\mathbf{b}$
  - Right preconditioned system $\mathbf{A}M^{-1}y = \mathbf{b}$ with $\mathbf{x} = M^{-1}y$
- **One level: Jacobi & Block Jacobi, Multi-level?**

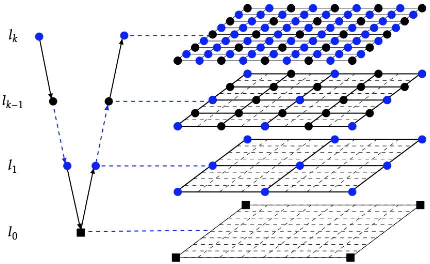
1. M. Seaid et al., *J. of Computational and Applied Mathematics*, v. 170 (2004).

**What**

- Use hierarchy discretizations
- Restrict and interpolate cycle

**Cons**

- Additional meshes
- Non-trivial for unstructured.




- Geometric MG

# Multigrid preconditioning

## What

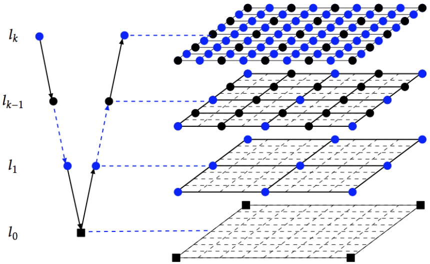
- Use hierarchy discretizations
- Restrict and interpolate cycle

## Cons

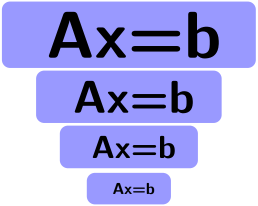
- Additional meshes
- Non-trivial for unstructured.

## Alternative: Algebraic Multigrid

- Construct coarse spaces from **A**
- Coarsen until LU or SVD applicable
- Difficult to implement and tune
  - Threshold parameter (coarsening rate)




Geometric MG




Algebraic MG

find **u** and $\phi : \Omega \to \mathbb{R}^3$

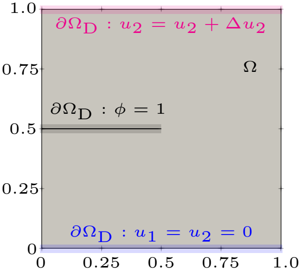
$$\mathcal{E}_{ME}(\mathbf{u}, \phi) = \left((1-\phi)^2 + k\right) \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = 0,$$

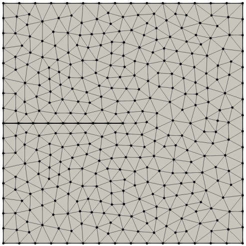
$$\mathcal{E}_{PFE}(\mathbf{u}, \phi) = \left(\frac{G_c}{l_0} + 2\mathcal{H}^+(\mathbf{u})\right) \phi - G_c l_0 \Delta\phi - 2\mathcal{H}^+(\mathbf{u}) = 0.$$

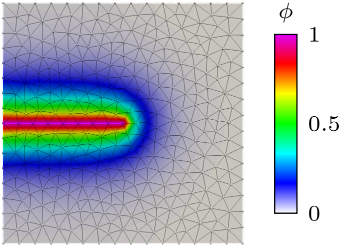
**find u and $\phi : \Omega \to \mathbb{R}^3$**

$$\mathcal{E}_{ME}(\mathbf{u}, \phi) = \left((1-\phi)^2 + k\right) \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = 0,$$

$$\mathcal{E}_{PFE}(\mathbf{u}, \phi) = \left(\frac{G_c}{l_0} + 2\mathcal{H}^+(\mathbf{u})\right) \phi - G_c l_0 \Delta\phi - 2\mathcal{H}^+(\mathbf{u}) = 0.$$





**find $\mathbf{u}$ and $\phi : \Omega \to \mathbb{R}^3$**

$$\mathcal{E}_{ME}(\mathbf{u},\phi) = \left((1-\phi)^2 + k\right) \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = 0,$$
$$\mathcal{E}_{PFE}(\mathbf{u},\phi) = \left(\frac{G_c}{l_0} + 2\mathcal{H}^+(\mathbf{u})\right)\phi - G_c l_0 \Delta\phi - 2\mathcal{H}^+(\mathbf{u}) = 0.$$

**Algorithm - staggered**

- Partition
- Build communication pattern $\times 2$
- Solve equilibrium equation for $\mathbf{u}$ with $\phi$ frozen
- Solve phase-field equation for $\phi$ with $\mathbf{u}$ frozen

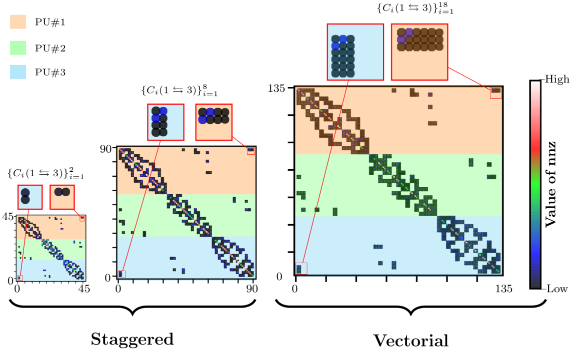
**find $\mathbf{u}$ and $\phi : \Omega \to \mathbb{R}^3$**

$$\mathcal{E}_{ME}(\mathbf{u},\phi) = \left((1-\phi)^2 + k\right) \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = 0,$$
$$\mathcal{E}_{PFE}(\mathbf{u},\phi) = \left(\frac{G_c}{l_0} + 2\mathcal{H}^+(\mathbf{u})\right)\phi - G_c l_0 \Delta\phi - 2\mathcal{H}^+(\mathbf{u}) = 0.$$

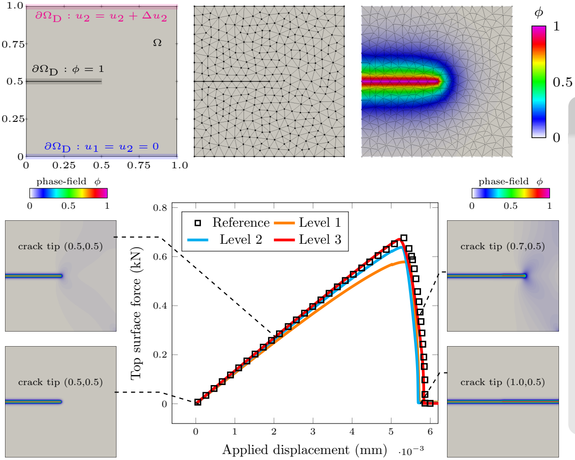
**Algorithm - monolithic/vectorial**

- Partition
- Build communication pattern $\times 1$ **GAIN**
- Solve equilibrium and phase-field equations for $< \mathbf{u}, \phi >$ **GAIN**

Advantages: **Faster** ($<$ iterations, $<$ overall-time, etc.) and apply **CG**.




- Faster MPI communication
- Less Krylov iterations to converge
- Coarser stepping



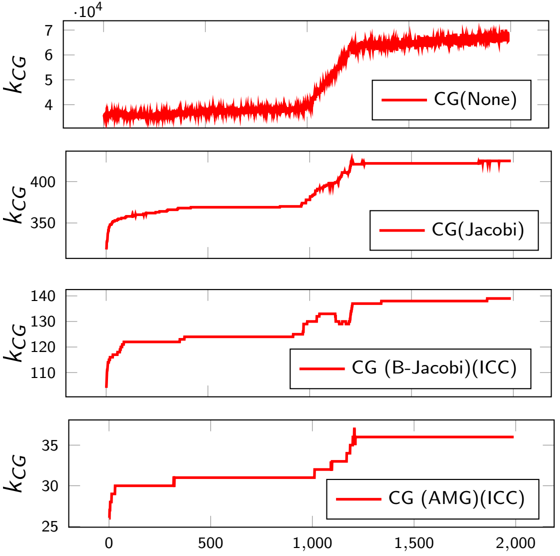

## Features

- Non-linear
- Quasi-static
- Tightly coupled as crack moves
- Fine meshing as brittle material
- Taken from Ambati et al.[1]

1. M. Ambati et al., *J. of Computational Mechanics*, v. 55 n. 2 (2014).

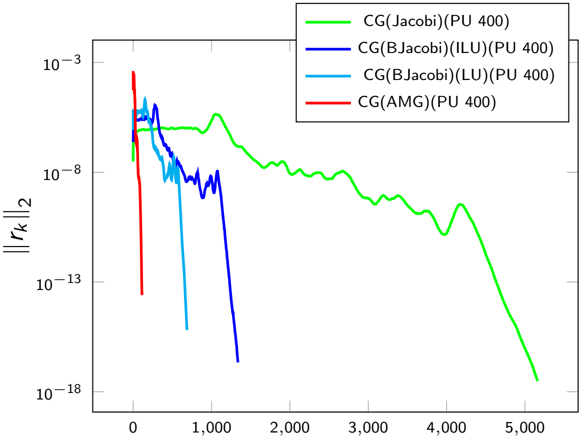
- Context
- Introduction
- Parallelization
- VFEM for solid mechanics
- HPC analysis
- Examples
- Conclusions

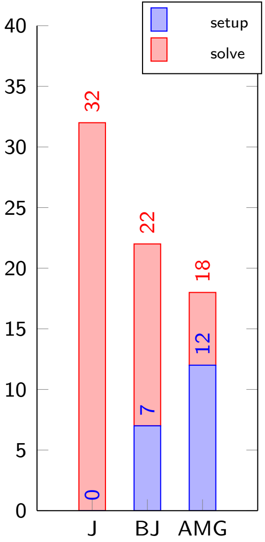
# PFM / Metric 1: Iterations to converge



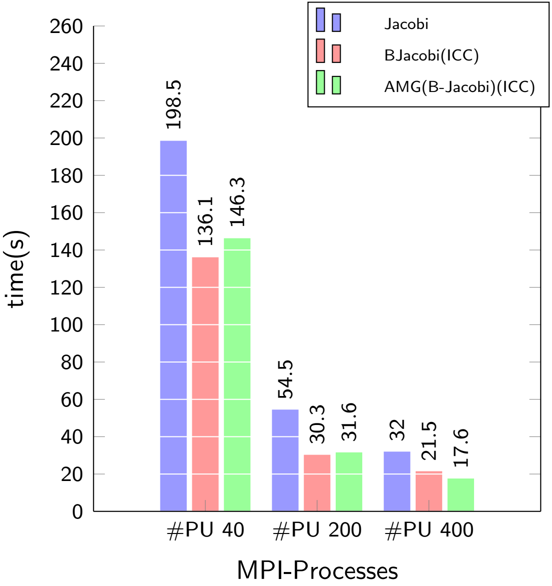

## Details

- SEMT2 cluster @CEA
  - Ivy Bridge 10 Node
  - Intel Xeon E5-2680@2.6GHz
  - 40 PU per Node
- 0.4M dof. on 80 PU
- Vectorial FEM
- 2D cracking
- Preconditioners:
  - None $\mathbf{A}\mathbf{x} = \mathbf{b}$
  - Jacobi
  - Block-Jacobi (ICC)
  - AMG(Block-Jacobi)(ICC)
  - Block-Jacobi & AMG with inner (CLU).



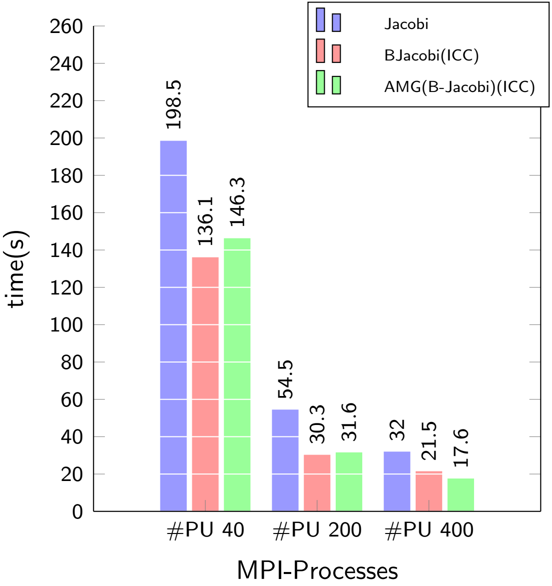




- Textbook monolithic CG residue drop for AMG.
- Even at peak AMG is dominated by setup cost.




## Details

- SEMT2 cluster @CEA
  - Ivy Bridge 10 Node
  - Intel Xeon E5-2680@2.6GHz
  - 40 PU per Node
- 10M dof. 3D cracking
- Preconditioners:
  - Jacobi
  - B-Jacobi (ICC)
  - AMG(B-Jacobi)(ICC)

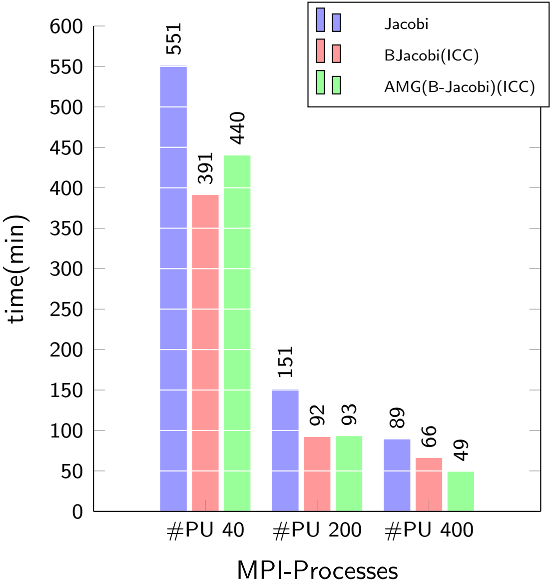



## Details

- SEMT2 cluster @CEA
  - Ivy Bridge 10 Node
  - Intel Xeon E5-2680@2.6GHz
  - 40 PU per Node
- 10M dof. 3D cracking
- Preconditioners:
  - Jacobi
  - B-Jacobi (ICC)
  - AMG(B-Jacobi)(ICC)

## Performance

- Peak
  - Jacobi =62%
  - B-Jacobi(ICC) =65%
  - AMG(B-Jacobi)(ICC) =83%

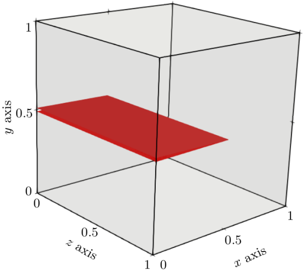



## Details

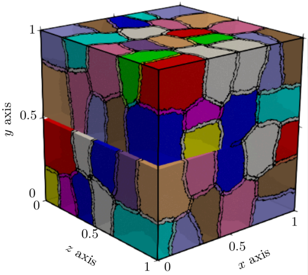
- SEMT2 cluster @CEA
  - Ivy Bridge 10 Node
  - Intel Xeon E5-2680@2.6GHz
  - 40 PU per Node
- 10M dof. 3D cracking
- Preconditioners:
  - Jacobi
  - B-Jacobi (ICC)
  - AMG(B-Jacobi)(ICC)

## Performance

- Peak
  - Jacobi =61%
  - B-Jacobi(ICC) =60%
  - AMG(B-Jacobi)(ICC) =89%



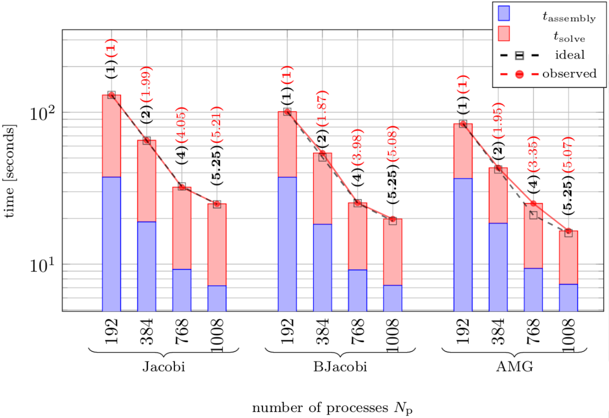

(a) domain $\Omega$ of test 2A and 2B.




(b) partitioned mesh $\{\Omega_i^h\}_{i=1}^{96}$ of test 2A.

## Details

- INTI cluster @CEA
  - Skylake 21 Node
  - Intel Xeon E5-8890 v4-@3.4GHz
  - 48 PU per Node
- 81M dof. 3D problem
- Preconditioners:
  - Jacobi
  - B-Jacobi (ICC)
  - AMG(B-Jacobi)(ICC)

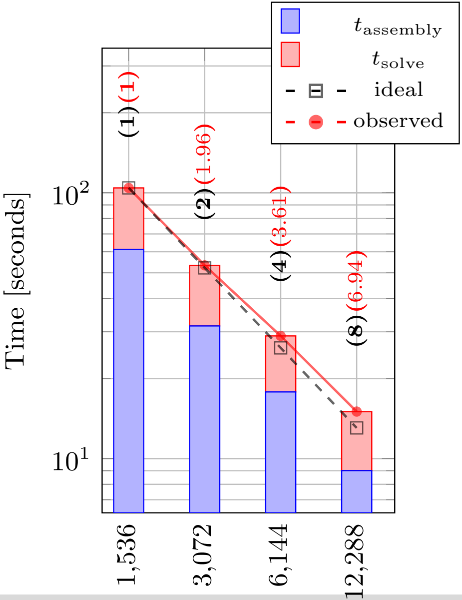



## Details

- INTI cluster @CEA
  - Skylake 21 Node
  - Intel Xeon E5-8890 v4-@3.4GHz
  - 48 PU per Node
- 81M dof. 3D problem
- Preconditioners:
  - Jacobi
  - B-Jacobi (ICC)
  - AMG(B-Jacobi)(ICC)

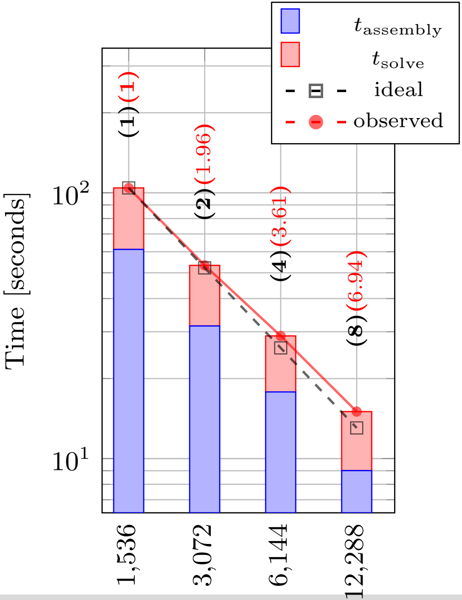
## Performance

- Quasi linear scaling




## Details

- IRENE supercomputer @TGCC-CEA
  - Skylake nodes
- 1.3B dof. sesimic simulation
  - * 64 bit integer support PETSc
- Preconditioners:
  - B-Jacobi (ICC)




## Details

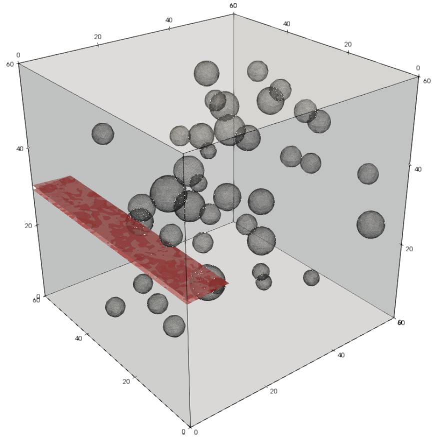
- IRENE supercomputer @TGCC-CEA
  - Skylake nodes
- 1.3B dof. sesimic simulation
- * 64 bit integer support PETSc
- Preconditioners:
  - B-Jacobi (ICC)

## Performance

- Peak
  - B-Jacobi(ICC) =85%

# Outline

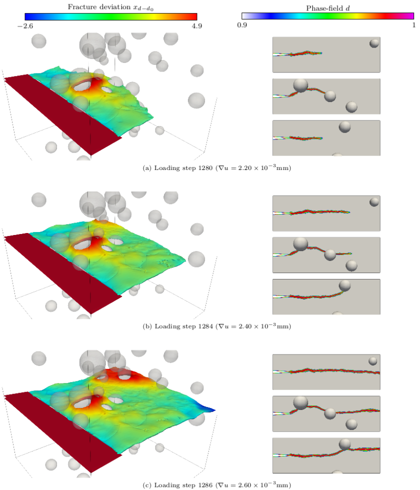
# Application: 3D non-linear fracture mechanics




- 3D fracture mechanics in randomly perforated medium
- Crack propagation needs extremely fine meshing
- 64 Million d.o.f needed to solved for a single time step
- 2,650 total global iterations (time steps + NL)
- 2,400 MPI processes 210 min. of total run-time on Irene Joliot-Curie supercomputer [1]
- Serial computation would take almost one year

1. Irene Joliot-Curie supercomputer, *Hosted at CEA/TGCC Bruyères-Le-Châtel, France.*

# Application: 3D non-linear fracture mechanics



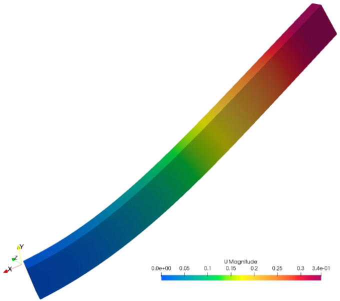

(a) Loading step 1280 ($\nabla u = 2.20 \times 10^{-3}$mm)

(b) Loading step 1284 ($\nabla u = 2.40 \times 10^{-3}$mm)

(c) Loading step 1286 ($\nabla u = 2.60 \times 10^{-3}$mm)

- 3D fracture mechanics in randomly perforated medium
- Crack propagation needs extremely fine meshing
- 64 Million d.o.f needed to solved for a single time step
- 2,650 total global iterations (time steps + NL)
- 2,400 MPI processes 210 min. of total run-time on Irene Joliot-Curie supercomputer [1]
- Serial computation would take almost one year

1. Irene Joliot-Curie supercomputer, *Hosted at CEA/TGCC Bruyères-Le-Châtel, France.*



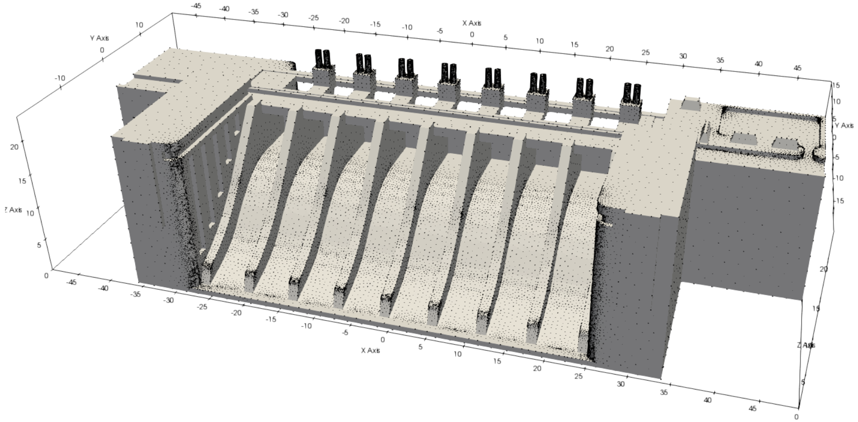

- Simple elasto-dynamics 3D bar problem
- 30K d.o.f 90 times (generalized-$\alpha$ method[1])
- 4 MPI processes 11 minutes of total run-time on laptop (Intel i7)
- Serial computation would mean 45 min.

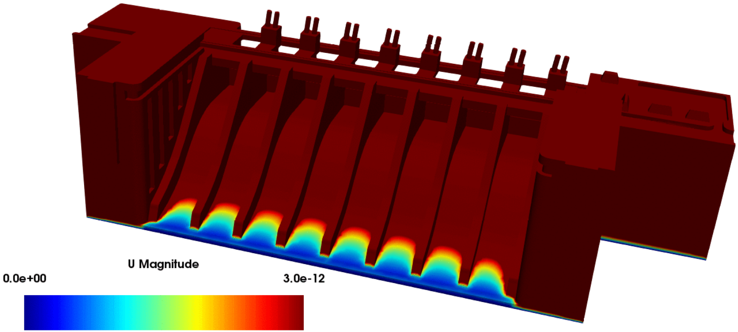
## Linear momentum variational form

$$\int_{\Omega_i^h} \rho \ddot{\mathbf{u}} \cdot \mathbf{v} + \int_{\Omega_i^h} \boldsymbol{\sigma}(\mathbf{u}^h) : \boldsymbol{\varepsilon}(\mathbf{v}^h) = \int_{\Omega_i^h} \rho \mathrm{f} \cdot \mathbf{v}^h + \int_{\partial\Omega_i^h} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{v}^h, \quad \forall \mathbf{v}^h \in [H_0^1(\Omega_i^h)]^3$$

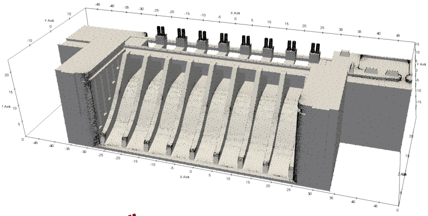
1. S. Erlicher,, *Computational Mechanics*, v. 28, p. 83–104, (2002).

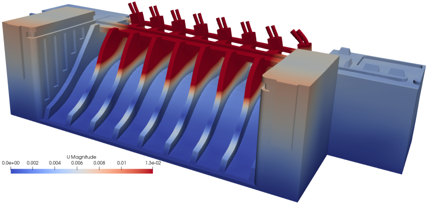
# Application: 3D dynamics on a full scale dam




U Magnitude
0.0e+00
3.0e-12

# Application: 3D dynamics on a full scale dam



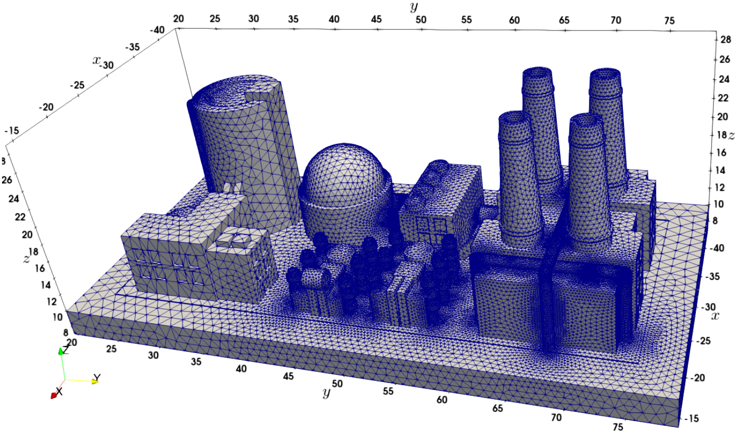



- San Andreas Hoover Dam model used for mesh construction (animation gaming mesh)
- Actual-full dam model 40 Million d.o.f
- 220 time steps (Time discretization using the generalized-$\alpha$ method [1])
- 400 MPI processes 1.5 hours of total run-time on SEMT cluster (10 nodes Ivy-Bridge)
- Serial computation would mean 25 days

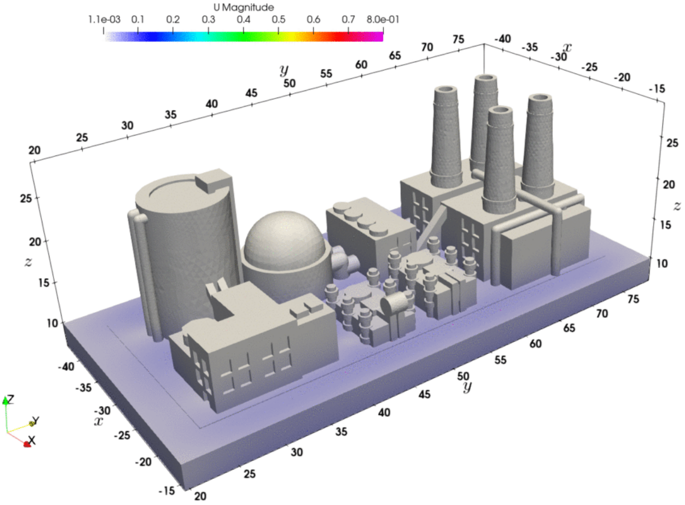
**balanced linear momentum variational form (decomposed)**

$$\int_{\Omega_i^h} \rho \ddot{\mathbf{u}} \cdot \mathbf{v} + \int_{\Omega_i^h} \boldsymbol{\sigma}(\mathbf{u}^h) : \boldsymbol{\varepsilon}(\mathbf{v}^h) = \int_{\Omega_i^h} \rho \mathrm{f} \cdot \mathbf{v}^h + \int_{\partial\Omega_i^h} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{v}^h, \quad \forall \mathbf{v}^h \in [H_0^1(\Omega_i^h)]^3$$

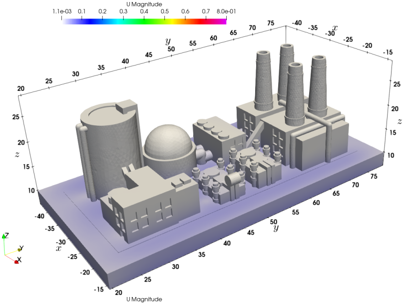
# Application: 3D full dummy nuclear structure

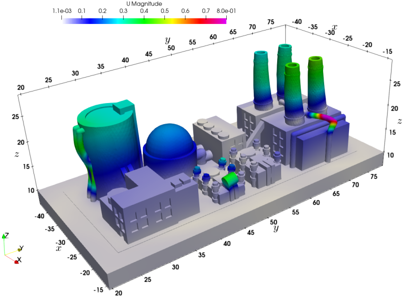


# Application: 3D full dummy nuclear structure



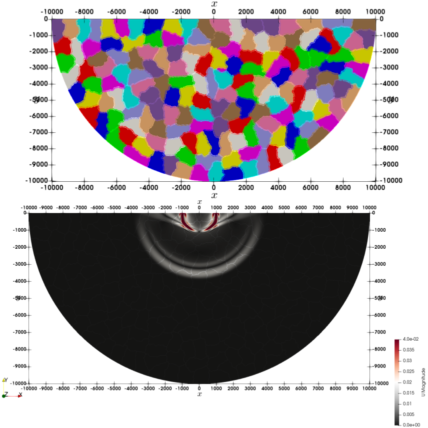
# Application: 3D full dummy nuclear structure



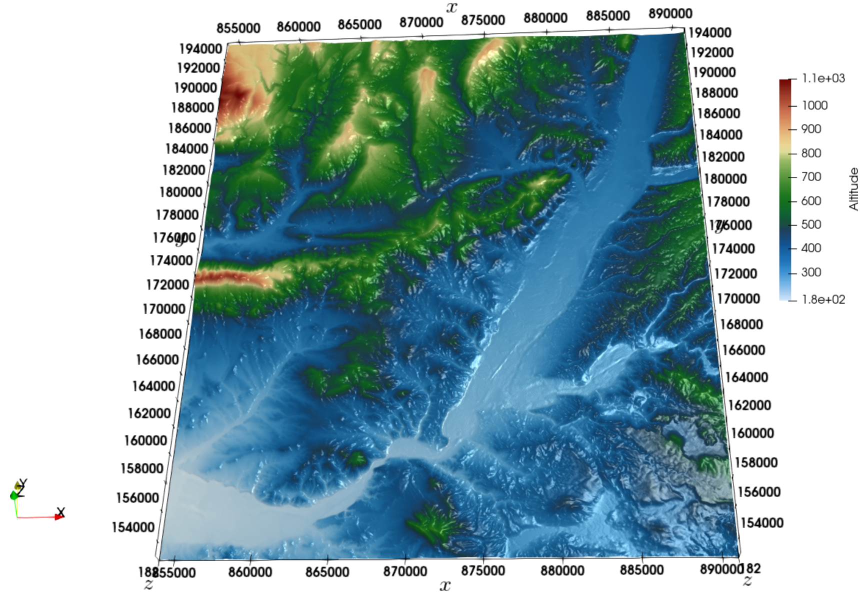

- Full scale dummy nuclear site
- 17 Million d.o.f.
- 250 time steps (Newmark-$\beta$ method)
- 768 MPI processes 1 hours of total run-time on INTI
- Serial computation would take almost one month

# Application: 2D soil seismic wave propagation

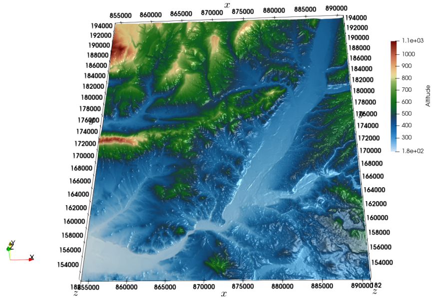


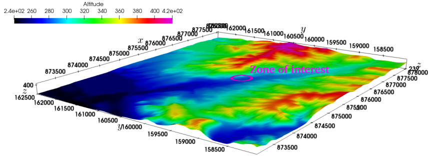

- 20 Km $\times$ 10 Km soil simulation with 2 m mesh size 13.5 Million d.o.f.
- Absorbing boundary conditions
- 2,000 time steps (Newmark-$\beta$ method)
- 240 MPI processes 25 minutes of total run-time on SEMT cluster (10 nodes Ivy-Bridge)
- Serial computation would mean 4 days

# Application: 3D site-scale soil dynamics

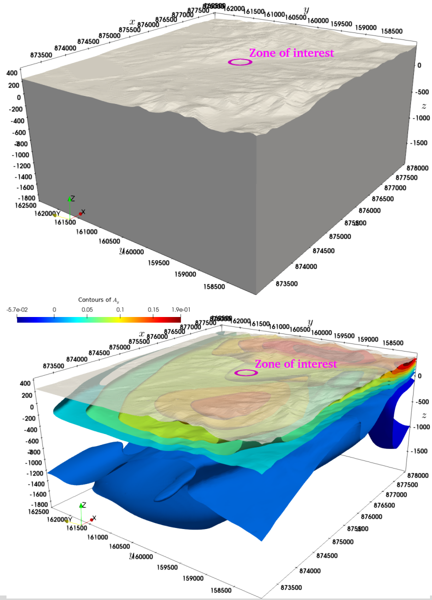


# Application: 3D site-scale soil dynamics




- Full scale site – Cadarache area France – with real topological data
- Parallel mesher - Top2Vol (novel development)
- Parallel post-processing on coarse mesh - ParaView
- 710 Million d.o.f.
- 400 time steps
- 6,144 MPI processes 2.5 hours of total run-time on Irene Joliot-Curie supercomputer [1]
- Serial computation would take almost two years




- Full scale site – Cadarache area France – with real topological data
- Parallel mesher - Top2Vol (novel development)
- Parallel post-processing on coarse mesh - ParaView
- 710 Million d.o.f.
- 400 time steps
- 6,144 MPI processes 2.5 hours of total run-time on Irene Joliot-Curie supercomputer [1]
- Serial computation would take almost two years

# Outline

## Conclusions:

- Presented PASSMO min-app PSD – **distributed memory parallelization** assisted solid mechanics solver.
- **Vectorial finite element** way of handling fracture mechanics.
- **Quasi-optimal scaling** of the solvers (millions of unknowns).

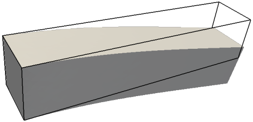
## Conclusions:

- Presented PASSMO min-app PSD – **distributed memory parallelization** assisted solid mechanics solver.
- **Vectorial finite element** way of handling fracture mechanics.
- **Quasi-optimal scaling** of the solvers (millions of unknowns).

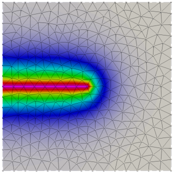
## Perspectives:

- **Integration in Arcane** FEM kernel and bechmarking.
- Testing on **massively parallel** machines.
  - TGCC Joliot Curie IRENE, CINES OCCIGEN.
  - R-CCS supercomputer (Fugaku) through on-going RIKEN-CEA collaboration?
- Preconditioning development – **null-space assisted AMG** (**Alien**).
- Testing n**onlinear constitutive models** – dynamic solving strategies.
- **Real site applications** (e.g. for PIA SINAPS@ & H2020 NARSIS projects).
  - seismic scenarios assessment for full Cadarache for SINAPS@
- "Meshing free finite element" for earthquake, **Top2Vol parallel mesher**.

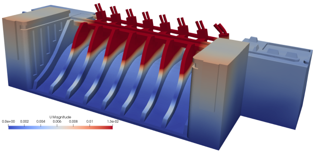
# Thank you for your attention



**Linear Elasticity**

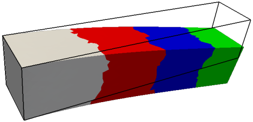


**Fracture Mechanics**

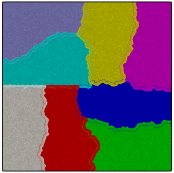


**Elasto-dynamics**

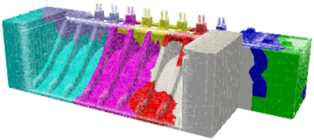
# ? Questions ?



**|| Linear Elasticity**



**|| Fracture Mechanics**



**|| Elasto-dynamics**