Time domain multiscale FWI

with waveform adapted meshes

Keith J. Roberts, WS3/4, STMI Project, Feb. 9, 2021

Aim

- A multiscale full-waveform inversion (FWI) approach in finite elements that can reduce the computational cost over conventional approaches (using finite element methods).
    - By adapting the mesh to the expected source characteristics and characteristics of the wavefield (e.g., P-wave velocity), the number of elements necessary to discretize the domain is reduced.
        - Adaptation occurs during FWI automatically (automatic mesh generation).

Introduction
Several methods

- Simplical, Continuous Galerkin FEM produces sparse system of equations that results in *relatively* slow wave propagation simulations and large matrix storage requirements.
  - Discontinuous Galerkin (e.g., IP-DG) FEM produces block diagonal systems of equations which can be more efficient than CG using explicit time stepping and lower matrix storage requirements.
  - Spectral Element Methods (SEM) work excellently on quad/hexaderal elements but poorly on simplical ones.
    - Unstructured mesh generation is more difficult for hexaderal elements.
  - **Higher-order mass lumped simplical elements by Mulder et al. lead to quick wave solutions with minimal storage requirements. Further, well-developed automatic mesh generation software exist for triangular meshes to take advantage of mesh adaptation.**

Approach

Inversion sketch

Algorithm 1 Optimized velocity $c(\mathbf{x})$ model over a range of source frequencies $freq$

procedure MULTISCALE FULL WAVEFORM INVERSION
  $c^0 \leftarrow$ initial velocity model
  $iter^{max.} \leftarrow$ maximum number of iterations per freq. band
  $k \leftarrow 0$
  for $f \leftarrow freq_{min.}$ to $freq_{max.}$ **do**
    Assign source frequency f;
    while $(\nabla J^k > 0 \& J^k > 0) || k < iter^{max.}$ **do**
      $J^k \leftarrow 0$
      $\nabla J^k \leftarrow 0$
      for shot $\in$ shots **do**
        Compute forward simulation for shot;
        Compute functional and add it to J^k.
        Compute gradient via discrete adjoint and add to ∇J^k;
      Given ∇J^k and J^k using L-BFGS produce c_f^{k+1}

Perform mesh adaptation here to velocity model and source freq.

Mesh adaptation

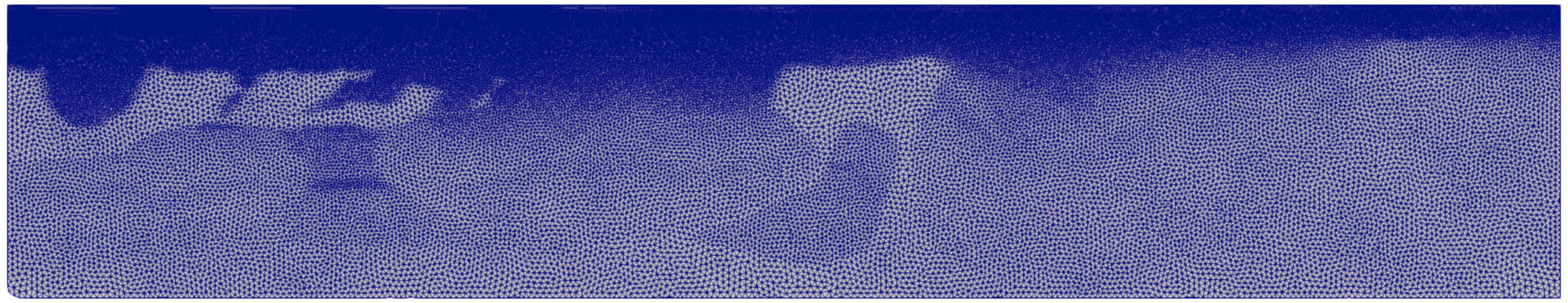
via SeismicMesh

- Completely automatic (no user intervention) occurs during FWI.
    - Using SeismicMesh (Roberts et. 2021)
- Pythonic interface to mesh generator is callable from our Firedrake (Python) code (in serial or parallel).
- Need to interpolate from …
    - *Firedrake.FunctionSpace* -> structured grid -> *Firedrake.FunctionSpace*
    - Linear interpolation.

Mesh adaptation

Parameters

- Primary parameters for mesh adaptation are
  - P-wave velocity.
  - Source frequency+ **g**rid **p**oints **p**er **w**avelength = gpwl
  - Desired simulation timestep (CFL condition) which ensures simulation will not go numerically unstable.
    - In 3D, we use a sliver removal technique to address degenerate elements to bound minimum element quality.



spyro: Full waveform inversion FEM code

Recent advances

- Now using the Rapid Optimization Library (ROL; Rizdal et al. 2017) called via pyROL (https://bitbucket.org/pyrol/pyrol) to do the optimization.
  - Provides interfaces to and implementations of algorithms for gradient-based unconstrained and constrained optimization.
  - Can incorporate proper inner product based on the L_2 inner product on the function space **(for mesh independent optimization behavior)**
    - **Riesz map in gradient calculation.**
  - Anecdotally faster than SciPy's L-BFGS.

spyro: Full waveform inversion FEM code

Recent advances

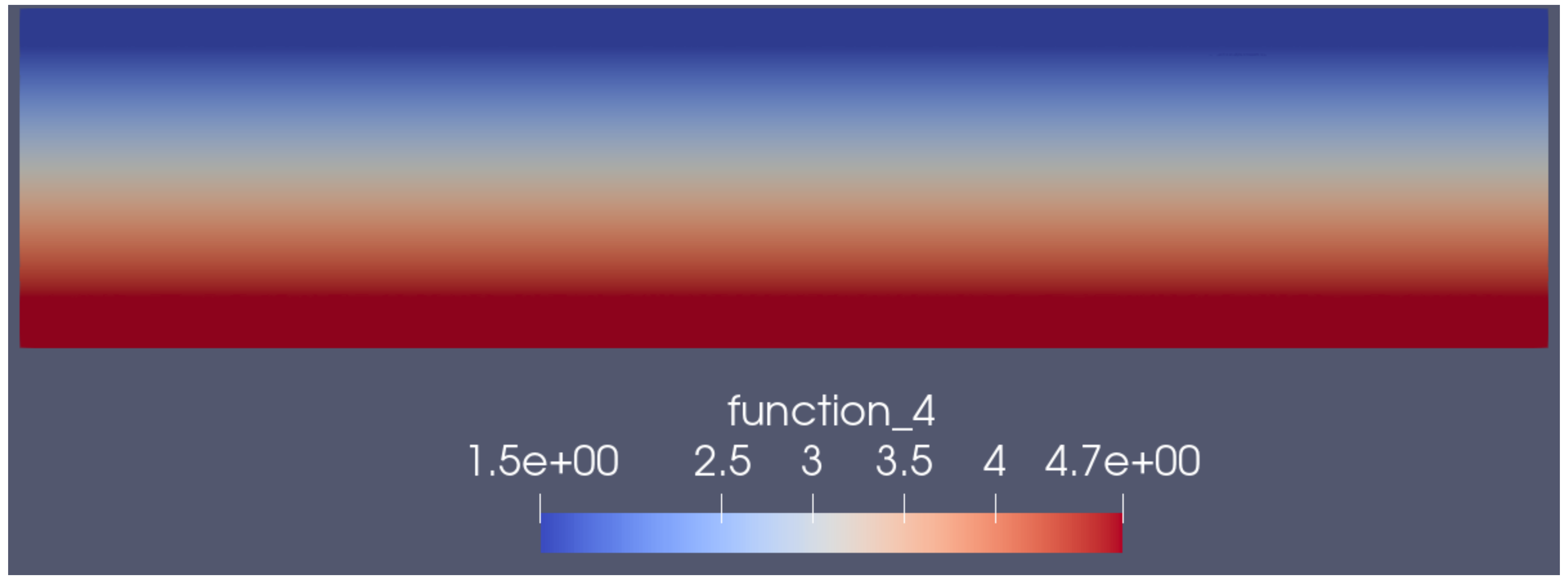
- Can use higher-order (P < 6) mass-lumped simplical elements (KMV) for:
  - Forward, adjoint, and gradient calculation in 2D/3D.
    - Gradient calculation in FEM requires a mass matrix inversion which these elements make significantly faster.
  - Perfectly matched layer (PML) also dramatically benefits from these elements.
    - Auxiliary PML equations need to be solved for each timestep.
- ***Firedrake details***: mixed function spaces and matrix free operations to reduce runtime memory overhead and more concisely pose the numerical schemes with/without PML in 2D/3D.

Universidade
de São Paulo

Benchmark 2D case

with mesh adaptation and time domain multiscale FWI

- Marmousi II model (17 km wide by 3.5 km deep)
  - 40 shots, 301 receivers, 4 second simulation.
  - Exact shots simulated with different grid using different numerics.
  - Linearly varying velocity model as starting model (1.5 km/s to 4.7 km/s)
  - **Gradient downsampling** 0.8 * Nyquist period of source frequency
  - 0.5 km PML on three sides.
  - Free-surface boundary on top



Experimentation

with FWI

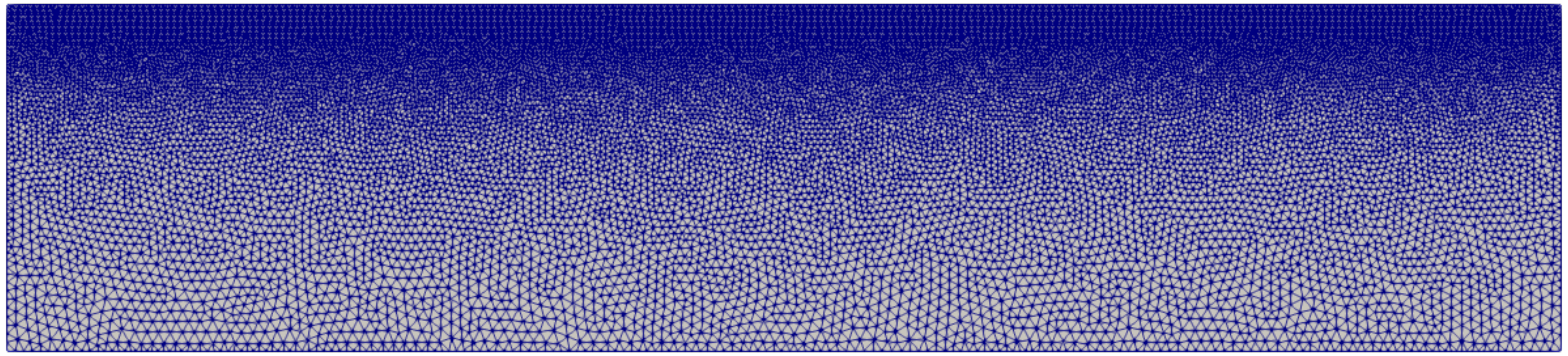
- **EXP001**: Static mesh, mass-lumped (KMV), P=2
- **EXP002**: Static mesh, Continuous Galerkin (CG), P=2
- **EXP003**: Adaptive mesh, mass-lumped (KMV), P=2

Starting meshes

with FWI

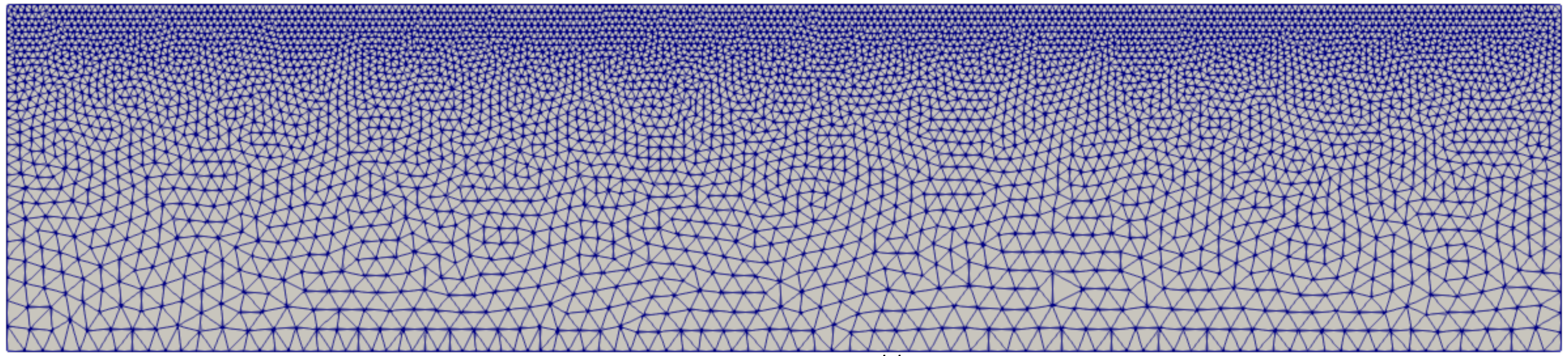
- Starting meshes for static simulations (EXP001, EXP002) need to be refined to **maximum source frequency** (e.g., 8 Hz) and **initial guess model** whereas EXP003 needs to be refined only to **starting source frequency** (e.g., 3 Hz)

Starting mesh for EXP001, EXP002. 5 ggpwl for 8 Hz source frequency



36,237 triangles, 18,534 nodes

Starting mesh for EXP003. 5 ggpwl for 3 Hz source frequency



9,016 triangles, 4,708 nodes

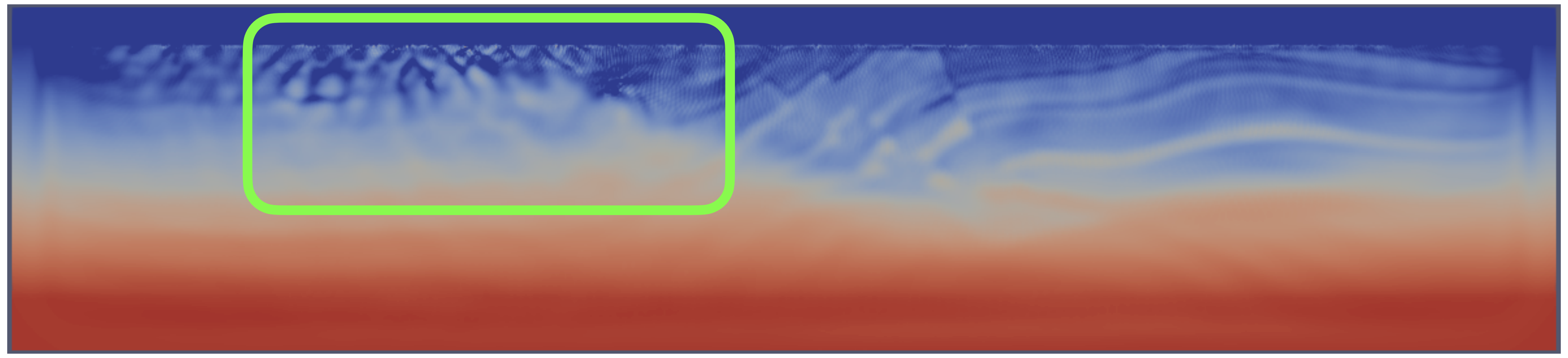
Universidade de São Paulo

Results

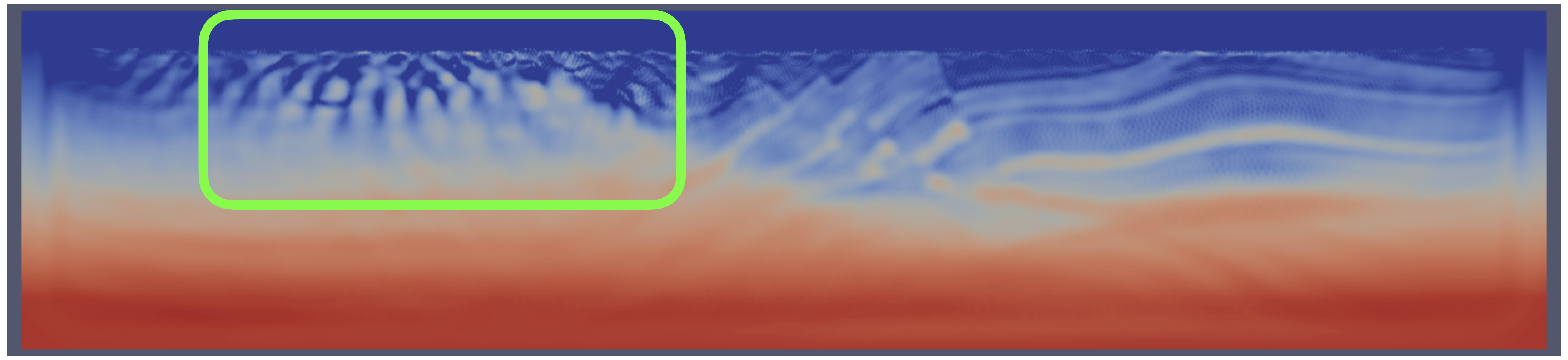
Final models

EXP002

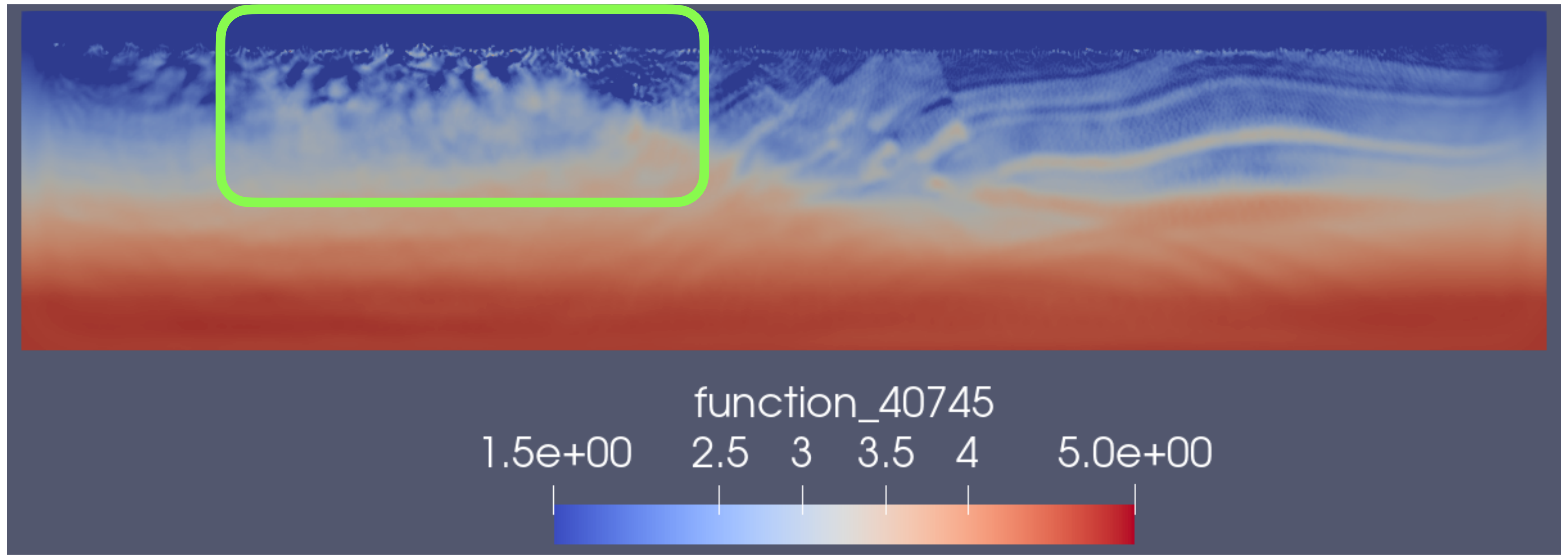
** 53 iterations after time limit on slurm system was reached



EXP001



EXP003

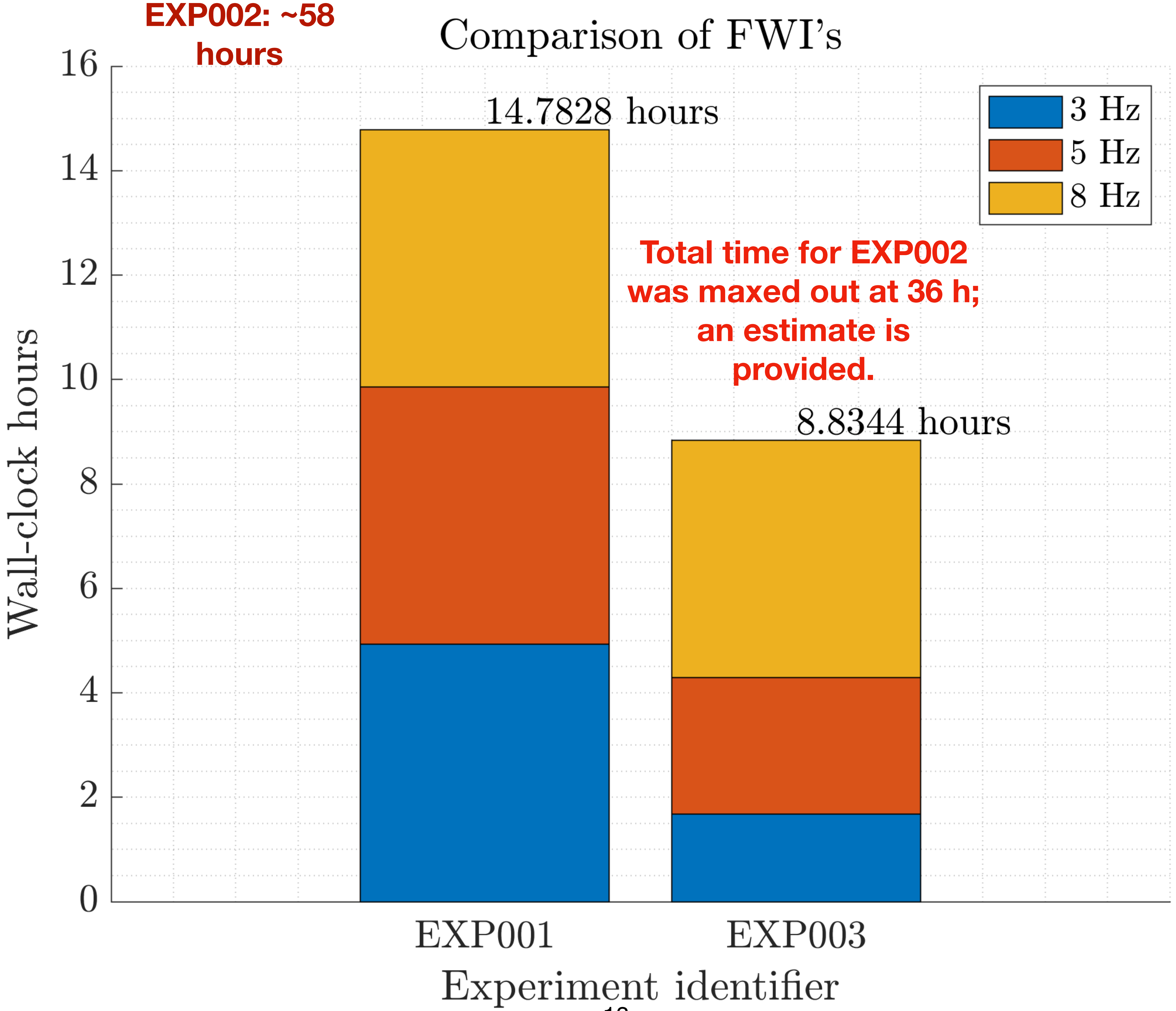


Run times
using ensemble parallelism

- 33% faster with mesh adapt.

- CG is 7-8 x slower than KMV

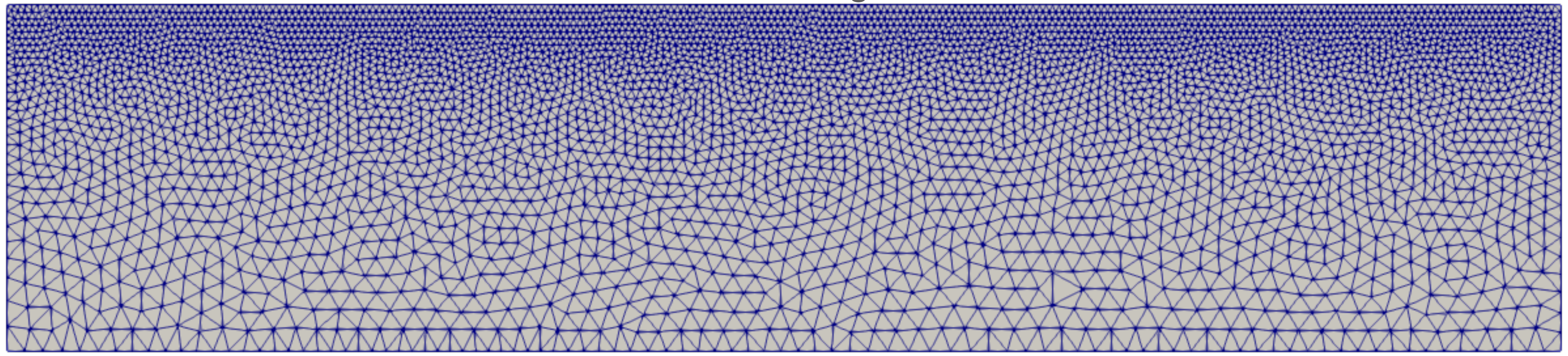
EXP002: ~58 hours



Mesh evolution

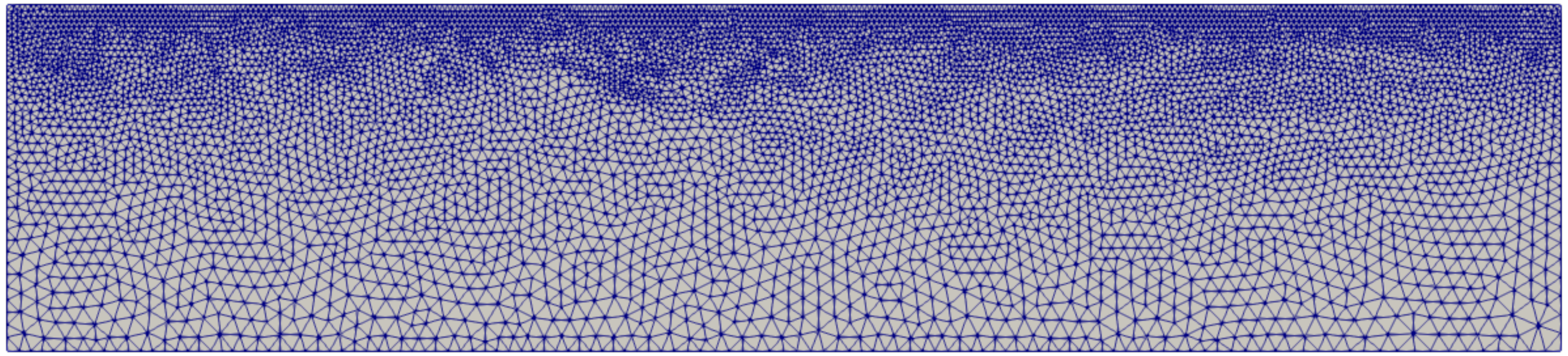
in EXP003 adaption occurs automatically

For the 3 Hz stage



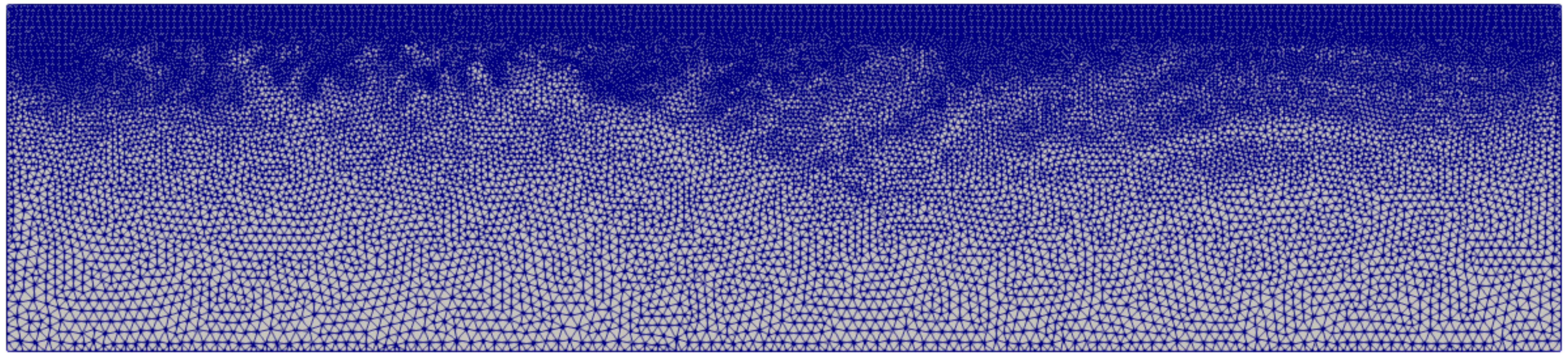
9,016 triangles, 4,708 nodes

For the 5 Hz stage



13,971 triangles, 7,237 nodes

For the 8 Hz stage

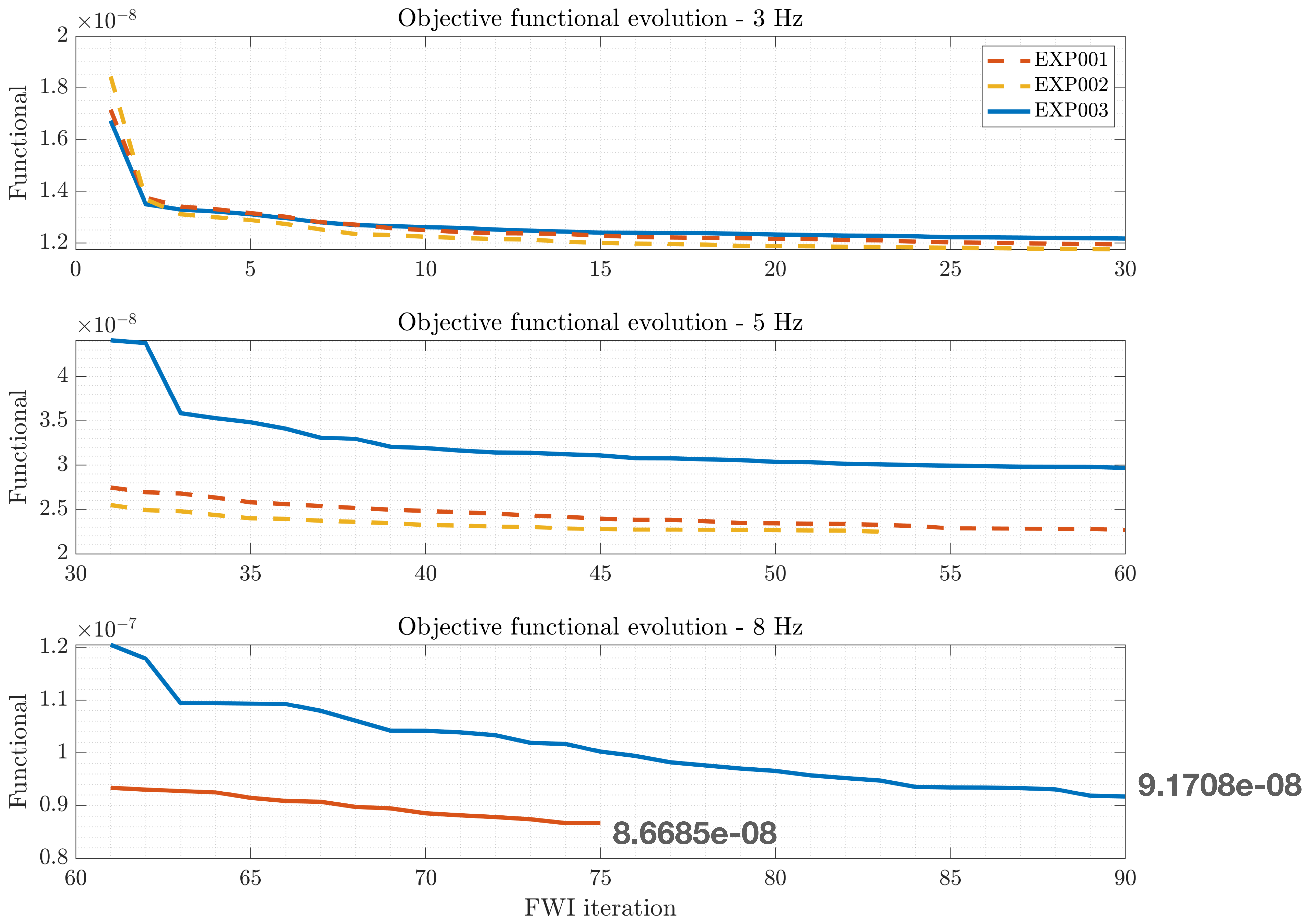


36,333 triangles, 18,593 nodes

Results

Objective functional evolution

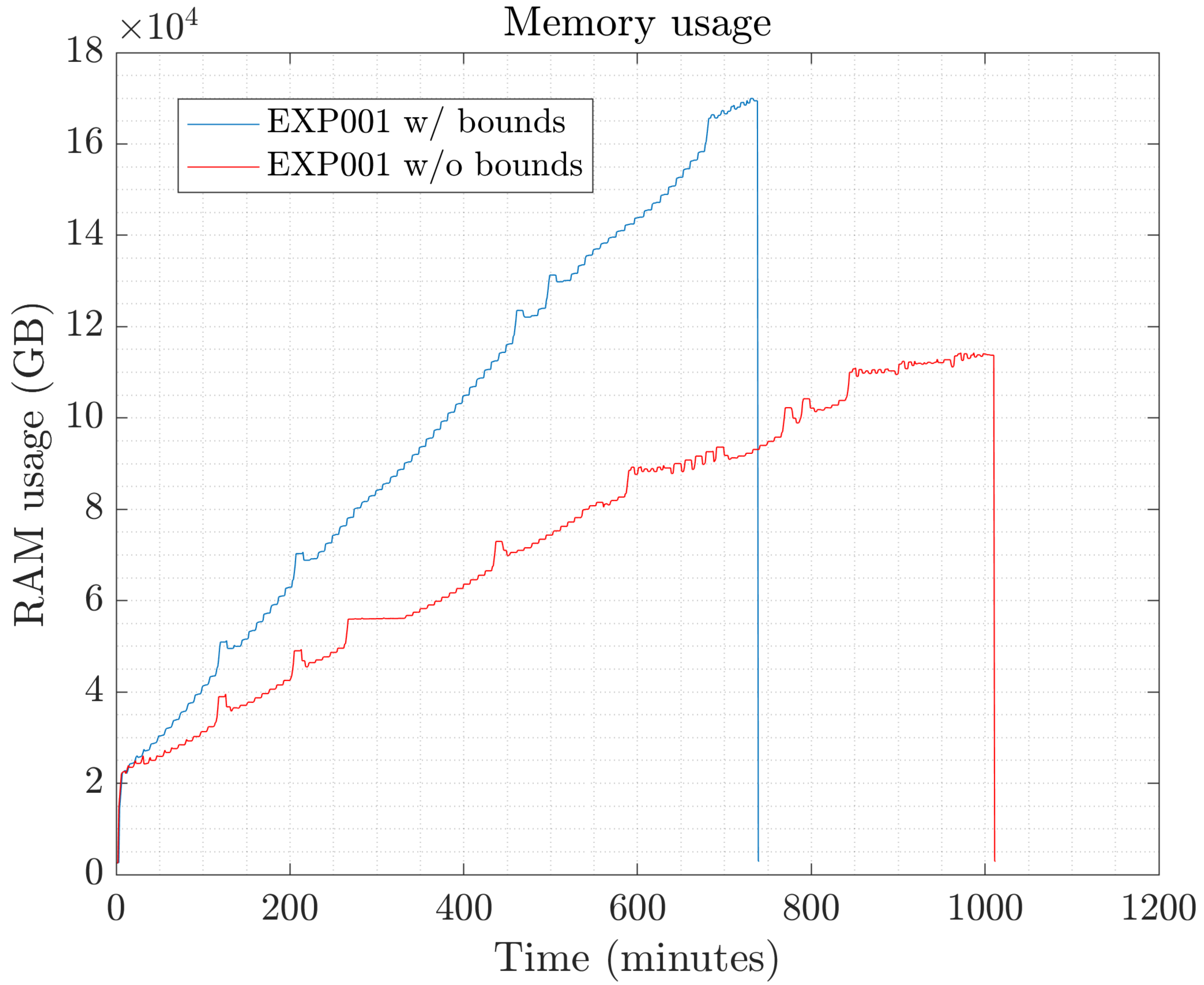
- A overall lower objective functional is reached with static fine mesh but not by much
- CG and KMV perform nearly identically.



Technical problem(s)

memory leak with optimization bounds

Using bounds with pyROL causes memory to grow rapidly. Removing the bounds alleviates the problem but only somewhat.



Next steps

Universidade de São Paulo

- **Multiscale time domain FWI is working with higher order mass lumped elements.**
  - Higher order mass lumped elements dramatically outperform the CG formulation.
- **Multiscale FWI with waveform mesh adaption yields 33% speedups while producing similar final answer as to a fine static mesh.**
- **Summarizing findings in journal article.**
- In 3D, the savings in mesh sizes will be more dramatic.
- Larger variations in velocity ranges may yield more dramatic time saving benefits.
  - For example in Gato do Mato (maximum velocity is up to 7 km/s)

Citations

- Ridzal, Denis, Drew Philip Kouri, and Gregory John von Winckel. *Rapid optimization library*. No. SAND2017-12025PE. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

- Roberts et al., (2021). SeismicMesh: Triangular meshing for seismology. Journal of Open Source Software, 6(57), 2687, https://doi.org/10.21105/joss.02687