Time domain multiscale FWI with waveform adapted meshes

Keith J. Roberts; Rafael Gioria

February 9, 2021

Abstract

In this short report, we present a multiscale time domain, full-waveform inversion (FWI) approach that can reduce the computational cost over conventional approaches. By adapting the mesh to the expected source frequency and properties of wavefield (e.g., P-wavespeed), the number of elements necessary to discretize the domain is reduced and thus the FWI process can be accelerated. Our approach uses a fully-explicit, higher-order (up to $P = 5$) mass lumping method on simplices to solve both state, adjoint, and the objective gradient. We provide an pilot example in 2-D to illustrate the concept, describe how the new workflow deviates from the standard FWI workflow, and demonstrate the benefits over a more traditional approach using finite elements.

1 Motivation

Finite element methods (FEM) can successfully reduce simulation sizes by an order of magnitude over structured approaches when mesh adaptation is exploited [Thrastarson et al., 2020]. Despite this advantage, FEM are generally dismissed in oil and gas exploration velocity building applications because the method requires a solution to a sparse system of equations at each simulation timestep. The computational cost of solving this system of equations can become prohibitive for simulations in a seismic inversion context where many state and adjoint wave solutions are required.

Nonetheless, the computational burden of the large sparse system inherent to FEM can be mitigated with the usage of Spectral Element Method (SEM). The SEM primarily relies on hexahedral meshes to discretize the domain [Patera, 1984] and, in fact, these methods are often preferred for regional and global scale seismic inversion problems as they can more efficiently discretize the domain [e.g, Krischer et al., 2018]. However, perhaps lesser known elements and quadrature rules also exist that permit stable and accurate mass lumping on simplices (e.g., triangles and tetrahedrals), which can be used to accelerate wave simulations on simplical meshes dramatically [Chin-Joe-Kong et al., 1999, Mulder et al., 2014, Geevers et al., 2018b,a]. Despite the existence of these triangular higher-order mass lumped elements, to the authors' knowledge, these have been used infrequently (or not at all) in peer-reviewed literature to perform seismic inversions.

There are some advantages to using simplices over hexaderal elements to discretize a computational domain. Several mesh generation software are readily available for the automatic generation of simplical elements [Alliez et al., 2020, Geuzaine and Remacle, 2009, Roberts et al., 2021]. For instance, Roberts et al. [2021] presented an an approach called SeismicMesh to incorporate the seismic velocity data directly to generate and adapt high-quality 2D/3D triangular meshes [Roberts et al., 2021]. Furthermore, mesh size variation can more readily be exploited using triangular elements as compared to hexahedral ones. Triangular elements permit larger mesh gradation rates and do not require mixed triangular/cube meshes around highly refined boundary layers, for instance. This latter point can greatly simplify the numerical schemes, coding/implementation complexity, and mesh generation effort.

2 Aim and contribution

A multiscale time domain, full-waveform inversion (FWI) approach that can reduce the computational cost over conventional approaches is demonstrated by automatically adapting the mesh to the expected source frequency, properties of wavefield (e.g., P-wavespeed), and considering numerical aspects (e.g., the CFL condition) for robustness.

For the spatial discretization, higher-order (up to $P = 5$) 2D/3D triangular elements are used that permit stable and accurate mass-lumped lumping. Combined with a 2^{nd} order accurate fully-explicit timestepping scheme, rapid and scalable solutions to large wave propagation problems common to seismic inversions are able to be efficiently simulated. These elements have been integrated into the open-source Firedrake library [Rathgeber et al., 2016].

Seismic inversions often employ a multiscale strategy to avoid local minimums in the optimization [**?**]. In this multiscale time-domain strategy, the inversion commences with a relatively low source frequency and progressively increases it after a number of optimization iterations (e.g., 20-30). In our approach, after a number of optimization

iterations in each frequency band of the multiscale strategy, the mesh is adapted to the progressively higher frequency source and the updated seismic velocity wavefield. The benefits of this approach twofold: reduced optimization cost for the the lower frequency bands as mesh adaption yields fewer elements in the discretization while providing more mesh refinement near emergent features in the optimized velocity model.

3 Tools & Methods

3.1 Mesh generation

For mesh generation and adaptation, the SeismicMesh package was used [Roberts et al., 2021]. SeismicMesh is a Python package for simplex mesh generation in two or three dimensions. As an implementation of DistMesh, it produces high-geometric quality meshes at the expense of speed. For increased efficiency, the core package is written in C++, works in parallel, and uses the Computational Geometry Algorithms Library.

SeismicMesh automatically generates graded mesh-density functions from seismological velocity models (e.g., P-wave or S-wave velocity data) that is seamlessly used in the mesh generator to size elements throughout the domain. The program has a Python application programming interface that is automatically called during the inversion to periodically re-mesh the velocity model.

The parameters used in the mesh adaptation step are: the number of grid points per wavelength (ggpwl), the central source frequency, and the anticipated simulation timestep in seconds. Besides this, SeismicMesh performs several automatic mesh quality improvement steps to ensure the simulation will be numerically stable and conform well to the computational domain.

3.2 Wave propagation

For the seismic inversions, we rely on the isotropic acoustic wave equation posed in the 2^{nd} order formulation. Density of the medium is assumed to be constant in this formulation. A Ricker wavelet with a specified central frequency convolved with a sharp Gaussian kernel is used to excite the domain and a number of receivers record the solution each timestep at locations near the top of the domain. A Perfectly Matched Layer [PML Kaltenbacher et al., 2013] is implemented into the wave propagator to absorb outgoing waves and the adjoint is consistent with these developments. Further technical details are deferred for the ongoing manuscript.

All aspects of the inversion rely on the same spatial discretization. In other words, the discrete adjoint and the functional gradient calculation use the same finite elements as the forward discretization.

3.2.1 Higher-order mass lumped elements

The finite elements originally first documented in Chin-Joe-Kong et al. [1999] and later improved upon by several works [Mulder et al., 2014, Geevers et al., 2018b,a] were implemented inside FIAT: Finite element Automator Tabulator [Kirby, 2004]. FIAT relies on the abstraction of the finite element posed by Ciarlet [doi] and is used by the Firedrake library at runtime.

When these elements are paired with special quadrature rules, they yield diagonal mass matrices. This property avoids the need to solve a sparse system of the equation using direct or iterative methods and thus yields dramatic accelerations of the simulation and improved parallel performance.

Triangular elements for $P < 6$ were implemented and the latest tetrahedral elements published [Geevers et al., 2018b] for $P < 5$ were implemented. Note the primary downside to these elements is that they require more nodes per element than tradiational CG elements. However, this cost is easily outweighed by the diagonal mass matrix property. We honorifically named these elements Kong-Mulder-Veldhuizen (KMV) elements after the last names of the authors in the first paper they were documented.

3.2.2 Continuous Galerkin

Our original formulation of FWI relied on Continuous Galerkin (CG) finite elements of arbitrary spatial order to discretize all components of FWI and this serves as a reference for comparison with the KMV elements.

3.3 Time domain FWI with mesh adaptation

The pseudo-code shown in 1 outlines each step in our time-domain multiscale full-waveform inversion code (FWI). Each computing unit (i.e., a core or a group of cores) is responsible for calculating one sensitivity kernel (e.g., functional gradient) and this was done to reduce runtime memory overheads associated with storing several forward wavefields simultaneously.

Our optimization code interfaces with the Rapid Optimization Library (ROL) [Ridzal et al., 2017] using a Python interface called pyROL and thus supports a number of different optimization strategies (e.g., trust-regions, gradient

descent, etc.). In this work, we rely on the quasi-Newton L-BFGS method. We note that mesh independence is handled to take into account the unstructured and variable resolution of the mesh; however, these technical details are deferred to our future manuscript.

Following Bunks et al. [1995] for the multiscale approach, the source is low-pass filtered using a Butterworth filter with a user-specified cutoff frequency. Additionally the so-called "observed" shot records are also low-pass filtered with the same filter configuration. In our numerical experiments to avoid the so-called "inverse crime", the observed shots are simulated using a variable density acoustic wave propagator in the finite difference method.

Our algorithm supports a combination of both ensemble and spatial parallelism on large scale CPU clusters. For instance if one has 10 shots and 40 cores, 4 cores can be used to simulate each of the 10 shots simultaneously producing one final summed sensitivity kernel.

The traditional $L2$-norm objective functional is used and regularization was not pursued.

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

1: **procedure** MULTISCALE FULL WAVEFORM INVERSION
2: $c^0 \leftarrow$ initial velocity model
3: $iter^{max.} \leftarrow$ maximum number of iterations per freq. band
4: $k \leftarrow 0$
5: **for** $f \leftarrow freq_{min.}$ to $freq_{max.}$ **do**
6: Assign source frequency f;
7: **while** $(\nabla J^k > 0 \& J^k > 0) || k < iter^{max.}$ **do**
8: $J^k \leftarrow 0$
9: $\nabla J^k \leftarrow 0$
10: **for** shot $\in$ shots **do**
11: Compute forward simulation for shot;
12: Compute functional and add it to J^k.
13: Compute gradient via discrete adjoint and add to ∇J^k;
14: Given ∇J^k and J^k using L-BFGS produce c_f^{k+1}
15:

3.3.1 The case of mesh adaptation

In the case of mesh adaptation, SeismicMesh is called automatically and during runtime between lines 6 and 7 in Algorithm 1 directly after the new source terms are constructed. In the case that mesh adaption *is not used*, the starting/initial mesh is adapted to the peak source frequency to be used in the multiscale inversion and the initial velocity model and the experiment is referred to as *static*. In the case mesh adaptation is used, then the initial mesh can be adapted to the initial source frequency and initial velocity model and the experiment is refereed to as *adaptive*.

4 Experiments with FWI

The 2D Marmousi II benchmark model [Martin et al., 2005] was chosen for this pilot example. Our experimental configuration used 40 shots equi-spaced at the top of the domain in the water layer at a specified frequency of 10 Hz. The solution was recorded at 301 equi-spaced receivers at the top of the domain below the sources and each shot was simulated for 4.0 seconds. A 500-m PML absorbing zone was included on three sides of the 2D domain leaving the marine region to reflect waves and generate free surface multiples.

To avoid inverse crime, "observed" data were simulated with a Ricker wavelet with a 10 Hz source frequency using a variable-density wave propagator with the finite difference method.

The initial velocity model that was used to commence FWI was a linear varying velocity profile that ranged from 1.5 km/s in the water layer to 4.7 km/s in the deepest parts of the model.

For the FWI, 30 optimization iterations were conducted in a sequence for the low-pass filter cutoffs of 3, 5, and 8 Hz. In other words, 30 iterations were conducted with a low-pass filter cutoff of 3 Hz and then the final velocity model from that stage was used to commence a round of 30 iterations using a low-pass cutoff of 5 Hz, and so on and so forth for the low-pass cutoff of 8 Hz.

The gradient was masked in the water layer. In other words, the gradient was set to zero in the region of the initial velocity model that had a grid point P-wave velocity less than 1.5 km/s.

Three experiments each performing FWI using the identical experimental setup detailed above and only varying the spatial discretization were performed:

1. EXP001: $P = 2$ KMV spatial discretization with a static mesh.

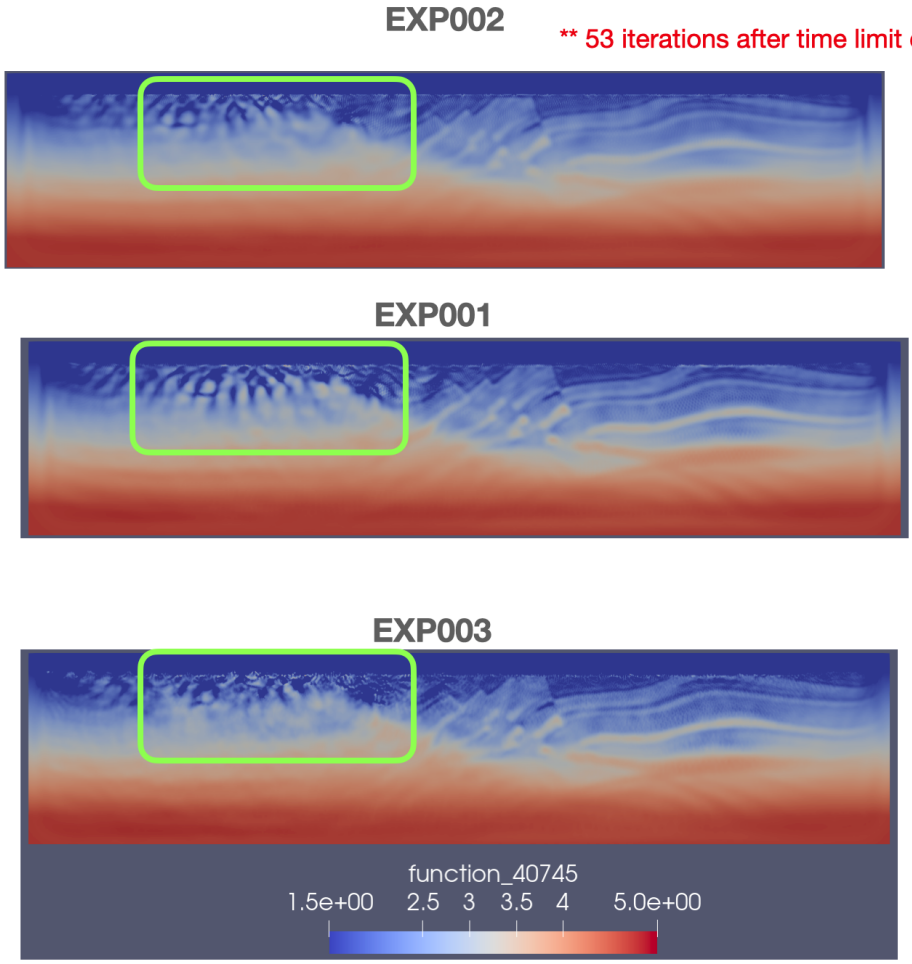

Figure 1: The final P-wave velocity models after all possible FWI iterations were completed. The green box annotates a region detailed in the text. Note EXP002 could not be completed in the time limit set forth by the computer administrators.

2. EXP002: $P = 2$ CG spatial discretization with a static mesh.
3. EXP003: $P = 2$ KMV spatial discretization with a variable mesh

5 Results

5.1 Final results

The final P-wave velocity models can be seen in Figure 1. Qualitatively there is little difference between EXP001 and EXP003. Considering EXP002 did not complete the same number of iterations as EXP001 and EXP003, it is more difficult to reach a statement about its final velocity model. However, all models accurately imaged the right side of the velocity model but struggled more so with the left hand side (Figure 1). In this regard, EXP003 with mesh adaption produced the least accurate images on the left hand side of the velocity model near the small water reservoir.

5.2 Execution time

EXP003 with mesh adaptation completed this experiment approximately 33% faster than the same configuration with a static fine mesh (Figure 2). EXP001 with CG elements was largely non-competitive with an estimated completion time of approximately 58 hours, which is around $6 - 7$ times slower than either EXP001 or EXP003.

Mesh adaption accelerated the lower frequency bands (e.g., 3 and 5 Hz), while producing an approximately similar execution time to EXP002 for the 8 Hz band. This observation can be seen by the variable height of the stacked bars in Figure 2.

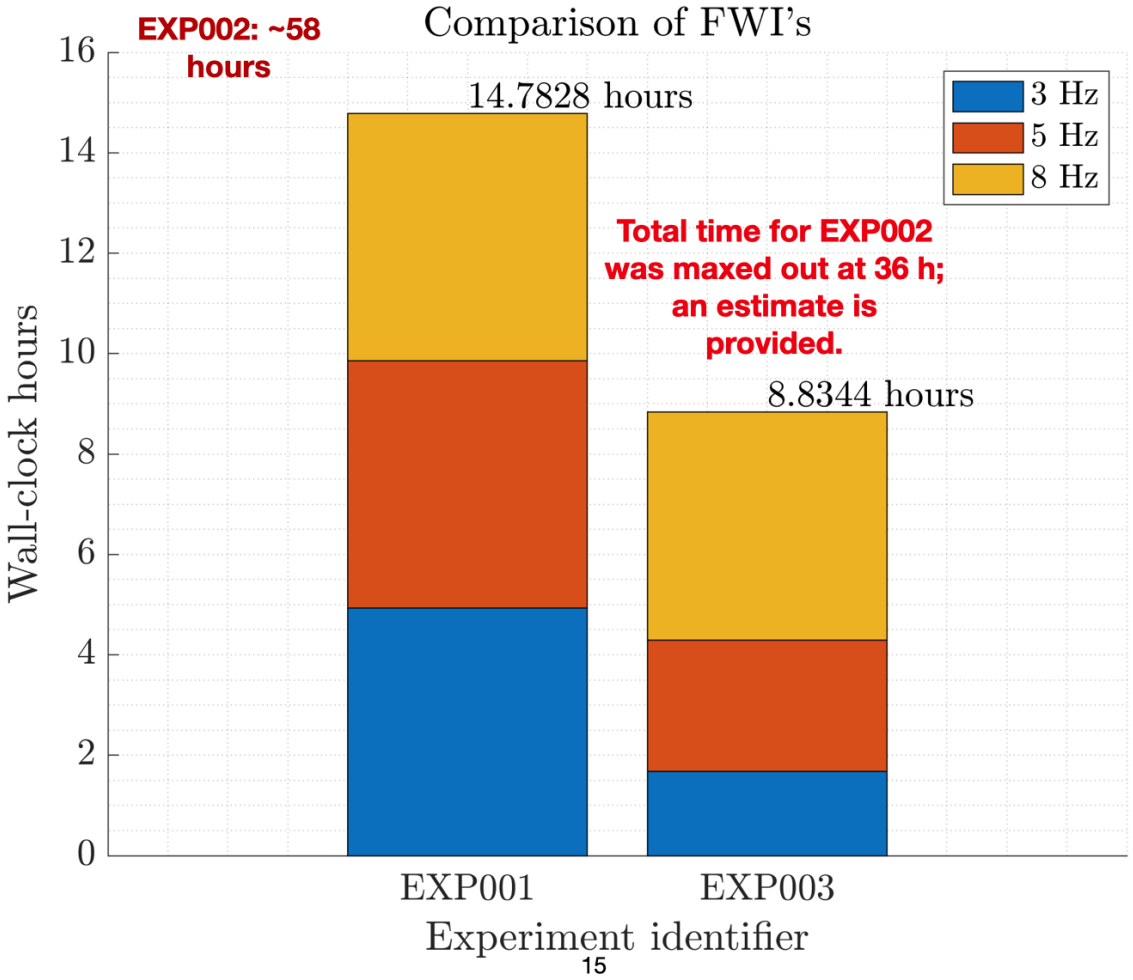

Figure 2: The total wall-clock time to complete the FWI using different spatial discretizations. Note that for EXP002, the maximum simulation time set by the admin. was reached and only an estimate is instead provided.

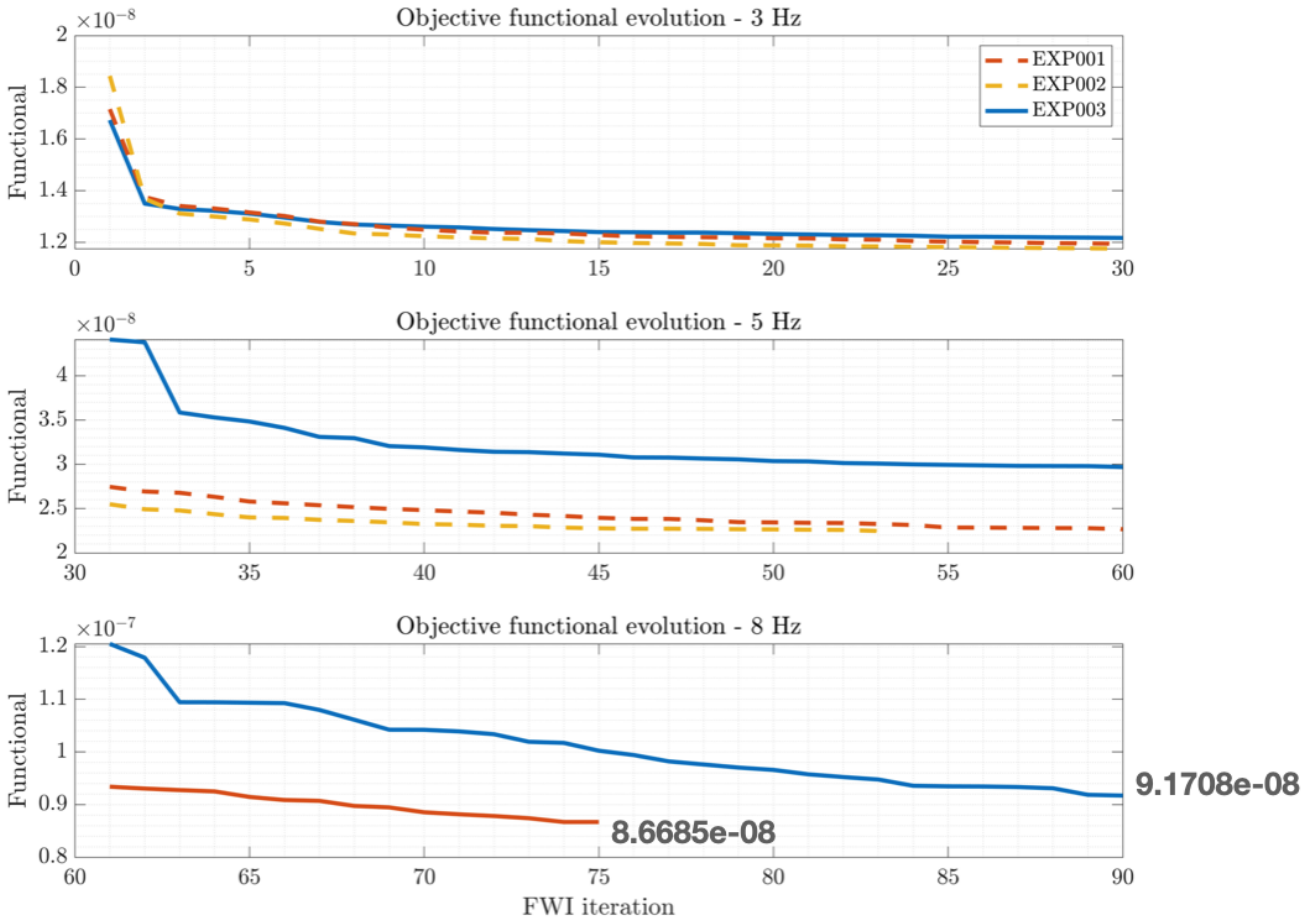

Figure 3: The evolution of the objective functional for the three experiments.

5.3 Objective functional evolution

The time evolution of the objective functional is shown in Figure 3. All experiments (e.g., both KMV and CG elements) performed very similar in the 3 Hz stage albeit significantly faster for the KMV elements (e.g., Figure 2). Similarly to what was observed in Figure 1, EXP002 produces an overall more accurate model by about 10% than EXP003 leading to a final lower objective functional in EXP002.

5.4 Mesh evolution

In EXP003, mesh adaption occupies less than 0.1% of the total simulation time. The size of the mesh ranges by nearly a factor of 4 from 4.7k vertices to 18,593 vertices for the 8 Hz frequency band (Figure 4). Mesh resolution variation becomes particularly evident on the right hand side of the velocity model where stratified layers with higher P-wave velocities lead to larger mesh sizes. In contrast, on the left hand side of the velocity model, mesh resolution distribution begins to spuriously follow numerical artifacts introduced in the inversion process. We note that the adaption procedure in SeismicMesh, namely the ability for it to bound an estimate of the CFL condition in the mesh, avoids fatal numerical instabilities.

6 Conclusions

1. Higher-order mass lumped elements (e.g., KMV elements) dramatically accelerate our time-domain multiscale FWI experiments that had previously used CG discretizations while still producing comparably accurate final images.

   (a) All components of the inversion can use these KMV elements now.

2. Automatic mesh adaptation during different frequency bands of FWI is shown to reduce the computational time per iteration of the lower frequency bands, which leads to a net speedup of the total FWI process.

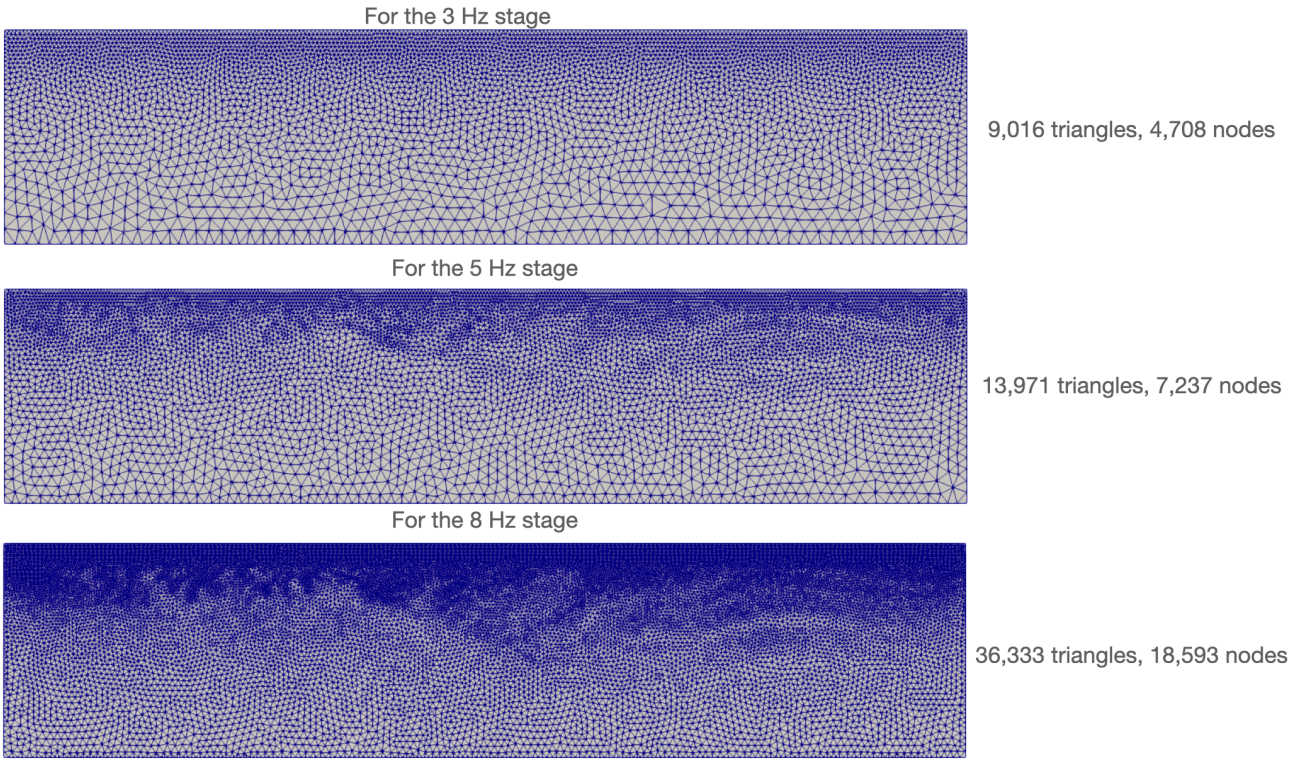

Figure 4: The evolution of the mesh in EXP003. The number of cells and vertices are annotated to the right of the figure.

   (a) Mesh adaptation does not occupy a significant portion of the total FWI time and produced numerically stable simulations.

3. Mesh adaptation however is susceptible to numerical artifacts that occur during the optimization process and leads to slightly less accurate final images in this case.

   (a) Mesh adaptation requires interpolation from a function space to a structured grid and back again which can be a source of error in the inversion process.

7 Codes used

1. For mesh generation: SeismicMesh: https://github.com/krober10nd/SeismicMesh
2. For the FWI experiments: spyro: https://github.com/krober10nd/spyro

References

2. Introduction to the Finite Element Method, pages 36–109. doi: 10.1137/1.9780898719208.ch2. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719208.ch2.

Pierre Alliez, Clément Jamin, Laurent Rineau, Stéphane Tayeb, Jane Tournois, and Mariette Yvinec. 3D mesh generation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.1 edition, 2020. URL https://doc.cgal.org/5.1/Manual/packages.html#PkgMesh3.

Carey Bunks, Fatimetou M Saleck, S Zaleski, and G Chavent. Multiscale seismic waveform inversion. *Geophysics*, 60(5):1457–1473, 1995.

MJS Chin-Joe-Kong, Wim A Mulder, and M Van Veldhuizen. Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation. *Journal of Engineering Mathematics*, 35(4):405–426, 1999.

S. Geevers, W. A. Mulder, and J. J. W. van der Vegt. Dispersion properties of explicit finite element methods for wave propagation modelling on tetrahedral meshes. *Journal of Scientific Computing*, 77:372–396, 2018a.

S. Geevers, W. A. Mulder, and J. J.W. Van Der Vegt. New higher-order mass-lumpe d tetrahedral elements for wave propagation modelling. *SIAM Journal on Scientific Computing*, 40(5):A2830–A2857, 2018b. ISSN 10957197. doi: 10.1137/18M1175549.

Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Meth. Engng.*, 79(11):1309–1331, May 2009. ISSN 0029-5981. doi: 10.1002/nme.2579. URL https://doi.org/10.1002/nme.2579.

B Kaltenbacher, M Kaltenbacher, and I. Sim. A modified and stable version of a perfectly matched layer technique for the 3-d second order wave equation in time domain with an application to aeroacoustics. *Journal of Computational Physics*, 235:407–422, 2013.

Robert C. Kirby. Algorithm 839: Fiat, a new paradigm for computing finite element basis functions. *ACM Trans. Math. Softw.*, 30(4):502–516, December 2004. ISSN 0098-3500. doi: 10.1145/1039813.1039820. URL https://doi.org/10.1145/1039813.1039820.

Lion Krischer, Andreas Fichtner, Christian Boehm, and Heiner Igel. Automated large-scale full seismic waveform inversion for north america and the north atlantic. *Journal of Geophysical Research: Solid Earth*, 123(7):5902–5928, 2018. doi: https://doi.org/10.1029/2017JB015289. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2017JB015289.

Gary S. Martin, Kurt J. Marfurt, and Shawn Larsen. *Marmousi-2: An updated model for the investigation of AVO in structurally complex areas*, pages 1979–1982. 2005. doi: 10.1190/1.1817083. URL https://library.seg.org/doi/abs/10.1190/1.1817083.

W. A. Mulder, E. Zhebel, and S. Minisini. Time-stepping stability of continuous and discontinuous finite-element methods for 3-D wave propagation. *Geophysical Journal International*, 196(2):1123–1133, feb 2014. ISSN 1365-246X. doi: 10.1093/gji/ggt446. URL http://academic.oup.com/gji/article/196/2/1123/579736/Timestepping-stability-of-continuous-and.

Anthony T Patera. A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *Journal of Computational Physics*, 54(3):468–488, 1984. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(84)90128-1. URL https://www.sciencedirect.com/science/article/pii/0021999184901281.

F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software*, 43(3), 2016. Article 24.

Denis Ridzal, Drew Philip Kouri, and Gregory John von Winckel. Rapid optimization library. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

Keith J. Roberts, Rafael dos Santos Gioria, and William J. Pringle. Seismicmesh: Triangular meshing for seismology. *Journal of Open Source Software*, 6(57):2687, 2021. doi: 10.21105/joss.02687. URL https://doi.org/10.21105/joss.02687.

Solvi Thrastarson, Martin van Driel, Lion Krischer, Christian Boehm, Michael Afanasiev, Dirk-Philip Van Herwaarden, and Andreas Fichtner. Accelerating numerical wave propagation by wavefield adapted meshes. part ii: full-waveform inversion. *Geophysical Journal International*, 221(3):1591–1604, 2020.