LLNL-TR-825807

Report on the LLNL Global Full-waveform Inversion Workflow and Progress

N. A. Simmons, C. Morency

August 16, 2021

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Report on the LLNL Global Full-waveform Inversion Workflow and Progress

Nathan A. Simmons & Christina Morency (co-lead investigators)
Lawrence Livermore National Laboratory

August 20, 2021

Background

LLNL has developed the SPiRaL global seismic tomography model based upon millions of body wave travel times and surface wave dispersion curves in the period range of 25-200 seconds (*Simmons et al*., 2021). The motivation to construct the SPiRaL model (which stands for S-, P-, Rayleigh, and Love waves) is based upon the guidance provided by a 3-D model workshop held in Berkeley in 2007 and documented in a paper by *Zucca et al*. (2009). It was concluded in *Zucca et al*. (2009) that a "practical and obtainable" model is one that predicts body wave travel times for event location while also predicting surface wave velocities for moment tensor estimation. We have largely followed that guidance which has led to the construction of SPiRaL. Looking forward, we now seek to improve the global model through full waveform inversion (FWI). This necessitates the development/customization of an efficient and flexible modeling workflow to automate the process using LLNL's supercomputing platforms. This report briefly outlines the work that being performed at LLNL to develop the necessary tools and framework to update the SPiRaL global model with full waveform information.

SPiRaL-SPECFEM Interfacing

The SPiRaL model is complex relative to most other global seismic tomography models. These complexities include:

- Multiple resolution scales (node spacing ranging from 0.25 to 2 degrees),
- Direct incorporation of multiple crustal units including water, ice, sediments and crystalline units
- Undulating surfaces including discontinuities in the crust and transition zone
- Full vertical transverse isotropy (VTI) treatment and parameters

Therefore, special treatment is required to properly translate the native SPiRaL format (spherical tessellation based) to a finite element representation required to perform waveform simulations with SPECFEM3D_GLOBE (*Komatitsch and Tromp*, 2002a, 2002b; *Komatitsch et al*., 2015).

The proper translation involves two primary steps including **sampling the model** to sets of grids using the LLNL-Earth3D software and **model incorporation** into SPECFEM3D_GLOBE using custom Fortran codes. The model sampling strategy involves breaking the Earth into latitude bands and sampling on regular latitude/longitude grids with resolution scales tailored to properly sample the high-resolution regions in the SPiRaL model (Figure 1). The crust is sampled separately from the mantle and the file formats follow the general format of the Crust1.0 model files provided in SPECFEM, with the exception that 5 stress-strain coefficients

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Security, LLC, Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

are provided in individual files to describe the vertical transverse isotropy (*C11*, *C13*, *C33*, *C44*, and *C66*). Below the crust, the model is initially flattened (undulations removed) and broken into depth zones bounded by major discontinuities on each side (Figure 2). A set number of samples are taken within each of the zones, spread out evenly with depth. In the example described in Figures 1-2, the mantle is broken into 13 latitude bands and 3 depth zones providing 13x3=39 subdomains. Additional information is stored to recover the undulations during model ingestion into SPECFEM3D_GLOBE.

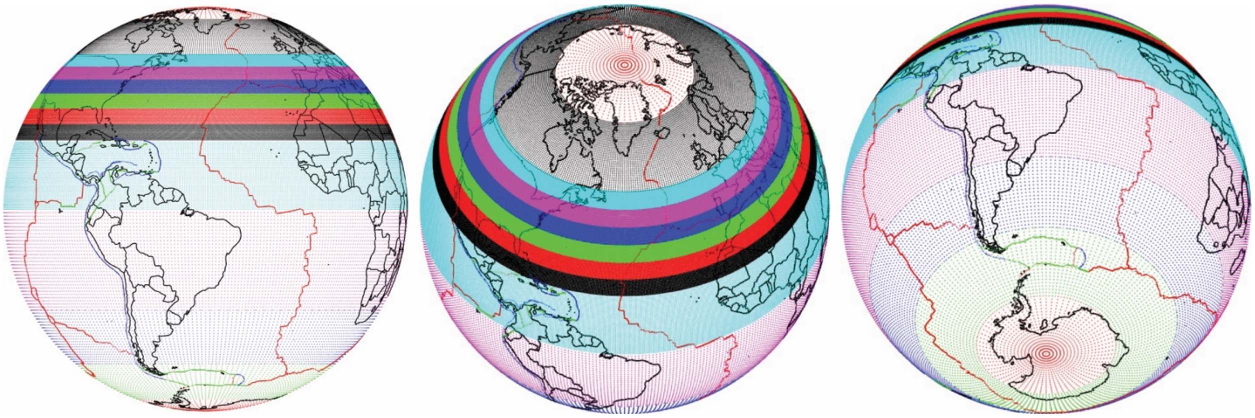

Figure 1. Sampling strategy using latitude bands. In this case, the Earth is broken into 13 latitude bands with high resolution bands in the northern hemisphere across North America and Eurasia, and low- resolution bands at the poles. Regular grids with LLNL-Earth3D user defined Δ*Lat* and Δ*Lon* spacings are constructed and output to separate files to be read into SPECFEM3D_GLOBE for computing finite element properties.

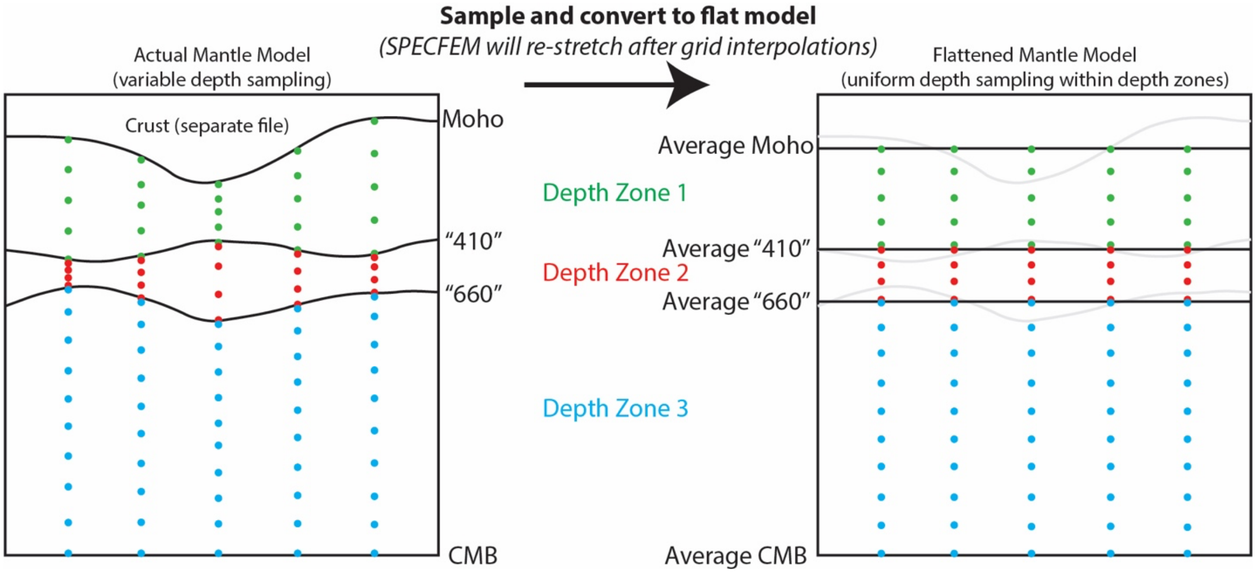

Figure 2. Mantle sampling strategy using depth zones. Not only is the mantle model broken into latitude bands, but also depth regions. In the case above, the mantle is broken into 3 major "depth zones" including: the upper mantle (Depth Zone 1), transition zone (Depth Zone 2), and the lower mantle (Depth Zone 3). The mantle is sampled at an evenly set of points within each zone between the undulating surfaces (left box). The effective depth sampling is even after conversion to a flat model that is more easily ingested by SPECFEM. The topography of the transition zone surfaces is also output, and the mantle is re-stretched by SPECFEM subroutines.

Several code additions/modifications were made to the most recent version of SPECFEM3D_GLOBE to construct the finite element version of the model. Figure 3 illustrates the basic meshing and finite element construction process. Using the sampling strategy described above and in Figures 1-2, the computation of finite element properties can be performed efficiently with the new and modified Fortran codes since the model is initially flattened and sampled over regular grids. After initial interpretation of the model on "simple" finite element meshes, an additional step is required to effectively stretch the elements to reintroduce the undulations and preserve the shape and material properties across discontinuities. The sampled SPiRaL model (version 1.4) and code modifications will soon be included in new versions of SPECFEM3D_GLOBE and available in the GitHub repository (https://github.com/geodynamics/specfem3d_globe). The SPiRaL model and additional/modified codes were employed to compute waveforms for approximately 1000 event-station pairs in a waveform prediction comparison study that is part of a paper accepted to *Geophysical Journal International* in July 2021 (*Simmons et al.* 2021).

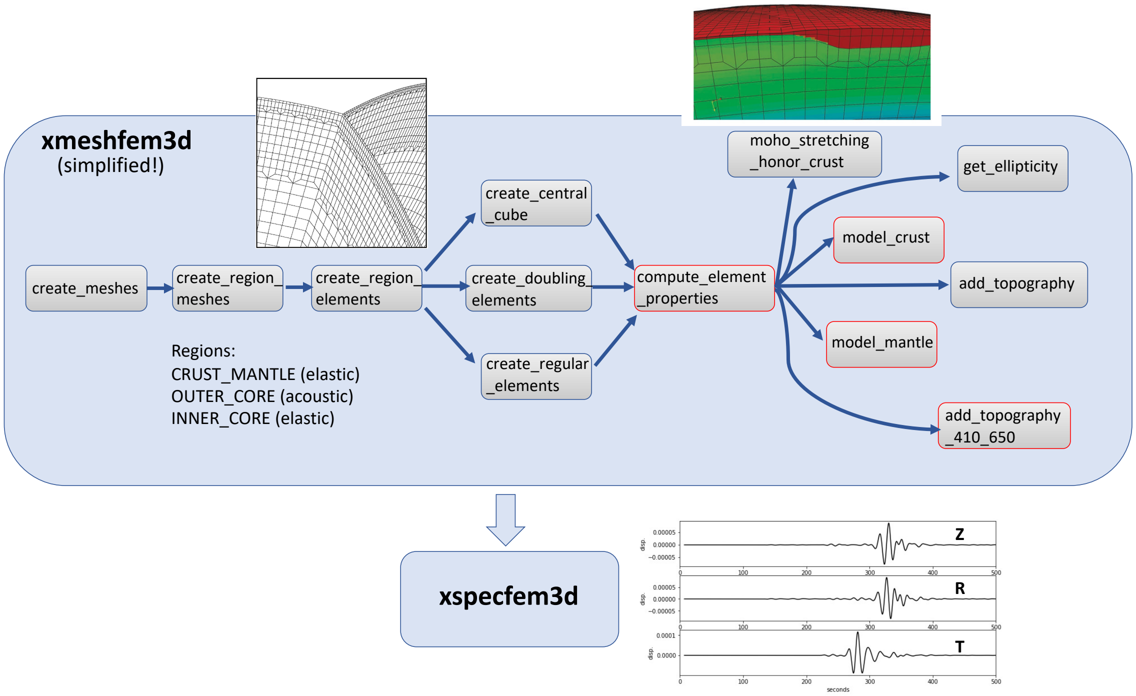

Figure 3. Meshing and interpreting the SPiRaL model using SPECFEM3D_Globe utilities and additions. The standard package includes Fortran codes (xmeshfem3d and subroutines) to construct meshes and populate the properties of the mesh with a standard set of models. Several code additions/modifications were made to the code package to properly populate and alter the finite elements to account for the undulations/discontinuities in the SPiRaL model. The red boxes highlight the primary portions of the process which required customization. After model ingestion, full waveform simulations can be performed (with xspecfem3d).

FWI Workflow Setup and Development

Adjoint tomography is a common method for full waveform inversion (FWI). The seismic adjoint methods stem from the theory developed by *Tarantola* (1984) and further development by later works including *Tromp et al.* (2005) which lay out some of the fundamental mathematical framework and application methods. The first major steps involved in adjoint tomography include forward wavefield simulations, reversed-time simulations of adjoint sources computed from some measure of misfit between data and synthetics, and development of sensitivity kernels from interaction of the forward and adjoint wavefields. An illustrated example of the development of sensitivity kernels is shown in *Tape et al.* (2007) and Figure 4.

$$K_i(\mathbf{x}) \;=\; \int_0^T \mathbf{D}(\mathbf{x},\, t) \;:\; \mathbf{D}_i^{\dagger}(\mathbf{x},\, T-t)\; dt$$

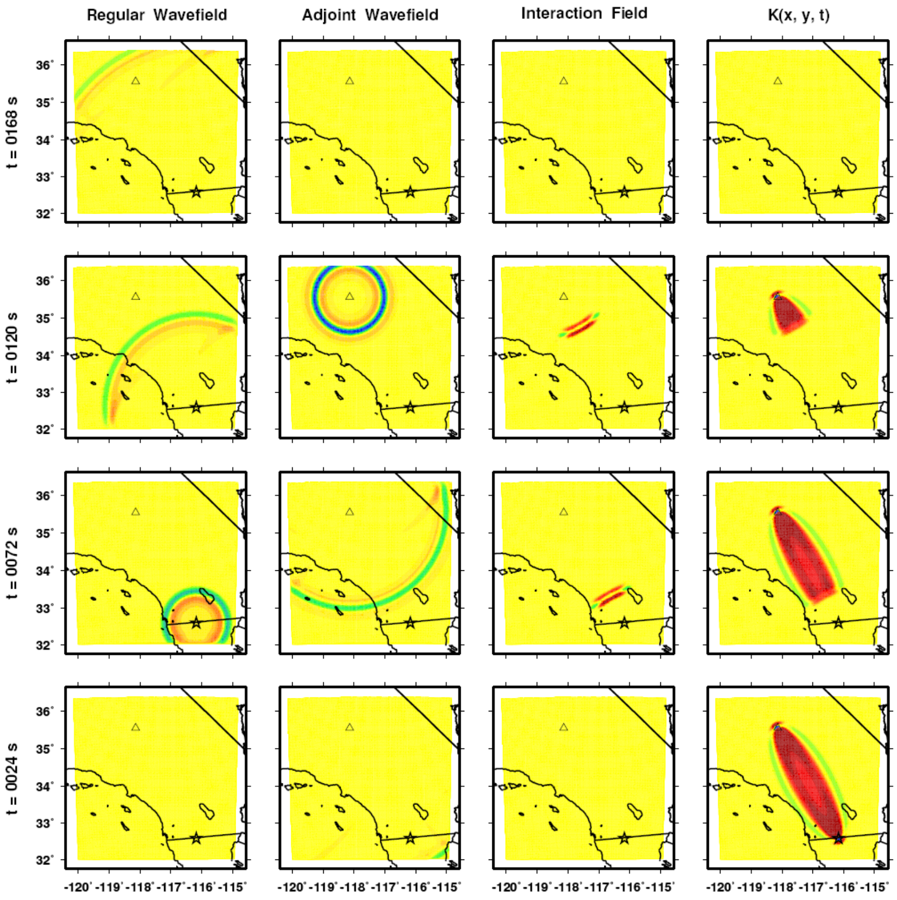

Figure 4. An illustration of the development of model sensitivity kernels using forward and adjoint simulations from *Tape et al.* (2007). The forward wavefield ("Regular Wavefield") emanating from a source (located at the star) is shown in the left column for four snapshots in time. Adjoint sources are developed from the differences between data and synthetics and these sources are propagated from the station (located at the triangle) in reversed time (2nd column). The product of these two wavefields creates the interaction field (3rd column) and the integration of this field over time and space creates the sensitivity kernel (4th column).

The calculation of sensitivity kernels similar to the one shown in Figure 4 can become computationally intensive which is determined by the minimum period computed and the number of required time steps. For many cases, including our case, the computation of the kernels necessitates high-performance computing (HPC). The actual inversion process involves several other steps including waveform data processing and ingestion, model updating through nonlinear optimization, and repeating the process with iterative loops (Figure 5). Collectively, these steps comprise the FWI workflow.

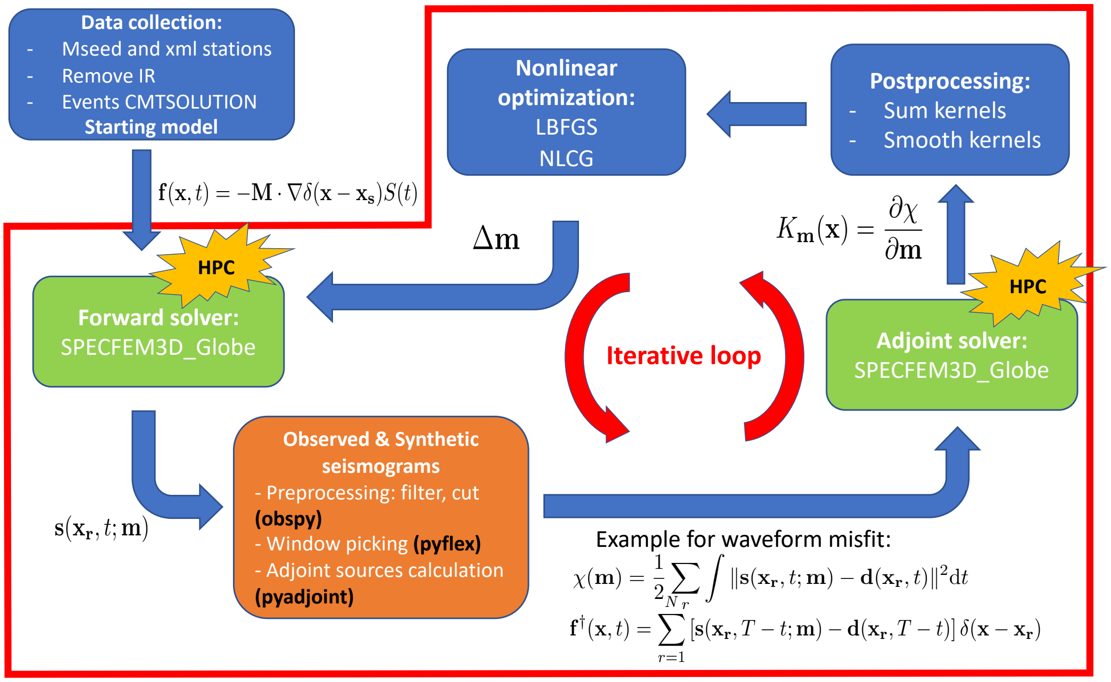
Data collection:
- Mseed and xml stations
- Remove IR
- Events CMTSOLUTION
Starting model
$\mathbf{f}(\mathbf{x},t) = -\mathbf{M}\cdot\nabla\delta(\mathbf{x}-\mathbf{x_s})S(t)$
Nonlinear optimization:
LBFGS
NLCG
Postprocessing:
- Sum kernels
- Smooth kernels
$\Delta\mathrm{m}$
$K_{\mathbf{m}}(\mathbf{x}) = \frac{\partial\chi}{\partial\mathbf{m}}$
HPC
Forward solver:
SPECFEM3D_Globe
Iterative loop
HPC
Adjoint solver:
SPECFEM3D_Globe
$\mathbf{s}(\mathbf{x_r},t;\mathbf{m})$
Observed & Synthetic seismograms
- Preprocessing: filter, cut (obspy)
- Window picking (pyflex)
- Adjoint sources calculation (pyadjoint)
Example for waveform misfit:
$\chi(\mathbf{m}) = \frac{1}{2}\sum_{N_r}\int \|\mathbf{s}(\mathbf{x_r},t;\mathbf{m}) - \mathbf{d}(\mathbf{x_r},t)\|^2 \mathrm{d}t$
$\mathbf{f}^{\dagger}(\mathbf{x},t) = \sum_{r=1}[\mathbf{s}(\mathbf{x_r},T-t;\mathbf{m}) - \mathbf{d}(\mathbf{x_r},T-t)]\,\delta(\mathbf{x}-\mathbf{x_r})$

Figure 5. Schematic workflow for adjoint seismic waveform tomography. The process involves waveform data collection/injection and several steps within an iterative loop including forward and adjoint simulations with SPECFEM3D_Globe executed with high-performance computing (HPC) (green boxes), adjoint source calculations based on some measure of waveform misfit (orange box), postprocessing and non-linear optimization using LBFGS or NLCG algorithms (blue boxes).

The workflow process requires several steps and application of numerous algorithms in addition to performing simulations with SPECFEM3D_GLOBE. Fortunately, there are several open-source code packages that have been developed primarily by the academic community. The most recent codes we have identified and evaluated are Python-based which are listed here:

- ObsPy – A python toolbox for seismology
  - Several tools for data gathering, processing, visualization, etc.
  - Source: https://github.com/obspy/obspy/wiki
  - References: *Beyreuther et al.* (2010); *Megies et al.* (2011); *Krischer et al.* (2015)
- Pyflex – Automated waveform window picking tool
  - Extension of the FLEXWIN algorithm (*Maggi et al*., 2009)
  - Selects waveform windows based on data/synthetic similarity parameters

    - Source: https://krischer.github.io/pyflex/
    - Reference: *Krischer and Casarotti* (2015)
- Pyadjoint – Python package to calculate adjoint sources
    - Can be used in conjunction with Pyflex to compute windowed adjoint sources for SPECFEM simulation
    - Source: https://krischer.github.io/pyadjoint/
- PyASDF – Python library to read and write ASDF files
    - **ASDF**=**A**daptable **S**eismic **D**ata **F**ormat
    - Convenient and compact container format for seismic data
    - Compatible with SPECFEM I/O
    - Source: https://github.com/SeismicData/pyasdf
- Pyatoa – Waveform assessment package
    - Misfit quantification tool for adjoint tomography
    - Source: https://github.com/bch0w/pyatoa
    - Reference: *Chow et al*. (2020)
- SeisFlows – Flexible waveform inversion software
    - General framework for waveform inversion workflow
    - Interfaces with SPECFEM for waveform simulations
    - Source: https://github.com/rmodrak/seisflows
    - Reference: *Modrak et al.* (2018)

SeisFlows provides the basic framework for automation of the waveform inversion process (*Modrak et al*., 2018). SeisFlows interfaces with SPECFEM for waveform simulations (forward and adjoint) and has built-in non-linear optimization algorithms including non-linear conjugate gradient (NLCG) and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) methods. In addition, SeisFlows has built-in options for distributed computing including Slurm job execution which is highly scalable for large batch jobs on supercomputers. Since SeisFlows contains many of the basic elements needed to automate our full waveform inversion process, we chose to adopt SeisFlows as our backbone workflow framework to build upon.

We tested SeisFlows with very small 2D inversion problems executed on laptops, and larger 3D problems executed on the LLNL’s Ruby supercomputer. We identified and made several necessary modifications and additions based on these tests and our needs going forward (listed below). With these modifications and additions, we successfully executed multiple synthetic “checkerboard” full waveform inversion tests with the modified SeisFlows package (referred to as SeisFlows+LLNLmod). These tests are illustrated in Figures 6-8.

Modifications and additions to SeisFlows and SPECFEM:

- Compatibility issues with Python 3.x
    - Support for Python 2.x packages has ended, requiring shifting to Python 3.x
    - Several minor modifications were made throughout the SeisFlows package for compatibility with Python 3.x and packages including newer versions of ObsPy, etc.
- Issues with the latest SPECFEM2D

    - Acoustic kernels in SPECFEM2D have different names than those hardwired in SeisFlows
        - Included a fix to rename kernels in seisflows/solver/base.py
        - Made corrections to SPECFEM2D save_adjoint_kernels.f90
    - The SH model failed with a checkerboard test
        - Made corrections to SPECFEM2D compute_kernels.f90
- Kernel smoothing issues
    - SeisFlows SPECFEM solver plugins set for smoothing with single CPU simulations
        - Modified to use SPECFEM xsmooth_sem utility to allow for multiple CPUs
    - Added anisotropic Gaussian smoothing capabilities/inputs to workflow
- SeisFlows/SPECFEM3D_Globe functionality issues
    - The included specfem3d_globe.py was not compatible with workflow
        - Heavily edited Python solver scripts in SeisFlows to work with most recent SPECFEM3D_Globe version
- Deployment issues on LLNL's Ruby supercomputer
    - Full-scale jobs require communication across compute nodes (Ruby has 1480 nodes and 56 cores/node) and SeisFlows uses Slurm workload management tools to execute the large batch jobs
        - The SeisFlows slurm_lg.py system call fails on LLNL HPC systems
            - Job status communication across compute nodes fails since the Munge background system daemon does not run on LLNL HPC compute nodes (only login nodes)
            - SeisFlows job status functionality was re-written using alternative job monitoring tools that do not require the Munge daemon
            - Full-scale SeisFlow Slurm batch jobs now execute successfully across multiple LLNL HPC systems (Ruby, Quartz, etc.)
- SeisFlows supplies a limited set of waveform misfit definitions/adjoint source possibilities
    - Customization is required to consider more complex misfit/adjoint source calculations
        - The Pyflex automated window picking tool was incorporated into the package to extract specific time series windows that meet certain similarity criteria between data and synthetics
        - The Pyadjoint adjoint source calculation tool was incorporated into the package to compute adjoint sources using the automatically picked windows within the workflow
        - Developed custom adjoint source input functionality to the workflow execution

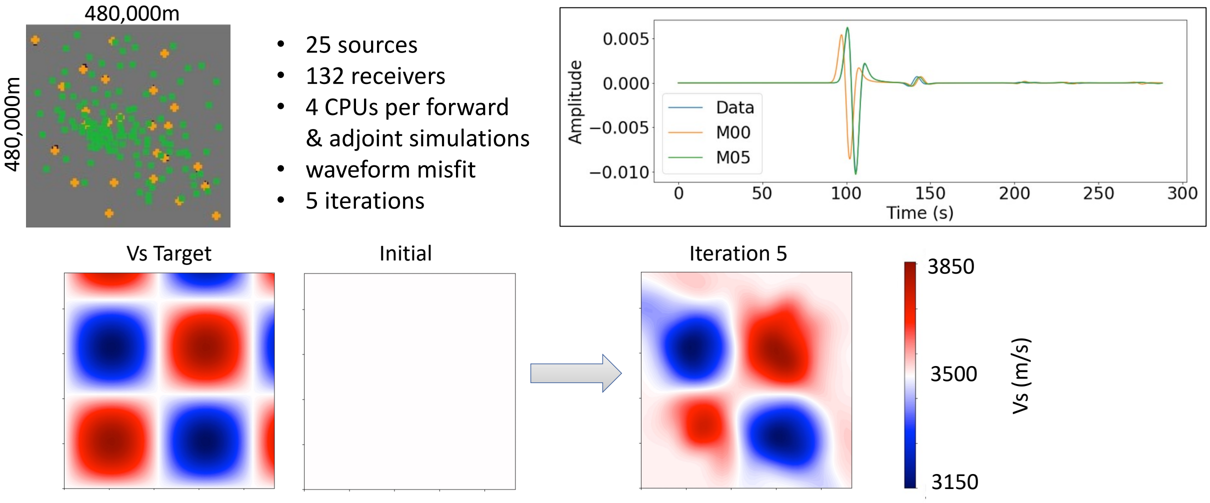

Figure 6. Synthetic 2D checkerboard test with SeisFlows+LLNLmod and SPECFEM2D.

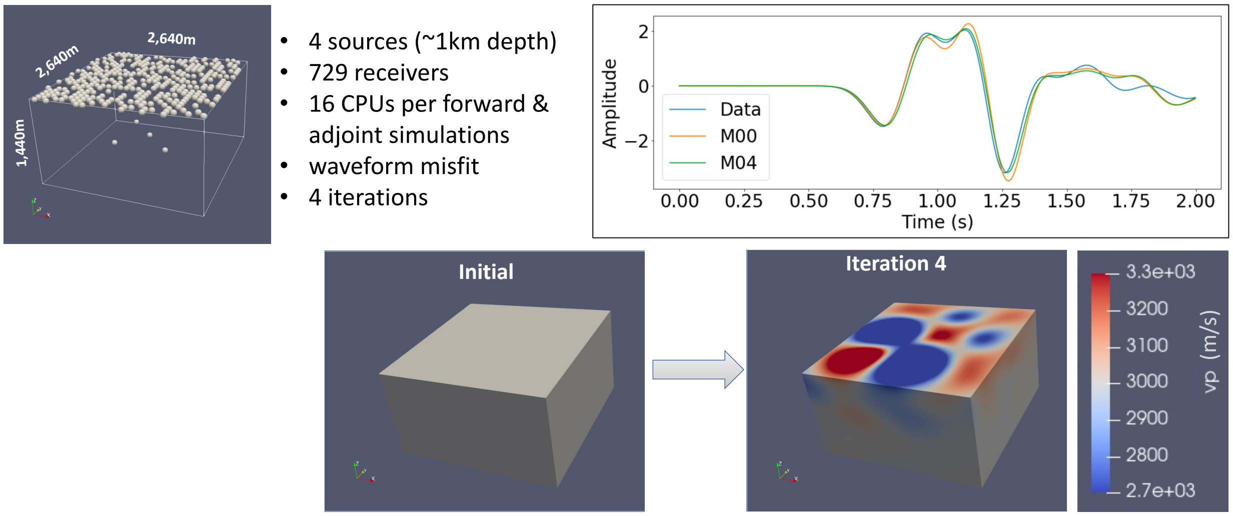

Figure 7. Synthetic 3D checkerboard test with SeisFlows+LLNLmod and SPECFEM3D_Cartesian.

Summary and Next Steps

Several necessary steps to develop an automated full waveform inversion workflow to update the global SPiRaL model on LLNL supercomputers has been completed. These include: 1) interfacing SPiRaL with SPECFEM3D_Globe with new LLNL-Earth3D codes, 2) creation of a new version of SPECFEM3D_Globe including SPiRaL and future iterations (shared with AFTAC in July 2021 and soon to be included in the official GitHub package), 3) modified versions of SPECFEM and the SeisFlows base workflow package to execute properly on LLNL's HPC platforms, 4) expanded functionality of SeisFlows for the global tomography problem, and 5) successfully executed several synthetic tests with the modified/expanded version of SeisFlows (SeisFlows+LLNLmod) on the Ruby supercomputer. The next steps include the integration of the PyASDF library into SeisFlows+LLNLmod for more efficient waveform data management, initial SPiRaL model updating with waveforms from a small set of events, global event selection and waveform data collection, and full-scale inversion with a larger set of global seismic events.

References

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A Python Toolbox for Seismology. *Seis. Res. Lett.*, 81(3), 530-533, doi: 10.1785/gssrl.81.3.530.

Chow, B., K. Yoshihiro, C. Tape, R. Modrak, and J. Townsend (2020). An automated workflow for adjoint tomography-waveform misfits and synthetic inversion for the North Island, New Zealand, *Geophys. J. Int.*, 223(3), 1461-1480, doi: 10.1093/gji/ggaa381.

Komatitsch, D.; Tromp, J. (2002a), Spectral-element simulations of global seismic wave propagation-I. Validation, *Geophys. J. Int.*, 149 (2) , 390-412, doi: 10.1046/j.1365-246X.2002.01653.x.

Komatitsch, D.; Tromp, J. (2002b), Spectral-element simulations of global seismic wave propagation–II. Three-dimensional models, oceans, rotation and self-gravitation, *Geophys. J. Int.*, 150 (1) , 303-318.

Komatitsch, D.; Vilotte, J.-P.; Tromp, J.; Afanasiev, M.; Bozdag, E.; Charles, J.; Chen, M.; Goddeke, D.; Hjorleifsdottir, V.; Labarta, J.; Le Goff, N.; Le Loher, P.; Liu, Q.; Maggi, A.; Martin, R.; McRitchie, D.; Messmer, P.; Michea, D.; Nissen-Meyer, T.; Peter, D.; Rietmann, M.; de Andrade, S.; Savage, B.; Schuberth, B.; Siemenski, A.; Strand, L.; Tape, C.; Xie, Z.; Zhu, H. (2015), SPECFEM3D GLOBE v7.0.0 [software], *Computational Infrastructure for Geodynamics*.

Krischer, L., T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann (2015). ObsPy: a bridge for seismology into the scientific Python ecosystem, *Comp. Sci. Disc.*, 8(1), 014003, doi: 10.1088/1749-4699/8/1/014003.

Krischer, L., and E. Casarotti, (2015, September 30). pyflex: 0.1.4 (Version 0.1.4). Zenodo. Doi: 10.5281/zenodo.31607.

Maggi, A., C. Tape, M. Chen, D. Chao, and J. Tromp (2009). An automated time-window selection algorithm for seismic tomography. *Geophys. J. Int.*, 178(1), 257-281, doi: 10.1111/j.1365-246X.2009.04099.x.

Megies, T., M. Beyreuther, R. Barsch, L. Krischer, and J. Wassermann (2011). ObsPy - What can it do for data centers and observatories? *Ann. Geophys.*, 54(1), 47-58, doi: 10.4401/ag-4838.

Modrak, R. T., D. Borisov, M. Lefebvre, and J. Tromp (2018). SeisFlows-Flexible waveform inversion software. *Comp. Geosci.*, 115, 88-95, doi: 10.1016/j.cageo.2018.02.004.

Simmons, N. A., S.C. Myers, C. Morency, A. Chiang, and D. R. Knapp (2021). SPiRaL: A multi-resolution global tomography model of seismic wave speeds and radial anisotropy variations in the crust and mantle. *Geophys. J. Int.*, doi: 10.1093/gji/ggab277.

Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation, *Geophysics*, 49, 1259–1266.

Tromp, J., C. Tape, and Q. Liu (2005). Seismic tomography, adjoint methods, time reversal, and banana-doughnut kernels. *Geophys. J. Int.*, 160, 195-216, doi: 10.1111/j.1365-246X.2004.02453.x.

Zucca, J.J, W.R. Walter, A.J. Rodgers, P. Richards, M.E. Pasyanos, S.C. Myers, T. Lay, D. Harris, and T. Antoun (2009), The prospect of using three-dimensional Earth models to improve nuclear explosion monitoring and ground motion hazard assessment, *Seism. Res. Lett.*, 80, 31-39, doi:10.1785/gssrl.80.1.31.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Security, LLC, Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.