SANDIA REPORT
SAND2018-9453
Unlimited Release
Printed August 2018

Computation of Kernels for Full Waveform Seismic Inversion Using Parelasti

Leiph A. Preston

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology and Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

SAND2018-9453
Printed August 2018
Unlimited Release

Computation of Kernels for Full Waveform Seismic Inversion Using Parelasti

Leiph A. Preston
Geophysics Department
Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-MS0750

Abstract

Full waveform inversion allows the seismologist to utilize an entire waveform and all the information it contains to help image the 3-D structure of the interior of the earth. This report summarizes the basic theory that has been developed in full waveform seismic inversion, primarily related to computation of sensitivity kernels. It then describes the implementation of this theory using Sandia Geophysics Department's Parelasti code, a 3-D full waveform elastic simulation algorithm. Finally, the code is validated using synthetics from simple homogeneous elastic earth models.

ACKNOWLEDGMENTS

The author would like acknowledge Dave Aldridge for his work in full waveform inversion that allowed this work to come to fruition. The author also acknowledges the National Nuclear Security Administration, Defense Nuclear Nonproliferation Research and Development (DNN R&D), and the Source Physics Experiment (SPE) working group, a multi-institutional and interdisciplinary group of scientists and engineers.

TABLE OF CONTENTS

FIGURES

NOMENCLATURE

Abbreviation	Definition
3-D	Three Dimensional
CPML	Convolutional Perfectly Matched Layers
MPI	Message Passing Interface

1. INTRODUCTION

An understanding of the structure of the earth on a variety of scales forms a basis for many of the scientific endeavors that geophysicists are pursuing. Some research, such as tectonic studies and oil exploration, seeks to use earth structure directly to build tectonic models or decide where to drill. In other cases, such as full waveform forward simulation and moment tensor inversion, earth structure itself may not necessarily be of direct interest, but still critical to achieving desired goals.

Our direct access to the earth is extremely limited due to the expense and many other factors that make it impractical to sample the earth directly. However, seismic waves do sample the earth and recordings of these waves can be used to infer the earth structure through which they passed. Many methods have been developed to utilize seismic waves to glean information about earth structure. One of the newer of these methods is full waveform seismic inversion. As the name implies, the entire seismic waveform, instead of portions or some parameterization of the waveform, is used to build an image of the earth. The primary advantage, of course, is that one is using all the information in the seismogram instead of a small portion as many other methods do. However, the cost is a much greater computational expense compared to more traditional methods, such as first arrival time tomography. Full waveform tomography also suffers from the same difficulties that all other inverse techniques do: non-uniqueness, incomplete sampling, noise, etc. Many tactics are used to mitigate these issues, but it is important to be cognizant of them.

Tarantola (1984) was one of the first authors to develop the theory behind full waveform inversion of seismic data. Since then, multiple authors (e.g, Tarantola, 1988; Fichtner et al., 2006; Plessix, 2006; Tromp et al. 2005; Aldridge, 2009) have further developed the theory and practice of full waveform inversion of seismic data for earth structure and source parameters. This report describes the implementation of Fréchet derivatives, or sensitivity kernels, for an elastic medium using Parelasti, Sandia Geophysics Department's massively parallel finite-difference elastic full waveform simulation software. It will also demonstrate for very simple models that the computed kernels are correct for computing the change in the model that would be necessary to reduce the misfit objective function.

2. THEORY

Much of the theory behind full waveform inversion of seismic data has been detailed in the literature (e.g., Tarantola, 1984; Tromp et al., 2005; Aldridge, 2009). Thus, a derivation will not be set forth here; instead, a brief overview of the theory will be reviewed for constructing the necessary pieces for the sensitivity kernels.

The inverse problem seeks to minimize some misfit objective function. This misfit objective function throughout this document will be defined in a L^2 norm sense. Specifically, we are seeking to minimize the waveform misfit function defined as (Tromp et al., 2005)

$$\chi = \frac{1}{2}\int_0^T \left| s(\mathbf{x_r}, t) - d(\mathbf{x_r}, t) \right|^2 \mathrm{d}t \tag{2.1}$$

where $s(\mathbf{x_r}, t)$ are synthetic displacement waveforms based on a current model of the earth and $d(\mathbf{x_r}, t)$ are observed displacement waveforms, both from receivers at positions $\mathbf{x_r}$ and time t. For multiple sources and receivers, we can simply sum over them to obtain the total waveform misfit function. We would like to find the change in the current model that would reduce the waveform misfit function in Equation 2.1 as much as possible. The quantity $\frac{\partial \chi}{\partial \mathbf{m}}$ describes how the misfit function will change as the model $\mathbf{m}$ changes and is called the Fréchet derivative of the misfit object function.

2.1 Fréchet Derivatives

An elastic earth model, $\mathbf{m}$, contains elastic moduli and densities for every point in the 3-D model domain. Thus, $\mathbf{m}$ can consist of a very large number of parameters. It would be computationally infeasible in most cases to compute $\frac{\partial \chi}{\partial \mathbf{m}}$ directly by varying a medium parameter at each point in the model and assembling the sensitivity vector piecemeal. However, one can use other much more computationally efficient means of computing the Fréchet derivatives. There are multiple equivalent approaches for arriving at the expressions for $\frac{\partial \chi}{\partial \mathbf{m}}$. Plessix (2006), for example, uses adjoint theory as a basis for deriving the Fréchet derivatives. Aldridge (2009) derives equivalent equations using the reciprocity theorem and first Born approximation with signal processing theory throughout without any appeal to adjoint state theory. Others, such as Tromp et al. (2005), employ the representation theorem and first Born approximation to arrive at similarly equivalent solutions.

All methodologies used to arrive at Fréchet derivatives utilize an adjoint field. This field is simply the seismic wave field that is induced by an adjoint source. An adjoint source is a body force-type source, activated at the receiver position, that uses the time-reversed residual (observed minus predicted) waveform as the source time function. It is a "body force-type" source because the source time function does not have units of force per unit volume, but of length (assuming displacement seismograms). However, it is activated just like a body force source in the algorithm

(i.e, $f_i(\mathbf{x}, t)$ in Equation 2.6). With multiple receivers these adjoint sources are activated simultaneously and, thus, their fields superimpose to produce the total adjoint field. It is also possible to use other source types (i.e., moment-type sources) for an adjoint source; however, this report uses force sources exclusively. See Aldridge (2009) for more information on using other source types.

This document will closely follow Tromp et al. (2005) since the subject of their paper includes fully anisotropic sensitivity kernels, but will transform these equations into forms similar to those in Aldridge (2009) since these are more readily implemented with Parelasti.

From Tromp et al. (2005), the sensitivity kernels for density, shear modulus, and bulk modulus for an isotropic elastic media are (Equations 14, 17, and 18 in their paper):

$$K_\rho^T(\mathbf{x}) = -\int_0^T \rho(\mathbf{x})\tilde{\mathbf{s}}(\mathbf{x}, T-t)\frac{\partial^2 \mathbf{s}(\mathbf{x}, t)}{\partial t^2} dt \tag{2.2}$$

$$K_\mu^T(\mathbf{x}) = -\int_0^T 2\mu(\mathbf{x})\tilde{\mathbf{D}}(\mathbf{x}, T-t) : \mathbf{D}(\mathbf{x}, t) dt \tag{2.3}$$

$$K_\kappa^T(\mathbf{x}) = -\int_0^T \kappa(\mathbf{x})\Big[\nabla \cdot \tilde{\mathbf{s}}(\mathbf{x}, T-t)\Big]\Big[\nabla \cdot \mathbf{s}(\mathbf{x}, t)\Big] dt \tag{2.4}$$

where the tilde indicates an adjoint field and $\mathbf{D}$ is the traceless strain deviator (strain tensor after removal of the trace).

These kernels are combined to form the gradient of the misfit function via

$$\partial\chi = \int_V \Big[K_\rho^T(\mathbf{x})\partial \ln\rho(\mathbf{x}) + K_\mu^T(\mathbf{x})\partial \ln\mu(\mathbf{x}) + K_\kappa^T(\mathbf{x})\partial \ln\kappa(\mathbf{x})\Big] dV \tag{2.5}$$

This function relates fractional changes in medium parameters to the change of the misfit objective function.

One can also show through simple manipulations that the change in the misfit objective function is related to the full change in medium parameters by

$$\partial\chi = \int_V \Big[K_\rho(\mathbf{x})\partial\rho(\mathbf{x}) + K_\mu(\mathbf{x})\partial\mu(\mathbf{x}) + K_\kappa(\mathbf{x})\partial\kappa(\mathbf{x})\Big] dV \tag{2.6}$$

for the revised kernels

$$K_\rho(\mathbf{x}) = -\int_0^T \tilde{\mathbf{s}}(\mathbf{x}, T-t)\frac{\partial^2 \mathbf{s}(\mathbf{x},t)}{\partial t^2}dt \tag{2.7}$$

$$K_\mu(\mathbf{x}) = -\int_0^T 2\tilde{\mathbf{D}}(\mathbf{x}, T-t) : \mathbf{D}(\mathbf{x},t)dt \tag{2.8}$$

$$K_\kappa(\mathbf{x}) = -\int_0^T \left[\nabla \cdot \tilde{\mathbf{s}}(\mathbf{x}, T-t)\right]\left[\nabla \cdot \mathbf{s}(\mathbf{x},t)\right]dt \tag{2.9}$$

2.2 Velocity-Stress Sensitivity Kernels

Parelasti solves the velocity-stress system of coupled first-order partial differential equations for an isotropic elastic media:

$$\frac{\partial v_i(\mathbf{x},t)}{\partial t} = \frac{1}{\rho(\mathbf{x})}\left(\frac{\partial \sigma_{ij}(\mathbf{x},t)}{\partial x_j} + f_i(\mathbf{x},t)\right) \tag{2.10}$$

$$\frac{\partial \sigma_{ij}(\mathbf{x},t)}{\partial t} = C_{ijkl}(\mathbf{x})\frac{\partial v_k(\mathbf{x},t)}{\partial x_l} + \frac{\partial m_{ij}(\mathbf{x},t)}{\partial t} \tag{2.11}$$

where v_i is the i^{th} component of particle velocity, σ_{ij} is a component of the 3x3 symmetric stress tensor, ρ is density, f_i is a body force source term, C_{ijlk} is the elastic modulus tensor, m_{ij} is a symmetric moment tensor source, $\mathbf{x}$ is a point in (assumed) 3-D space, and t is time (repeated indices imply summation). C_{ijlk} is symmetric with respect to exchanges in $i \leftrightarrow j$; $k \leftrightarrow l$; and $i, j \leftrightarrow k, l$, meaning there are only 21 independent moduli in a fully anisotropic medium (e.g., Aki and Richards, 2002). For an isotropic elastic medium, $C_{xxxx} = C_{yyyy} = C_{zzzz} = \lambda + 2\mu$, $C_{xxyy} = C_{xxzz} = C_{yyzz} = \lambda$, and $C_{xyxy} = C_{xzxz} = C_{yzyz} = \mu$, and all other non-symmetric C_{ijkl} components are zero. λ is the Lamé parameter and is related to the bulk modulus, κ, via $\lambda = \kappa - \frac{2}{3}\mu$.

Equations 2.2-2.4 contain displacements and accelerations, whereas Parelasti solves for velocities. Instead of integrating (differentiating) velocities to displacements (accelerations) in Parelasti, one can find the sensitivity equations that use velocities directly. Aldridge (2009) provides the sensitivity kernel for $K_\rho(\mathbf{x})$ using exclusively velocities, which can be easily derived from Equation 2.7 using convolution theorems. Equations 2.8 and 2.9 are both related to the strain tensor. To see relation of Equation 2.9 to the strain tensor, first recall the strain tensor components are defined as

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial s_i}{\partial x_j} + \frac{\partial s_j}{\partial x_i}\right) \tag{2.12}$$

Thus, $\nabla \cdot \mathbf{s}(\mathbf{x}, t)$ is just the trace of the strain tensor ($\mathrm{Tr}(\epsilon) = \epsilon_{ii}$). Using this information, one can see how one derives the $K_\kappa(\mathbf{x})$ equation in Aldridge (2009) (Equation 5.5b therein) from Equation 2.9. The reformulated kernels from Equations 2.7-2.9 are now:

$$K_\rho(\mathbf{x}) = -\int_0^T \tilde{\mathbf{v}}(\mathbf{x}, T-t)\mathbf{v}(\mathbf{x}, t)dt \tag{2.13}$$

$$K_\mu(\mathbf{x}) = -\int_0^T 2\tilde{\mathbf{D}}(\mathbf{x}, T-t) : \mathbf{D}(\mathbf{x}, t)dt \tag{2.14}$$

$$K_\kappa(\mathbf{x}) = -\int_0^T \tilde{\epsilon}_{ii}(\mathbf{x}, T-t)\epsilon_{jj}(\mathbf{x}, t)dt \tag{2.15}$$

These equations close to what is needed for Parelasti. However, Equations 2.14-2.15 use strains, which are not directly calculated in Parelasti. Utilizing Equation 2.11, one can derive equations for the traceless strain deviator and the trace of the strain tensor needed for Equations 2.14 and 2.15:

$$D_{ij} = \frac{\sigma_{ij}}{2\mu}, \quad i \neq j \tag{2.16}$$

$$D_{xx} = \frac{1}{\mu}\left[\frac{1}{3}\sigma_{xx} - \frac{1}{6}\left(\sigma_{yy} + \sigma_{zz}\right)\right] \tag{2.17}$$

$$D_{yy} = \frac{1}{\mu}\left[\frac{1}{3}\sigma_{yy} - \frac{1}{6}\left(\sigma_{xx} + \sigma_{zz}\right)\right] \tag{2.18}$$

$$D_{zz} = \frac{1}{\mu}\left[\frac{1}{3}\sigma_{zz} - \frac{1}{6}\left(\sigma_{yy} + \sigma_{xx}\right)\right] \tag{2.19}$$

$$\epsilon_{ii} = \frac{1}{2\mu + 3\lambda}\left[\sigma_{xx} + \sigma_{yy} + \sigma_{zz}\right] \tag{2.20}$$

2.3 Alternative Parameterizations

Equation 2.6 gives the change in the misfit object function when the medium parameters ρ, μ, and κ change. However, other parameterizations are possible, including α (P-wave speed), β (S-wave speed), and λ, among others (see Aldridge (2009)). The expressions for kernels based on these parameters are derived via change of variable techniques, since these are all interrelated.

2.3.1 Density, P-, and S-wave Speed Parameterization

$$\partial\chi = \int_V \left[K_\rho^w(\mathbf{x})\partial\rho(\mathbf{x}) + K_\beta^w(\mathbf{x})\partial\beta(\mathbf{x}) + K_\alpha^w(\mathbf{x})\partial\alpha(\mathbf{x}) \right] dV \tag{2.21}$$

$$K_\rho^w = K_\rho + \beta^2 K_\mu + \left(\alpha^2 - \frac{4}{3}\beta^2 \right) K_\kappa \tag{2.22}$$

$$K_\beta^w = 2\beta\rho K_\mu - \frac{8}{3}\beta\rho K_\kappa \tag{2.23}$$

$$K_\alpha^w = 2\alpha\rho K_\kappa \tag{2.24}$$

These are obviously non-linear with respect to the medium parameters.

2.3.2 Density, Shear Modulus, and Lamé Parameterization

$$\partial\chi = \int_V \left[K_\rho^\lambda(\mathbf{x})\partial\rho(\mathbf{x}) + K_\mu^\lambda(\mathbf{x})\partial\mu(\mathbf{x}) + K_\lambda^\lambda(\mathbf{x})\partial\lambda(\mathbf{x}) \right] dV \tag{2.25}$$

$$K_\rho^\lambda = K_\rho \tag{2.26}$$

$$K_\mu^\lambda = K_\mu + \frac{2}{3}K_\kappa \tag{2.27}$$

$$K_\lambda^\lambda = K_\kappa \tag{2.28}$$

The kernels are the same as for the ρ, μ, κ parameterization except for Equation 2.27.

3. IMPLEMENTATION

Section 2 outlined the basic theory and tools needed to perform full waveform seismic inversion using Parelasti. Equations 2.13-2.20 in combination with Equation 2.6 provide the means to compute the gradient of the waveform misfit function. From Equations 2.13-2.15, one needs both the adjoint field and forward field simultaneously to calculate the kernels. Of course, to produce the residual waveforms needed for the adjoint source and field, one must have already calculated the forward seismic field. Thus, a brute force approach might require storage of the full 3-D wave field for all simulation times, which could demand a vast amount of disk space even by today's standards depending on the model. An approach proposed in Tromp et al. (2005) greatly reduces the computational storage requirements at the cost of increased execution burden. Basically, one first computes the original forward simulation, while storing only boundary information (and receiver traces). One then can run the adjoint simulation using the residual waveforms, while simultaneously running the original simulation backwards in time from the stored boundary data. Although Equations 2.13-2.15 have time-reversed adjoint fields, not forward fields, it is easy through a change of variable to show that the equations will yield the same result if the adjoint field is used forward in time and the original field is computed backwards in time.

3.1 Storing Boundary Information

The forward field is first computed in a standard way with true sources and receivers with the current best-estimate earth model. While it is running, however, we can collect boundary information so that we can replay the same calculation backwards in time when we are computing the kernels. From a theoretical perspective, to run the simulation backwards one needs the wave field variables on the boundaries of the model for all times, plus a complete 3-D snapshot of all wave field variables at the end time of the simulation. This information provides the initial and boundary conditions for the reverse-time partial differential equations. From a practical perspective, the discrete nature of the underlying standard staggered finite-difference grid and numerical absorbing boundary conditions used in Parelasti must be considered.

Parleasti uses a standard staggered finite-difference grid (Virieux, 1986). In this scheme compressional stress nodes exist on integer node points, velocity component nodes are staggered one-half grid node in their respective directions from compressional nodes on the edges of a cell, and shear stress nodes are centered on faces of the cell, one-half grid node in two directions from compressional nodes (Figure 3.1). Material parameters are colocated with the compressional nodes and interpolated as needed to half-integer node points. Besides being spatially staggered, the grid is also temporally staggered, so that stresses are updated on integer time steps and velocities are updated on half-integer time steps. This allows Parelasti to use centered finite-difference operators, which are 4^{th} and 2^{nd} order accurate in space and time, respectively. This means that only one previous time step is needed for time stepping, but one and a half grid nodes are needed for spatial derivatives. This can be visualized in Figure 3.2. To compute the values of the velocities and shear stresses at position D in Parelasti, the compressive stress and vertical velocities at positions A and

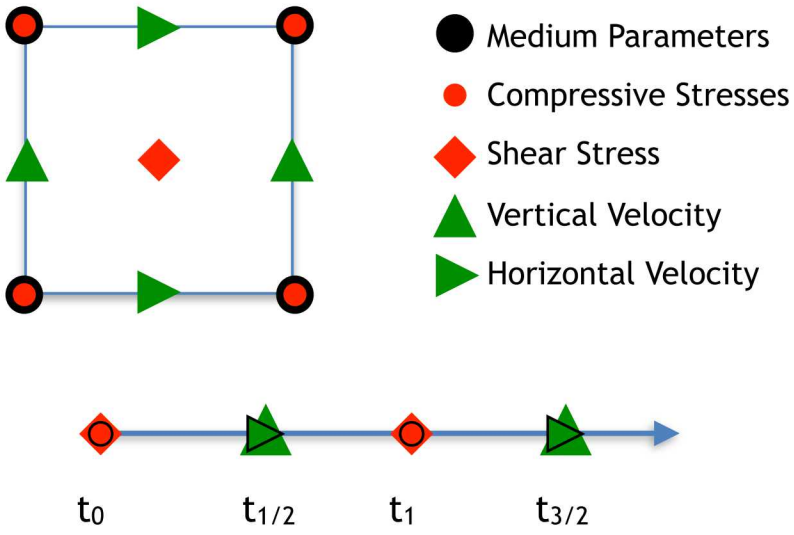

Figure 3.1: Unit cell (top) and time axis (bottom) for the staggered finite-difference scheme.

C from the boundary zone must be provided. Once the dependent variables at position D have been computed, then the compressive stresses and vertical velocities at position E can be calculated from data in position B from the boundary zone and the previously computed values at D. Thus, three "layers" of variables, with a total thickness of one and a half grid nodes, must be stored for each of the six sides of the model volume at each time step. Edges (where two sides meet) and corners (where three sides meet) do not need stored since all derivatives align with the coordinate axes (no off-axis information is needed).

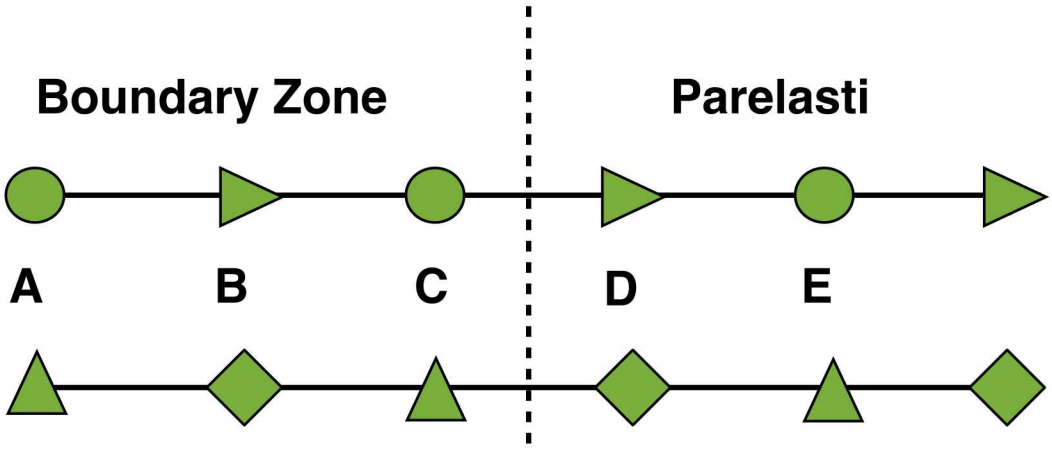

Figure 3.2: Required spatial distribution of velocities (triangles) and stresses (circles, diamonds) needed by Parelasti for reverse-time boundary conditions.

The other consideration is that of absorbing boundary conditions. Parelasti uses Convolutional Perfectly Matched Layers (CPML) (Komatitsch and Martin, 2007) to mitigate unphysical numerical domain boundary reflections. These absorbing boundary conditions work well at minimizing these undesirable numerical artifacts. From the standpoint of performing a reverse-time simulation, the only consideration is that they do their job well in the forward simulation. We capture the boundary field of the forward simulation just before the beginning of the CPML zones on all sides, while the field is still in "normal" space. This means that absolutely no CPMLs are needed in the reverse time simulation, saving some computational effort, and removes the need to undo the effects of the CPMLs, which could lead to numerical issues.

3.2 Adjoint Calculation

Once the forward simulation is complete, the adjoint source(s) can be computed. As outlined in Section 2, the residual wave forms are first calculated by subtracting the synthetic (forward) simulation traces from the observed traces for all of the components at all receivers. Section 2 provides the theory based on displacement waveforms for both observed and synthetic traces. In many instances this will mean the observed traces must be integrated from velocity traces to displacement, which can have numerical issues if not handled carefully for certain datasets. For synthetics, this is not (usually) an issue, since Parelasti computes the integration internally with appropriate calling flags. Additionally, the observed waveforms should be low-pass filtered so that they have approximately the same frequency content as the synthetics. This will ensure that the adjoint calculation is accurate for the given model discretization. Using an adjoint source with too high of frequency content will lead these higher frequencies to be erroneous due to numerical dispersion issues (e.g., Haney and Aldridge, 2008).

These residual waveforms are then time reversed to form the adjoint source time function. These are input as simultaneous force sources at the true receiver locations. The component of the force source will be the same as the component of the receiver from which the residual waveform was derived. Once the adjoint sources have been determined, a forward run of Parelasti can commence. As far as the underlying equations and CPMLs are concerned, this is a standard forward simulation of Parelasti. The differences arise in combination with the reverse-time simulation of the original forward problem of Section 3.1.

3.3 Computing the Kernels

Simultaneously with the adjoint calculation, the forward simulation in Section 3.1 is replayed backwards in time. At start up, the last time step of the original forward problem is read back to populate all dependent variables, forming the initial conditions. Then at each subsequent time step, only the boundary zones for that time step are read in to replace the dependent variables that correspond to those points in the original forward simulation. Recall from Section 3.2 that CPML conditions are not needed in the reverse-time simulation because the full wave field is defined by the initial and boundary conditions supplied by the input data. In addition, the time-reversed (to match the time-reversed simulation) and negative source time function

from the original forward simulation must be utilized. The only other difference in the reverse-time simulation compared to a standard forward calculation is that the time step is negative that of the original simulation. When run alone in reverse-time mode, the simulation looks just like the original wave field played in reverse.

The adjoint and time-reversed original calculations are run in sync, such that the first time step in the adjoint calculation is run at the same time as the last step of the original simulation. The second time step corresponds to the penultimate time step of the original simulation and continuing until the final time step of the adjoint simulation corresponds to the first time step of the original simulation. The kernels are built up over the simulation time, via the integrals over time given in Equations 2.13-2.15 from information available at each time step. Each kernel is a 3-D volume with no time component; thus, each kernel will be $n_x n_y n_z$ in size, which can become large for certain models.

3.4 Parallelization and Data Management

Parelasti is parallelized using the message passing interface (MPI). Thus, the steps above can be computed in a massively parallel framework on thousands of cores, if available, down to a single work station. The forward model simulation (Section 3.1) when saving boundary conditions does require more computational effort than a standard run (without saving boundary conditions). It also requires significant disk space: a full 3-D volume for each of the 9 dependent variables at the final time step and boundary zones for all six sides at all time steps. The boundary zone requires 12 dependent variables on each side since one and a half grid nodes worth of information is needed. Thus, the 3-D volume will require storage of $9 \times n_x n_y n_z$ values and the boundary zones will need roughly $2 \times 12 \times \left(n_x n_y + n_x n_z + n_y n_z \right) \times n_t$ values stored, where n_t is the number of time steps. Although some compression of this could be accomplished with future work, currently all the above is written to disk.

All of this information is written to netcdf (https://www.unidata.ucar.edu/software/netcdf/) files, which store the data along with associated metadata in binary files. These netcdf files are read during the adjoint phase in order to replay the original simulation in reverse. Additionally, the trace data from the original forward simulation, needed to make the adjoint sources, are written to netcdf files. The kernels output from the adjoint phase are also written to netcdf files. Most modern computer languages and programs have netcdf interfaces that allow access to the data on a wide variety of platforms.

4. BASIC VALIDATION

The kernels computed via Equations 2.13-2.15 in combination with Equation 2.6 give the sensitivity in changes of the waveform misfit objective function to variations in the medium parameters. This section provides a basic validation of those equations as implemented in Parelasti for very simple cases to confirm the mechanics of the kernel calculation are correct.

For these validation tests, a homogeneous whole space with medium properties V_p = 2500 m/s, V_s = 1500 m/s, and density = 2000 kg/m^3 (κ = 6.5e9 Pa, μ = 4.5e9 Pa) was chosen for the best-estimate (synthetic) earth model. A vertical force source located at the center of the 100 m by 100 m by 100 m model with a 50 Hz Ricker source time function and recorded at a single vertical displacement receiver situated 10 m vertically above the source. This model remains unchanged throughout the tests; thus, the synthetic trace, boundary conditions, and final time wave field files are created once and remain unchanged (Section 3.1). Only the "observed" data from the "true" earth will be varied to test the kernels.

Equation 2.6 describes the relationship among changes in medium parameters to changes in the misfit object function. It is important to recognize that the kernels, which quantify the change in the misfit objective function for a change in one of the medium parameters, is itself dependent on the underlying model. Thus, Equation 2.6 is a nonlinear equation of the medium parameters. One method of solving Equation 2.6 to minimize the misfit objective function is to perform an iterative inversion scheme. In this scheme, the problem is solved like it is a linear problem, a step is taken in the implied direction, and a new model is created. This new model then is used to create new kernels that are subsequently used to solve the equations again in a linear fashion. This process is repeated until some convergence criteria is satisfied. In the following examples (except one) we will perform only one linear step to observed how closely one single step approaches the solution. Of course, all simulations contain numerical error. Numerical error can also slow convergence by not accurately replicating exact solutions.

4.1 Density Variation Only

Equations 2.6 indicates that if the density is the only parameter to change, then only the density kernel will have any bearing on the outcome. Additionally, if the density perturbation is constant in space (a change in the whole space), then,

$$\partial\rho = \frac{\partial\chi}{\int K_\rho \mathrm{d}V} \tag{4.1}$$

The density for this "true" earth model is set to 2100 kg/m^3. In order to keep the bulk and shear moduli fixed relative to the synthetic model, V_p and V_s must be adjusted accordingly to 2440 m/s and 1464 m/s, respectively. A standard forward run of Parelasti is performed for the source and receiver identical to that of the synthetic model.

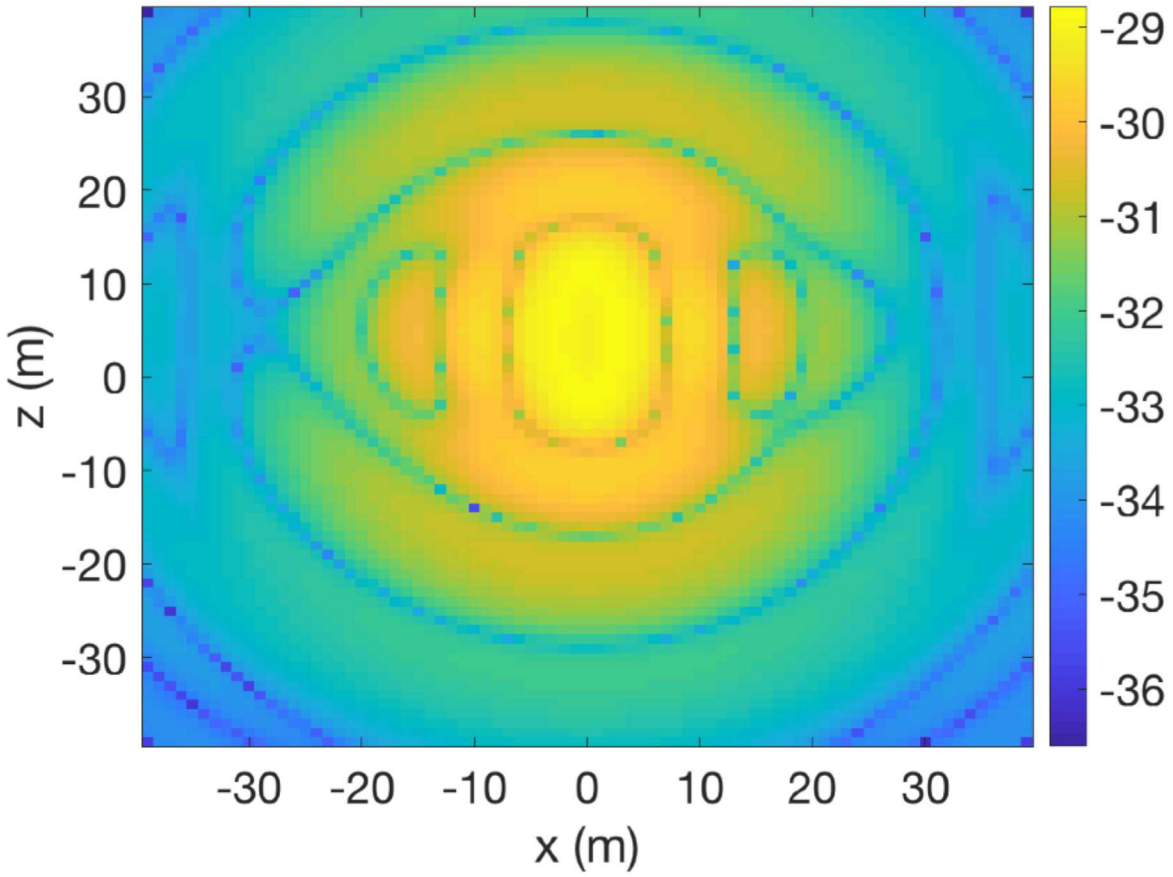

Figure 4.1: Log_{10} of the absolute value of the 1-D integrated density kernel. Source is located at (0,0), receiver at (0,10).

The output from this model is the "observed" data. The residual waveform is computed, time reversed and written to a netcdf file as a vertical force source time function as the adjoint source (Section 3.2). Finally, an adjoint simulation is made simultaneously with the time-reversed synthetic model to find the density kernel.

According to Equation 4.1, the misfit objective function (Equation 2.1) divided by the volume integral of the density kernel will give the change in density needed by the synthetic model to match the observed data. The result of performing this operation gives $\partial\rho$ = +99 kg/m^3, approximately a 1% error. Thus, a single assumed linear step is able to get us 99% of the way to the true solution.

Figure 4.1 shows the density kernel, integrated along the y dimension. The $\log_{10}$ of the absolute value of this operation is made to show the structure. The kernel peaks at the source and receiver locations and rapidly decreases in magnitude away from these locations except near the ray path connecting source and receiver. The blue lines are zero crossings of the kernel.

4.2 Bulk Modulus Variation Only

In this case, density and shear modulus are fixed; thus, only the bulk modulus kernel will have an effect. As before, if the change in bulk modulus is independent of space, then

$$\partial\kappa = \frac{\partial\chi}{\int K_\kappa \mathrm{d}V} \tag{4.2}$$

The bulk modulus is changed to approximately 6.25e9 Pa ($\Delta\kappa$ = -248,750,000 Pa), which corresponds to V_p = 2475 m/s. V_s is unaffected and remains at 1500 m/s and density at 2000 kg/m^3. Again, a standard forward run is made with Parelasti, from which the adjoint source is computed.

Computing Equation 4.2 using the resulting kernel yields $\partial\kappa$ = -256,428,112 Pa, which is an error of approximately 3%. A single assumed linear step achieves 97% of the total distance toward the true solution, not as far as for density variations, but still very nearly linear.

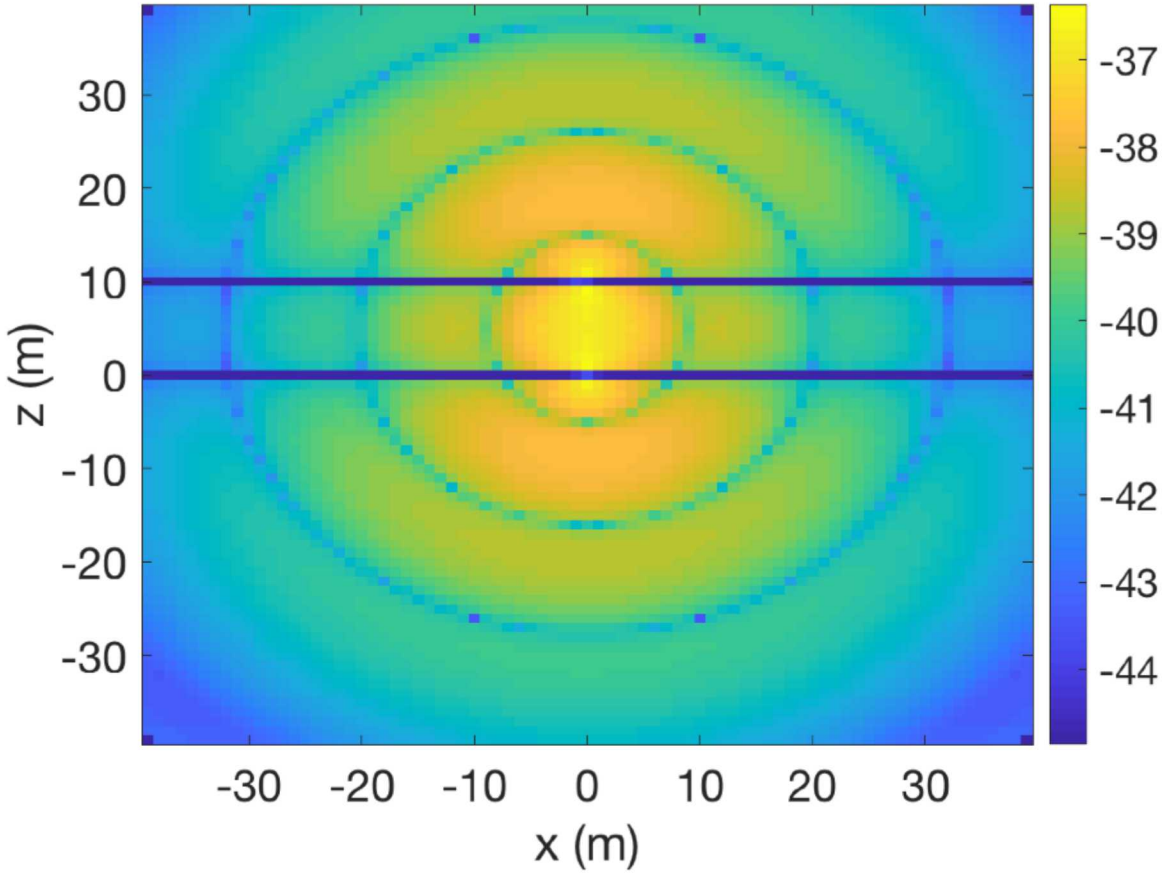

Figure 4.2: Log$_{10}$ of the absolute value of the 1-D integrated bulk modulus kernel. Source is located at (0,0), receiver at (0,10).

The 1-D y-integrated bulk modulus kernel (Figure 4.2) also peaks near the source and receiver locations, but has nodal lines of zero sensitivity at the z-positions of the source and receiver for this source-receiver geometry, given that only the vertical component of displacement is utilized. Otherwise the kernel exhibits a circular pattern of kernel zero crossings surrounding the source-receiver locations. It has lower relative amplitude along the ray path connecting the source and receiver compared to the density kernel.

For the this case only, a second iteration of the process was completed to demonstrate that it does converge to the true answer. In this case, the computed $\partial\kappa_1$ = -256,428,112 Pa from above was added to the starting synthetic model κ and an updated synthetic model was constructed and simulated. The residual trace from this model was used to form the adjoint source and perform the adjoint calculation to compute a new $\partial\kappa_2$ using the same procedure as before. This yielded $\partial\kappa_2$ = +7,710,590 Pa, which when combined with the $\partial\kappa$ from the first iteration gives a total $\partial\kappa_{tot} = \partial\kappa_1 + \partial\kappa_2$ = -248,717,522 Pa, a 0.013% error relative to true model. Thus, further iterations do converge toward the true value and, in this case, quite rapidly.

4.3 Shear Modulus Variation Only

For this "true" model, the shear modulus is altered, but the bulk modulus and density are set at the synthetic model values. Analogously to Sections 4.1 and 4.2, if the change in shear modulus is independent of space then

$$\partial\mu = \frac{\partial\chi}{\int K_\mu \mathrm{d}V} \tag{4.3}$$

The change in shear modulus, besides the obvious change in V_s, will also affect V_p. The shear modulus is changed to 4.455e9 Pa ($\Delta\mu$ = -45,000,000 Pa). This corresponds to Vp = 2494 m/s and Vs = 1492 m/s. Density is unchanged. The adjoint source is computed from the trace produced from this model.

Using Equation 4.3, for the above case the calculation yields $\partial\mu$ = -45,713,908 Pa, which is about 2.5% error. Similar to bulk modulus variations, a change in shear modulus is very nearly linear, with a single linear step providing 97.5% of the change toward the true solution.

The 1-D integrated shear modulus kernel shown in Figure 4.3 also demonstrates peak values near the source and receiver locations, with lower magnitudes in a quasi-circular region connecting source and receiver (distorted by numerical noise in the image). The ray path between source and receiver has higher relative magnitude similar to the density kernel.

4.4 Vp Variation Only

From Equations 2.21-2.24, one can also allow α to vary while keeping density and β fixed. Solving for $\partial\alpha$, and assuming it is not a function of space, while setting $\partial\beta$ and $\partial\rho$ to zero, one obtains

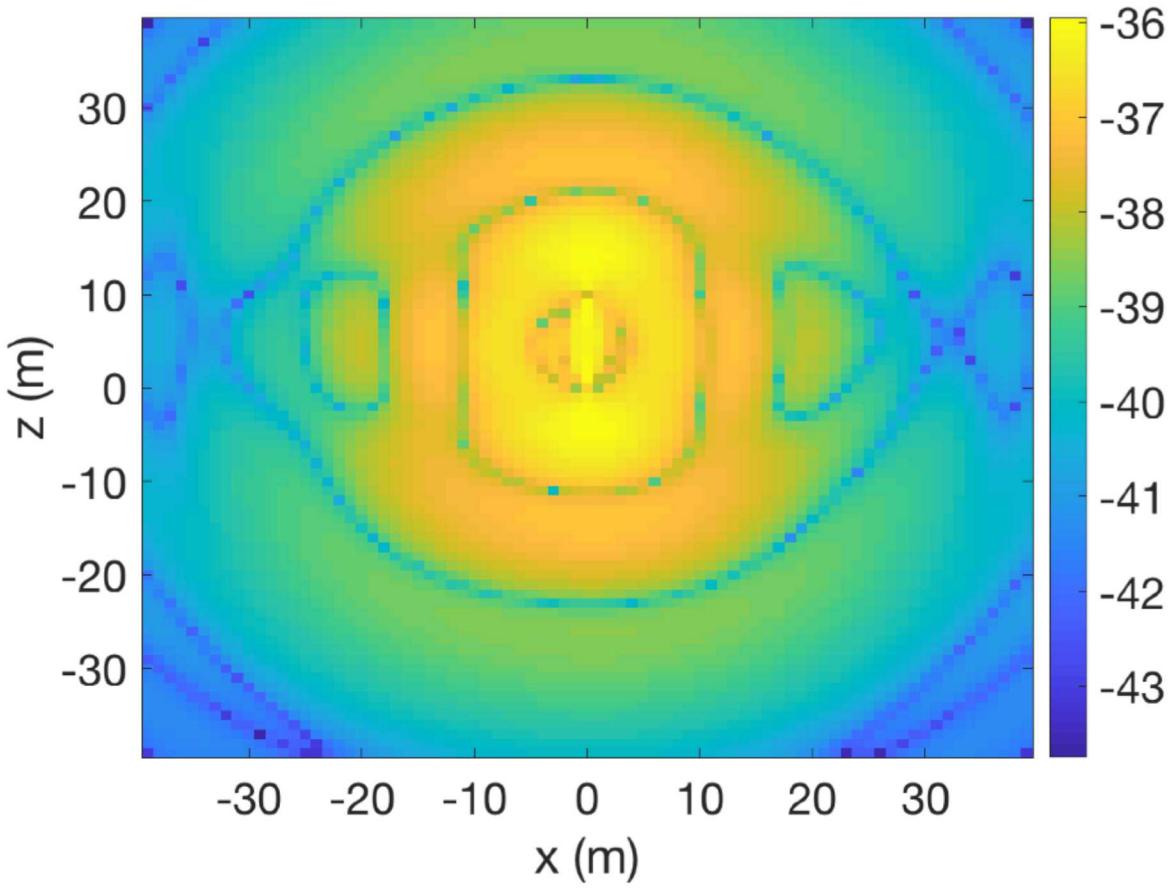

Figure 4.3: Log_{10} of the absolute value of the 1-D integrated shear modulus kernel. Source is located at (0,0), receiver at (0,10).

$$\partial\alpha = \frac{\partial\chi}{2\alpha\rho\int K_\kappa \mathrm{d}V} \tag{4.4}$$

The P-wave speed is changed to 2475 m/s ($\Delta\alpha$=-25 m/s), while the shear wave speed and density remain unchanged from the synthetic case. Equation 4.4 is clearly non-linear with respect to P-wave speed in addition to the nonlinearity of K_κ. Solving Equation 4.4 for the assumed linear step gives $\partial\alpha$=-25.6 m/s, which is a 2.6% error. This is comparable to the $\partial\kappa$ and $\partial\mu$ errors, but still indicates very nearly linear behavior.

Figure 4.4 shows the 1-D y-integrated P-wave speed kernel. It is simply a scaled version of the bulk modulus kernel shown in Figure 4.2.

4.5 Vs Variation Only

Finally, the shear wave speed will be varied while the P-wave speed and density remain fixed. These are again described by Equations 2.21-2.24. Solving for a spatially independent $\partial\beta$, while setting $\partial\rho$ and $\partial\alpha$ to zero yields

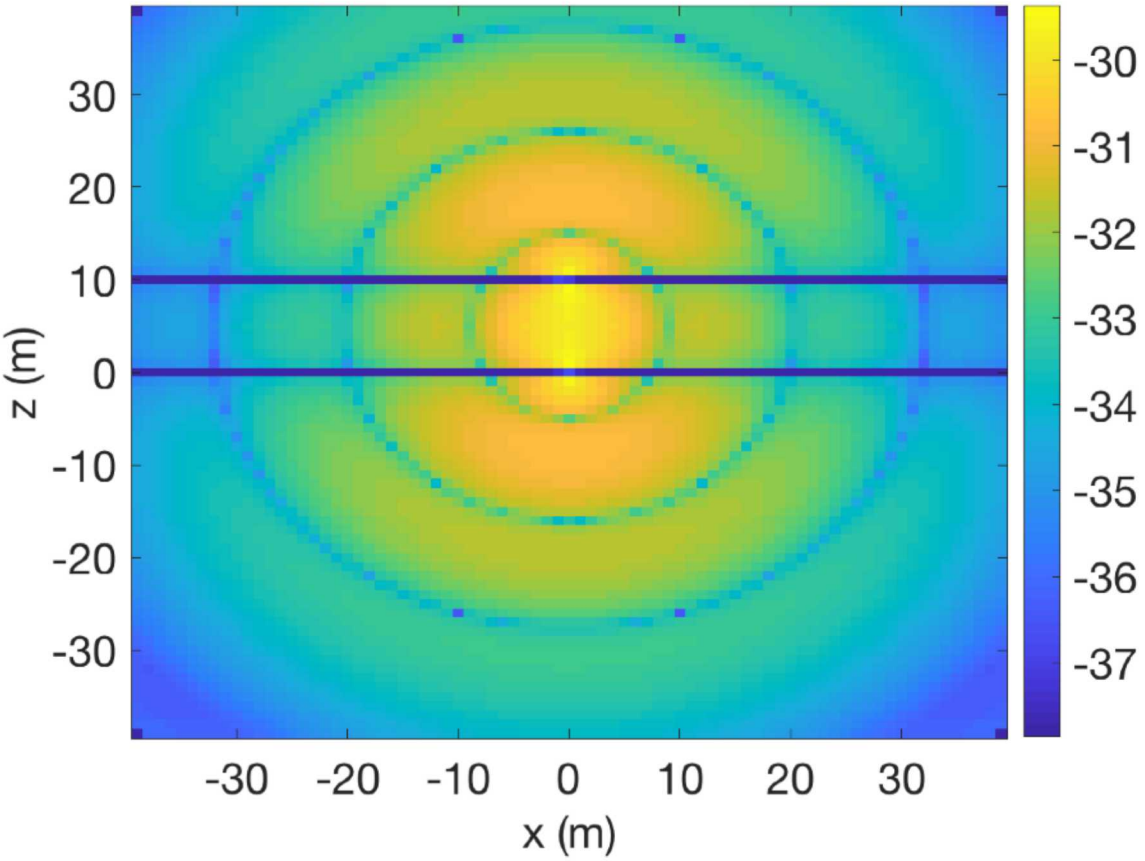

Figure 4.4: Log_{10} of the absolute value of the 1-D integrated P-wave speed kernel. Source is located at (0,0), receiver at (0,10).

$$\partial\beta = \frac{\partial\chi}{2\beta\rho\int K_\mu \mathrm{d}V - \frac{8}{3}\beta\rho\int K_\kappa \mathrm{d}V} \tag{4.5}$$

The shear wave speed in this test is changed to 1492.5 m/s ($\Delta\beta$=-7.5 m/s) while the P-wave speed and density remain at the same values as the synthetic model. Equation 4.5 depends on two basic kernels and is also dependent on the shear wave speed itself, so again clearly nonlinear. Solving Equation 4.5 gives $\partial\beta$=-8.4 m/s, a 11% error when assuming a single linear step. Given the structure of Equation 4.5, it is not overly surprising that this is the most nonlinear of the tested variations.

The 1-D integrated shear wave speed kernel (Figure 4.5) is a scaled combination of the shear modulus kernel and bulk modulus kernel. Thus, it shows properties of both kernels, but is closer in similar to the shear modulus kernel. As with all the other kernels, the peak values are near the source and receiver points. There are also high relative magnitudes for the kernel near the ray path between source and receiver. Similar to the shear modulus kernel, a quasi-circular region of lower magnitude connects source and receiver.

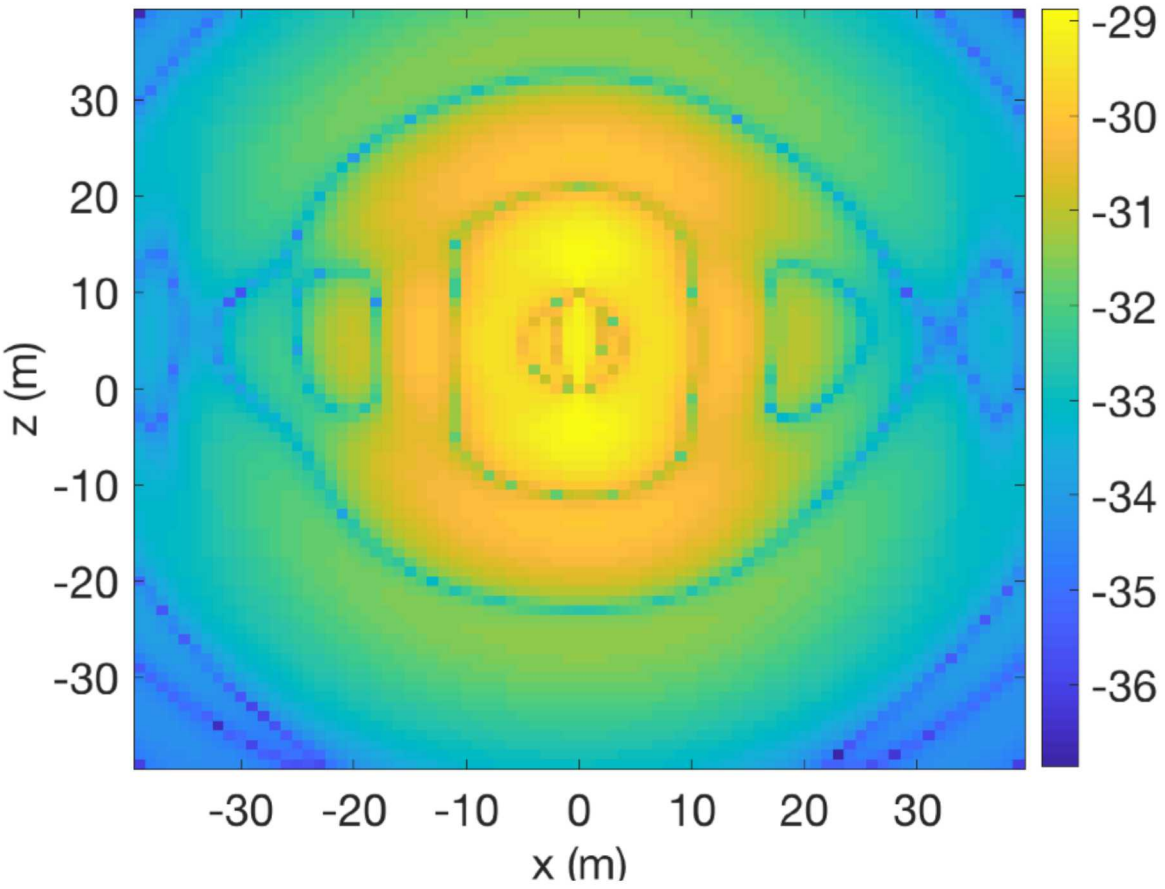

Figure 4.5: Log_{10} of the absolute value of the 1-D integrated shear wave speed kernel. Source is located at (0,0), receiver at (0,10).

5. SUMMARY

This report has described the implementation and basic validation of computing elastic kernels for full waveform seismic inversion. The basis of the theory was elaborated in Section 2 with Section 3 providing the implementation in Parelasti. Section 4 demonstrates the validity of the implementation for simple earth models. The Appendix gives the flags needed in Parleasti for both the forward and adjoint portions of the problem.

Although the validation exercises in Section 4 are simple, they do demonstrate that computed homogeneous changes in the model can nearly replicate the true solution in one step, despite the underlying nonlinearity of the kernels. For the bulk modulus kernel, the rapid convergence of the process toward the true solution was also demonstrated. The K_ρ, K_κ, and K_μ kernels (Equations 2.13-2.15) provide the simplest form of the equations investigated. They are also nearly linear, achieving 97-99% of the full solution step in one linear step. Kernels of the wave speeds are more nonlinear with the K_β kernel showing the most nonlinearity, but still achieving 89% of the full solution step in one linear step.

This work is just the first step in a true full waveform inversion capability. The kernels form the basis upon which a full inversion package can be built. The next steps will involve deciding how to mitigate the large null space inherent in many inverse problems; which algorithm should be used to perform the inversion; and a framework for performing the iterative inversion from start to finish with specified convergence criteria.

REFERENCES

1. Aki, K., and P.G. Richards, *Quantitative Seismology, Second Edition*, University Science Books, Sausalito, CA, 2002.
2. Aldridge, D.F., *Fréchet Derivatives for the Full Waveform Seismic Inverse Problem: Ideal Fluid Medium*, SAND2009-2952, Sandia National Laboratories, Albuquerque, NM, May 2009.
3. Fichtner, A., H.-P. Bunge, and H. Igel, The Adjoint Method in Seismology I: Theory, *Phys. Earth Plan. Int.*, **157**, 86-104, 2006.
4. Haney, M.M., and D.F. Aldridge, *Numerical Dispersion for the Conventional-Staggered-Grid Finite-Difference Elastic Wave Propagation Algorithm*, SAND2008-4991, Sandia National Laboratories, Albuquerque, NM, July 2008.
5. Komatitsch, D., and R. Martin, An Unsplit Convolutional Perfectly Matched Layer Improved at Grazing Incidence for the Seismic Wave Equation, *Geophysics*, **72**, P.SM155-SM167, doi:10.1190/1.2757586, 2007.
6. Plessix, R.-E., A Review of the Adjoint-State Method for Computing the Gradient of a Functional with Geophysical Applications, *Geophys. J. Int.*, **167**, 495-503, 2006.
7. Tarantola, A., 1984, Linearized Inversion of Seismic Reflection Data, *Geophys. Prosp.*, **32**, 998-1015, 1984.
8. Tarantola, A., 1988, Theoretical Background for the Inversion of Seismic Waveforms, Including Elasticity and Attenuation, *Pure and Appl. Geophys.*, **128**, 335-399, 1988.
9. Tromp, J., C. Tape, and Q. Liu, Seismic Tomography, Adjoint Methods, Time Reversal, and Banana-Doughnut Kernels, *Geophys. J. Int.*, **160**, 195-216, 2005.
10. Virieux, J., P-SV Wave Propagation in Heterogeneous Media: Velocity-Stress Finite-Difference Method, *Geophysics*, **51**, 889-901, 1986.

APPENDIX A: PARELASTI USAGE

Computing the kernels necessary for the gradient of the misfit objective function requires several steps (outlined in Section 3) and some additional flags be given to Parelasti when running simulations. In general, Parelasti can use the same flags as for a standard run; however, some flags may be of limited usage during certain steps in practice. For example, during an adjoint calculation of the kernels, calculation of time slices and receivers is disabled because these would provide unnecessary information except for debugging purposes. This appendix provides the new flags that can be used with Parelasti during the different steps of the process.

A.1 Forward Calculation

The first step in computing kernels involves a standard forward waveform simulation. In this step all flags that are used in a standard forward simulation are available for use. In fact, one sets up the forward part of the simulation just like one would if one were not eventually performing an adjoint simulation, except that one should ensure that displacement traces are created. However, in order that the forward simulation be utilized as part of the adjoint simulation step later, the boundary and final time-step volume information must be stored. To do this, one simply adds the -Ef flag with two arguments:

-Ef *fullWaveformFile.cdf boundaryFileBase*: *fullWaveformFile.cdf* is the name of the netcdf file where you would like the full 3-D waveform from the final time step stored. *boundaryFileBase* is the base file name you would like the boundary information stored. Files will be named *boundaryFileBase*+"*_N*.cdf" where *N* is a number starting at 0 and incrementing as needed to keep file sizes manageable.

No other changes are necessary. Note that the files created can be quite large.

A.2 Adjoint Source

This step does not use Parelasti. Instead it uses the displacement traces created in step A.1 to compute residual waveforms from the observed data. An external program such as Matlab, python, etc., can be used to pull the synthetic traces from the created netcdf files and compute the residual waveform. These same external programs can then be used to create the adjoint source netcdf model file. For example, after time reversing the residual waveforms to form the adjoint source time functions, the Matlab function 'writeSgfdModel.m' can be used to write the adjoint source to a model file with the 'forceSource' optional input parameter. This model will serve as the input 3-D earth model, containing the adjoint source, for step A.3.

A.3 Kernel Calculation

In this step, the original simulation from A.1 is re-computed backwards in time at the same time that the adjoint calculation is performed forward in time with the adjoint source. There are many similarities in a kernel calculation run compared to a standard run, but with a few subtleties. In the first place the adjoint source is expected to be contained within the input model file as a 'forceSource' as described in A.2. The source from the original forward simulation is given on the command line like it was in the original forward simulation of A.1, with an additional flag that time-reverses and negates that original source waveform. Some additional flags are also

given to inform the Parelasti to read in the 3-D full waveform and boundary information from step A.1., compute the kernels, and write the kernels out to file. Here are the additional flags:

-SNn 0: Time reverse and negate the command line source time function. The input source time function and source must match that for the original forward simulation from step A.1. The '0' argument tells Parelasti to shift the input source time function by 0 time steps after time reversing.

-IR *fullWaveformFile.cdf boundaryFileBase*: This reads in the 3-D volume from the final time step created in A.1 from *fullWaveformFile.cdf* and boundary information from the *boundaryFileBase* file(s) and sets up a reverse-time simulation of the original simulation in A.1.

-K *kernName1 … kernNameN*: Compute the kernels given as arguments. Currently, the options are 'kkap', 'kmu', and 'krho'. At least one *kernName* must be provided, but additional ones may also be given.

-Eow *kernOutputFileName.cdf*: The output from the requested kernels will be output to the file *kernOutputFileName.cdf*.

The above flags are all that are necessary for a kernel calculation Parelasti run. The kernels will be output into the file given with the -Eow flag and can be read by any external program that can process netcdf files (e.g., Matlab, python). Making time slices (with the -En flag, for example) and receivers (-R family of flags) will not work in kernel calculation mode.

A.4 Time Reverse Flag

One other flag will simply read in the full 3-D volume and boundary information and perform a reverse simulation alone (no kernel or adjoint simulations are computed). This flag is

-Ir *fullWaveformFile.cdf boundaryFileBase*: This reads in the 3-D volume from the final time step created in A.1 from *fullWaveformFile.cdf* and boundary information from the *boundaryFileBase* file(s) and runs a reverse-time simulation of the original simulation in A.1.

In this case, receivers and slices are allowed, just as in a standard Parelasti simulation. This flag could be used diagnostically or, with further development, as part of a reverse time migration process. It is not used as part of a standard full waveform inversion workflow.

DISTRIBUTION

1	MS0750	Leiph Preston	8861
1	MS0750	Christian Poppeliers	8861
1	MS0750	Robert E. Abbott	8861
1	MS0899	Technical Library	9536 (electronic copy)


Sandia National Laboratories