LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-573912

Source Estimation by Full Wave Form Inversion

Bjorn Sjogreen and N. Anders Petersson

August 15, 2012

Submitted to Journal of Scientific Computing

Disclaimer

Source estimation by full wave form inversion

Björn Sjögreen* and N. Anders Petersson*

August 15, 2012

Abstract

We consider the inverse problem of estimating the parameters describing a seismic source, based on time-dependent ground motion recordings at a number of receiver stations. The source is modeled as a point moment tensor forcing, characterized by its location, moment tensor components, start time, and frequency parameter in the time function. In total, there are 11 unknown parameters. We use a non-linear conjugate gradient algorithm to minimize the full waveform misfit between observed and computed ground motions at the receiver stations.

An important underlying assumption of the minimization problem is that seismic wave propagation can be accurately modeled by the elastic wave equation in a heterogeneous isotropic material. We use a fourth order accurate finite difference method to evolve the seismic waves in time. The discretization satisfies a summation by parts property that guarantees stability of the explicit time-stepping scheme. The adjoint of the discretized elastic wave equation is used to compute the gradient of the misfit, which is needed by the non-linear conjugated gradient minimization algorithm.

A new moment tensor source discretization is derived that is twice continuously differentiable with respect to the source location. It guarantees that the Hessian of the misfit is a continuous function of the source parameters. We show how the Hessian can be calculated by solving 11 elastic wave equations and one adjoint wave equation. Because Hessian of the unscaled problem has a very large condition number, a preconditioner must be used to scale the parameters in the non-linear conjugated gradient algorithm. Compared to several other scaling approaches, we find that the diagonal of the Hessian provides the most reliable alternative. Numerical experiments are presented for estimating the source parameters from synthetic data in a layer over half-space problem (LOH.1), demonstrating good convergence properties of the proposed approach.

1 Introduction

This article presents a computational technique for estimating the parameters specifying the source in a seismic event, such as an earthquake, a mine implosion, or an explosion.

*Center for Applied Scientific Computing, L-422, LLNL, P.O. Box 808, Livermore, CA 94551, USA. This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This is contribution LLNL-JRNL-573912.

Our approach uses a preconditioned non-linear conjugated gradient technique to match a number of time-dependent seismographic ground motion observations, which are measured at fixed receiver locations.

We consider seismic source estimation as a minimization problem constrained by the elastic wave equation subject to appropriate boundary and initial conditions. Our objective of the source estimation is to minimize the difference between the recorded and simulated wave forms. We assume that ground motion observations are recorded at the fixed spatial locations $\mathbf{x}_r$, $r = 1, \ldots, R$, and that three orthogonal components of the displacement are measured as functions of time at all recording stations, denoted by $\mathbf{d}_r(t)$. Let $\mathbf{u}(\mathbf{x}, t)$ be the displacement field governed by the elastic wave equation. The displacement field depends implicitly on the source parameters, which we collect in the P-dimensional real-valued vector $\mathbf{p}$. The continuous minimization problem is defined through the misfit functional

$$\mathcal{X}_c(\mathbf{p}) = \frac{1}{2} \sum_{r=1}^{R} \int_{t=0}^{T} s(t) \left| \mathbf{u}(\mathbf{x}_r, t) - \mathbf{d}_r(t) \right|^2 \, dt, \tag{1}$$

where $s(t) > 0$ is a weight function and $|\mathbf{w}|$ denotes the magnitude of the vector $\mathbf{w} \in \Re^3$. Note that the misfit is a non-negative real scalar functional of $\mathbf{u}$, which accounts for differences between the time-dependent wave forms $\mathbf{u}(\mathbf{x}_r, t)$ and $\mathbf{d}_r(t)$ in the time interval $0 \leq t \leq T$. Hence, $\mathcal{X}_c = 0$ implies perfect agreement between the wave forms at all recording stations, i.e., $\mathbf{u}(\mathbf{x}_r, t) = \mathbf{d}_r(t)$ for $0 \leq t \leq T$ and $r = 1, 2, \ldots, R$.

We minimize the full waveform misfit functional (1) by using a gradient based optimization algorithm, where the gradient is efficiently computed by solving the adjoint problem. Full waveform inversion, i.e., inversion by minimizing the functional (1), is a technique that has gained popularity for seismic problems in recent years. For a recent example see [7], where both the material inversion and the source inversion problems are addressed. In that paper the elastic wave equation is solved by a spectral element method. The adjoint problem is formulated for the PDE and is discretized by the spectral element method. The minimization is done by a non-linear conjugate gradient method. Two different source models are used, one with seven parameters (six momentum components and depth) and one with nine parameters (six momentum components and location). Tromp et al. [19] discuss the seismic inversion problem for a number of different misfit functionals. They solve the source inversion problem in two space dimensions for a source with two parameters, by minimizing the travel time misfit functional. A similar approach is given in [17], where a source in two space dimensions is parametrized by its location and the source origin time. The source inversion problem is solved by minimizing the travel time misfit functional with respect to the three source parameters.

If only the location of the source is needed, a time-reversal method can be used instead of the full wave form inversion. In time-reversal, the elastic wave equation is solved backwards in time, using the observed data at the receivers as sources. If the entire displacement field were know at some time T, it would be possible to use this field as 'initial data' and solve the equation backwards from T, since the elastic wave equation is time reversible. The waves would then converge to the source location, which could be inferred. Often, the observed time history data at the receiver points is enough to recreate

the backward in time waves with good precision. This technique has been extensively used in acoustics, see [5]. An application of time-reversal to locate seismic sources is given in [9].

Source estimation by minimizing the full waveform misfit (1) can give unreliable results when the actual material properties (i.e. the earth) are poorly represented by the material model. To compensate for an imperfect material model, it is often necessary to introduce windowing functions into the misfit functional, to only select certain parts of the measured seismograms, see [7]. Windowing can be done through the function $s(t)$ in (1), where it is also possible to have different windows at different receivers. Another issue is that there is always some uncertainty in measured data. It can therefore be beneficial to consider both measured data and source parameters as probability distributions. A general description of the probabilistic approach is given [18]. See [3] for an application of these ideas to uncertainty quantification of seismic source inversion.

There is a large number of optimization methods available to perform the minimization of the misfit functional. Direct search methods, i.e., minimization algorithms that do not use the gradient information have been used to solve the inverse problem in seismology, e.g., the downhill simplex method was used in [10]. See [8] for a review of direct search minimization algorithms. In the same spirit are methods that use source-receiver reciprocity to efficiently compute the misfit functional for a large number of different source locations, see [4]. Although these methods are very robust, they tend to require a large amount of computational work as the number of undetermined parameters grows.

Optimization methods that make use of both the gradient and the Hessian (or an approximate Hessian), such as Newton and quasi-Newton methods have been successfully used for inverse problems, see, e.g., [6] for electromagnetic scattering and [15] for resistivity imaging in oil exploration. These methods are made efficient by approximating the Hessian by the positive definite secant update (BFGS), see [2] for a description of secant update quasi-Newton methods. For the source inversion problem in the current study, we have found that with good preconditioning, the non-linear conjugated gradient method is a very efficient alternative to quasi-Newton techniques.

In the following we assume that the displacement field $\mathbf{u}(\mathbf{x}, t)$ satisfies the elastic wave equation in the three-dimensional domain Ω, subject to initial and boundary conditions. Here, the boundary is denoted $\Gamma = \Gamma_1 \cup \Gamma_2$. The displacement is governed by

$$\begin{aligned}
\rho \mathbf{u}_{tt} &= \nabla \cdot \boldsymbol{\tau}(\mathbf{u}) + \mathbf{f}(\mathbf{x}, t; \mathbf{p}), && \mathbf{x} \in \Omega,\ 0 \le t \le T, \\
\mathbf{u}(\mathbf{x}, 0) &= 0, && \mathbf{x} \in \Omega,\ t = 0, \\
\mathbf{u}_t(\mathbf{x}, 0) &= 0, && \mathbf{x} \in \Omega,\ t = 0, \\
\mathbf{n} \cdot \boldsymbol{\tau}(\mathbf{u}) &= 0, && \mathbf{x} \in \Gamma_1,\ 0 \le t \le T, \\
\mathbf{u} &= 0, && \mathbf{x} \in \Gamma_2,\ 0 \le t \le T,
\end{aligned} \tag{2}$$

where $\rho(\mathbf{x})$ is the density. We further assume that the earth can be described as a heterogeneous isotropic elastic material. The stress tensor $\boldsymbol{\tau}(\mathbf{u})$ is then related to the displacement gradient through

$$\boldsymbol{\tau}(\mathbf{u}) = \lambda \operatorname{div}(\mathbf{u})\, I + \mu\, (\nabla \mathbf{u} + \nabla \mathbf{u}^T), \tag{3}$$

where $\lambda(\mathbf{x})$ and $\mu(\mathbf{x})$ are the first and second Lamé parameters of the material.

The function $\mathbf{f}$ in (2) describes the seismic source. For smaller seismic events, it can be modeled by a point moment tensor source,

$$\mathbf{f}(\mathbf{x}, t; \mathbf{p}) = g(t)\mathcal{M}\nabla\delta(\mathbf{x} - \mathbf{x}_s). \tag{4}$$

Here, $\nabla\delta(\mathbf{x})$ is the gradient of the Dirac distribution. The source time function $g(t) = g(t; t_0, \omega_0)$ is assumed to depend on two parameters; a time shift t_0 and a frequency parameter ω_0. The source is located at $\mathbf{x}_s = (x_s,\ y_s,\ z_s)$ and the elements of the symmetric matrix $\mathcal{M}$ are denoted

$$\mathcal{M} = \begin{pmatrix} m_{xx} & m_{xy} & m_{xz} \\ m_{xy} & m_{yy} & m_{yz} \\ m_{xz} & m_{yz} & m_{zz} \end{pmatrix}.$$

Under these assumptions the forcing function $\mathbf{f}$ depends on $P = 11$ parameters,

$$\mathbf{p} = (x_s,\ y_s,\ z_s,\ m_{xx},\ m_{xy},\ m_{xz},\ m_{yy},\ m_{yz},\ m_{zz},\ t_0,\ \omega_0). \tag{5}$$

The continuous source estimation problem can be stated as the constrained minimization problem

$$\min \mathcal{X}_c(\mathbf{p}), \quad \mathbf{u} \text{ satisfies (2) with forcing } \mathbf{f}(\mathbf{x}, t; \mathbf{p}).$$

Unfortunately, the elastic wave equation can not be solved analytically except in highly idealized situations, such as when the free surface boundary is flat and the material has homogeneous properties. For this reason, we discretize the elastic wave equation by a fourth order accurate finite difference method and solve it numerically. This allows us to account for general heterogeneous material properties and also make our approach extendable to realistic topographies.

The remainder of this article is organized in the following way. Section 2 gives an overview of our fourth order accurate finite difference discretization, defines the discrete source estimation problem, derives the adjoint of the discretized elastic wave equation, and proves the adjoint property. This property gives a relation between the solutions of the elastic and adjoint wave equations and their forcing functions, which is used in Section 3 to derive an efficient approach for computing the gradient and Hessian of the discrete misfit. Section 4 develops a new spatial discretization of the singular source function (4). This source discretization is designed to be compatible with a fourth order accurate difference scheme, and to be twice continuously differentiable with respect to the source location. Section 5 describes how ray tracing, combined with the linearity of the elastic wave equation, can be used to estimate an initial guess for the source parameters. In Section 6, we perform numerical experiments with the complete source inversion algorithm on a synthetic problem. We investigate how different scaling strategies affect the convergence rate of the minimization algorithm, and demonstrate that a scaling based on the Hessian makes the non-linear conjugate gradient method converge rapidly. Conclusions are given in Section 7.

2 The discretized problem

2.1 A self-adjoint fourth order accurate finite difference scheme

Consider the elastic wave equation (2) on the box shaped domain $(x, y, z) \in [0, x_{max}] \times [0, y_{max}] \times [0, z_{max}]$, and the time interval $0 \leq t \leq T$. Let the computational grid be

$$x_i = (i-1)h, \qquad y_j = (j-1)h, \quad \text{and} \quad z_k = (k-1)h,$$

where $h > 0$ is the grid size and i, j, and k are integers in the ranges $i \in [0, N_x + 1]$, $j \in [0, N_y + 1]$, and $k \in [0, N_z + 1]$. The domain sizes are chosen such that $x_{N_x} = x_{max}$, $y_{N_y} = y_{max}$, and $z_{N_z} = z_{max}$. The points with $i = 0$, $i = N_x + 1$, $j = 0$, $j = N_y + 1$, $k = 0$, or $k = N_z + 1$ are ghost points, which are used to impose boundary conditions. Time is discretized on the grid $t_n = n\Delta_t$, where $\Delta_t > 0$ is the fixed time step and n is an integer. The time step is chosen to satisfy the CFL stability condition, and $t_M = M\Delta_t = T$ where $M > 0$ is the total number of time steps.

The numerical approximation of the displacement vector $\mathbf{u}(\mathbf{x}, t)$ at grid point (i, j, k) and time level t_n is denoted by $\mathbf{u}^n_{i,j,k} = (u^n_{i,j,k}, v^n_{i,j,k}, w^n_{i,j,k})$. To improve readability, we occasionally suppress the subscript or superscript on $\mathbf{u}$, for example by writing $\mathbf{u}^n$ for $\mathbf{u}^n_{i,j,k}$. When convenient we also use the vector index notation $\mathbf{i} = (i, j, k)$ to indicate a spatial grid point index.

In [16], we developed a fourth order accurate symmetric discretization of the divergence of the stress tensor (3). This operator, denoted by $\mathbf{L}_h(\mathbf{u})$, has the property that

$$(\mathbf{v}, \mathbf{L}_h(\mathbf{u}))_h = (\mathbf{L}_h(\mathbf{v}), \mathbf{u})_h, \tag{6}$$

for any two grid functions $\mathbf{u}$ and $\mathbf{v}$ that satisfy the discretized boundary conditions

$$\mathbf{B}(\mathbf{u})_{i,j,k} = 0, \quad \mathbf{x}_{i,j,k} \in \Gamma. \tag{7}$$

The scalar product in (6) is defined by

$$(\mathbf{v}, \mathbf{u})_h = h^3 \sum_{k=1}^{N_z} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} a_{i,j,k} \langle \mathbf{v}_{i,j,k}, \mathbf{u}_{i,j,k} \rangle, \tag{8}$$

where $a_{i,j,k}$ are positive weights determined from the summation by parts property of $\mathbf{L}_h(\mathbf{u})$ that is needed to enforce (6). Also, $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{q=1}^{3} u^{(q)} v^{(q)}$, is the inner product between real-valued vectors with three components. Using this notation, the magnitude of $\mathbf{u}$ satisfies $|\mathbf{u}|^2 = \langle \mathbf{u}, \mathbf{u} \rangle$.

We consider boundary operators $\mathbf{B}$ that either discretize free surface or Dirichlet boundary conditions,

$$\mathbf{B}(\mathbf{u}^n)_{i,j,k} = \begin{cases} \mathcal{B}(\mathbf{u}^n)_{i,j,k} \mathbf{n}_{i,j,k}, & \text{Free surface,} \\ \mathbf{u}^n_{i,j,k}, & \text{Dirichlet.} \end{cases}$$

Here, $\mathcal{B}(\mathbf{u})$ is a special difference approximation of the stress tensor on the boundary that matches $\mathbf{L}_h(\mathbf{u})_{i,j,k}$ such that (6) is satsified, which makes the overall discretization stable.

The vector $\mathbf{n}_{i,j,k}$ is the outward boundary normal. A detailed description of the interior and boundary discretizations can be found in [12, 16].

We discretize the elastic wave equation (2) using the fourth order accurate difference method described in [16]. This method computes the displacement field $\mathbf{u}^n$, $n = 1, 2, \ldots, M$, as is outlined in Algorithm 1. Note that the grid function $\mathbf{F}(t; \mathbf{p})$ in this algorithm represents a discretization of the singular source term $\mathbf{f}(\mathbf{x}, t; \mathbf{p})$. This discretization will be described in detail in Section 4. The operator $\mathbf{S}_G(\mathbf{u})$ is a damping operator, used

Algorithm 1 *4th order accurate predictor-corrector scheme for the elastic wave equation.*

procedure FORWARD($\mathbf{u}, \mathbf{F}$)
  Initial conditions: $\mathbf{u}^0 = \mathbf{0}$ and $\mathbf{u}^{-1} = \mathbf{0}$
  for $n = 0, 1, \ldots, M-1$ **do**
    Predictor step:

$$\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1} + \frac{\Delta_t^2}{\rho}\big(\mathbf{L}_h(\mathbf{u}^n) + \mathbf{F}(t_n; \mathbf{p})\big)$$

    Impose boundary condition (7) on $\mathbf{u}^*$ to define its ghost point values
    Acceleration: $\mathbf{v}^n = \big(\mathbf{u}^* - 2\mathbf{u}^n + \mathbf{u}^{n-1}\big)/\Delta_t^2$
    Corrector step:

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \frac{\Delta_t^4}{12\rho}\big(\mathbf{L}_h(\mathbf{v}^n) + \mathbf{F}_{tt}(t_n; \mathbf{p})\big) + \mathbf{S}_G(\mathbf{u}^n - \mathbf{u}^{n-1})$$

    Impose boundary condition (7) on $\mathbf{u}^{n+1}$ to define its ghost point values
  end for
end procedure

in a supergrid sponge layer near artificial far-field boundaries. $\mathbf{S}_G(\mathbf{u})$ is consistent with

$$-\gamma h^4 \Delta_t \left((\sigma^{(x)}(x)\mathbf{u}_{xxt})_{xx} + (\sigma^{(y)}(y)\mathbf{u}_{yyt})_{yy} + (\sigma^{(z)}(z)\mathbf{u}_{zzt})_{zz} \right),$$

where γ is a constant that controls the strength of damping. The scalar, non-negative, taper functions $\sigma^{(x)}, \sigma^{(y)}$, and $\sigma^{(z)}$ are non-zero only in the sponge layers, where they increase from 0 to 1 in the outward direction through the layer. The domain is terminated at the artificial boundary by a homogeneous Dirichlet boundary condition, enforced on the approximation such that the symmetry property

$$(\mathbf{v}, \mathbf{S}_G(\mathbf{u}))_h = (\mathbf{S}_G(\mathbf{v}), \mathbf{u})_h \tag{9}$$

holds, and such that the discrete energy of the solution becomes non-negative and non-increasing. Note that the material properties (ρ, λ, and μ) are smoothly modified in the supergrid layer, such that the compressional and shear wave speeds become very small near the outer boundaries of the layer. The purpose of this modification is to slow down all waves that enter the layer to make the damping operator more efficient. More details on the supergrid technique, its discretization, and the proof of (9), will be given in [14].

2.2 The discrete source estimation problem

A straight forward generalization of the continuous formula (1) leads to the discrete misfit functional

$$\mathcal{X}(\mathbf{p}) = \frac{1}{2}\sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)|\mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n)|^2. \tag{10}$$

As in the continuous problem, $s(t_n) > 0$ is a weight function. We assume that all recording stations coincide with grid points, i.e., $\mathbf{x}_r = \mathbf{x}_{\mathbf{i}_r}$ for some vector index $\mathbf{i}_r = (i_r, j_r, k_r)$. Furthermore, the observed displacements $\mathbf{d}_r(t)$ are assumed to have already been filtered in time such that they only contain motions that can be captured on the computational grid.

Similar to the continuous case, the displacement at the recording stations depends implicitly on the parameter vector $\mathbf{p}$ in the discretized forcing function $\mathbf{F}$. Given the source parameters $\mathbf{p}$, we can use Algorithm 1 to calculate the solution of the elastic wave equation, which then can be inserted into (10) to evaluate the discrete misfit $\mathcal{X}(\mathbf{p})$. Hence, the discrete source estimation problem can be stated as the constrained minimization problem,

$$\min \mathcal{X}(\mathbf{p}), \quad \mathbf{u}^n \text{ is calculated by Algorithm 1 with forcing } \mathbf{F}(t_n; \mathbf{p}).$$

2.3 The adjoint wave equation

An efficient approach for computing the gradient of the misfit uses the adjoint wave field, $\boldsymbol{\kappa}_{\mathbf{i}}^n$. Let the source term in the adjoint equation be $\mathbf{G}(t_n)$. The adjoint wave field satisfies the adjoint of the discretized elastic wave equation. A method for calculating $\boldsymbol{\kappa}$ is given in Algorithm 2.

Algorithm 2 *The adjoint of the 4th order scheme for the elastic wave equation.*

procedure ADJOINT($\boldsymbol{\kappa}, \mathbf{G}$)
  Terminal conditions: $\boldsymbol{\kappa}^{M-1} = \mathbf{0}$ and $\boldsymbol{\kappa}^{M} = \mathbf{0}$
  for $n = M-1, M-2, \ldots, 1$ **do**
    Predictor step:

$$\boldsymbol{\kappa}^* = 2\boldsymbol{\kappa}^n - \boldsymbol{\kappa}^{n+1} + \Delta_t^2 \frac{\mathbf{L}_h(\boldsymbol{\kappa}^n)}{\rho} \tag{11}$$

    Impose boundary condition (7) on $\boldsymbol{\kappa}^*$ to define its ghost point values
    Compute acceleration: $\boldsymbol{\zeta}^n = \big(\boldsymbol{\kappa}^* - 2\boldsymbol{\kappa}^n + \boldsymbol{\kappa}^{n+1}\big)/\Delta_t^2$
    Corrector step:

$$\boldsymbol{\kappa}^{n-1} = \boldsymbol{\kappa}^* + \frac{\Delta_t^4}{12}\frac{\mathbf{L}_h(\boldsymbol{\zeta}^n)}{\rho} + \frac{\Delta_t^2}{\rho}\mathbf{G}(t_n) - \mathbf{S}_G(\boldsymbol{\kappa}^{n+1} - \boldsymbol{\kappa}^n), \tag{12}$$

    Impose boundary condition (7) on $\boldsymbol{\kappa}^{n-1}$ to define its ghost point values
  end for
end procedure

The adjoint property is made precise in the following theorem.

Theorem 1. *Let* $\mathbf{F}$ *(with second time derivative* $\mathbf{F}_{tt}$*) be the source term in the discretized elastic wave equation, and use Algorithm 1 to calculate* $\mathbf{u}$*. Furthermore, let* $\mathbf{G}$ *be the source term in the adjoint wave equation and use Algorithm 2 to calculate* $\boldsymbol{\kappa}$ *(with acceleration* $\boldsymbol{\zeta}$*). Then the grid functions* $\mathbf{u}$ *and* $\boldsymbol{\kappa}$ *are adjoint in the sense that*

$$\sum_{n=0}^{M-1} (\mathbf{G}^n, \mathbf{u}^n)_h = \sum_{n=0}^{M-1} \left(\boldsymbol{\kappa}^n, \mathbf{F}^n + \frac{\Delta_t^2}{12} \mathbf{F}_{tt}^n \right)_h + \frac{\Delta_t^2}{12} \sum_{n=0}^{M-1} (\boldsymbol{\zeta}^n, \mathbf{F}^n)_h \,. \tag{13}$$

Proof: See Appendix A.

3 Minimizing the misfit

We use a preconditioned Fletcher-Reeves algorithm to minimize the discrete misfit. This technique generalizes the conjugate gradient method to non-quadratic problems, see for example [11]. The preconditioning corresponds to a change of variables, $\hat{\mathbf{p}} = S\mathbf{p}$, where S is a non-singular matrix. The change of variables is introduced to improve the convergence properties of the Fletcher-Reeves algorithm. We first formulate the minimization algorithm in the scaled variables, and then transform it back to the original variables. The resulting algorithm with m restarts, and where the parameter vector $\mathbf{p}$ has P components is given in Algorithm 3. The algorithm terminates after all restarts have been completed, or when the maximum norm of the scaled gradient is smaller than the tolerance θ, $0 < \theta \ll 1$. In practice we usually set $\theta = 10^{-12}$. Note that the algorithm is given for a general preconditioning matrix S. When S is diagonal, $S^T S = S^2$. The Fletcher-Reeves algorithm uses the gradient of the misfit with respect to the components of the parameter vector $\mathbf{p}$. It is defined by

$$\nabla \mathcal{X}(\mathbf{p}) = \left(\frac{\partial \mathcal{X}}{\partial p_1}, \frac{\partial \mathcal{X}}{\partial p_2}, \ldots, \frac{\partial \mathcal{X}}{\partial p_P} \right)^T .$$

In Section 3.1, we discuss an efficient approach for calculating all components of the gradient by solving one adjoint wave equation.

The convergence properties of the Fletcher-Reeves algorithm depend on the properties of the scaled Hessian matrix, with elements $\hat{H}_{i,j} = \partial^2 \mathcal{X} / \partial \hat{p}_i \partial \hat{p}_j$. In matrix notation, we have

$$\hat{H} = S^{-T} H S^{-1}, \quad H_{i,j} = \frac{\partial^2 \mathcal{X}}{\partial p_i \partial p_j},$$

where H is the unscaled Hessian. Here, S^{-T} denotes the transpose of the matrix S^{-1}. The conjugated gradient algorithm is not guaranteed to converge unless the Hessian matrix is a continuous function of all parameters, i.e., the misfit function must be twice continuously differentiable with respect to $\mathbf{p}$. It is straightforward to see that the displacement field depends linearly on the matrix elements of $\mathcal{M}$. Hence $\mathbf{u}$ and thereby $\mathcal{X}$ are infinitely differentiable with respect to the elements of $\mathcal{M}$. We assume that the time function depends on t_0 through a time shift $g(t; t_0, \omega_0) = \tilde{g}(t - t_0; \omega_0)$. Because the source term

Algorithm 3 *The preconditioned Fletcher-Reeves algorithm. Here,* $\nabla \mathcal{X}_k = \nabla \mathcal{X}(\mathbf{p}_k)$.

procedure PRECOND-FLETCHER-REEVES($\mathbf{p}_0$)
  for $r = 1, 2, \ldots, m$ **do**
    Initial search direction: $\mathbf{q}_0 = -(S^T S)^{-1} \nabla \mathcal{X}(\mathbf{p}_0)$
    for $k = 0, 1, \ldots, P-1$ **do**
      Line search: find steplength α_k that minimizes $\mathcal{X}(\mathbf{p}_k + \alpha_k \mathbf{q}_k)$
      Next parameter vector: $\mathbf{p}_{k+1} = \mathbf{p}_k + \alpha_k \mathbf{q}_k$
      Compute β_k:

$$\beta_k = \frac{\nabla \mathcal{X}_{k+1}^T (S^T S)^{-1} \nabla \mathcal{X}_{k+1}}{\nabla \mathcal{X}_k^T (S^T S)^{-1} \nabla \mathcal{X}_k}$$

      Next search direction: $\mathbf{q}_{k+1} = -(S^T S)^{-1} \nabla \mathcal{X}(\mathbf{p}_{k+1}) + \beta_k \mathbf{q}_k$
      if $||S^{-1} \nabla \mathcal{X}_{k+1}||_\infty < \theta$ **then**
        $\mathbf{p}_0 = \mathbf{p}_{k+1}$
        return
      end if
    end for
    Initial guess for next outer iteration $\mathbf{p}_0 = \mathbf{p}_P$
  end for
end procedure

F enters into the finite difference scheme with two time derivatives (see Algorithm 1), a requirement for the Hessian to be defined is that $\tilde{g}(t; \omega_0)$ is four times differentiable with respect to t and twice differentiable with respect to ω_0. In this article, $\tilde{g}$ is assumed to have this regularity. The spatial discretization and regularity of the moment tensor source with respect to its location, $\mathbf{x}_s$, are discussed in Section 4.

A crucial component of the preconditioned Fletcher-Reeves algorithm is the line search algorithm, i.e., to minimize $\mathcal{X}(\mathbf{p}_k + \alpha \mathbf{q}_k)$ with respect to the step length α. In our implementation, we use the backtracking algorithm A6.3.1 in [2], with the minor modification that the full step is $\mathbf{p}_k + \alpha_s \mathbf{q}_k$, instead of $\mathbf{p}_k + \mathbf{q}_k$, as assumed in [2]. The full step size α_s is taken from the linear conjugated gradient algorithm, which assumes that $\mathcal{X}(\mathbf{p})$ is a quadratic function of $\mathbf{p}$,

$$\alpha_s = -\frac{\nabla \mathcal{X}_k^T \mathbf{q}_k}{\mathbf{q}_k^T H_k \mathbf{q}_k}. \tag{14}$$

In our case, the Hessian is evaluated at $\mathbf{p}_k$, i.e., $H_k = H(\mathbf{p}_k)$. In Section 3.2, we present an algorithm for evaluating $\mathbf{q}_k^T H_k \mathbf{q}_k$ that only requires one additional wave equation to be solved. In most iterations of the Fletcher-Reeves algorithm, the full step length α_s yields an acceptable approximation of the minimum in the line search. Hence, the backtracking is rarely invoked. The "typical x" vector, needed in backtracking algorithm A6.3.1 (see [2]), is taken as the inverse of the diagonal elements of the scaling matrix S.

3.1 The gradient of the misfit

Straightforward differentiation of (10) gives

$$\frac{\partial \mathcal{X}}{\partial p_j} = \sum_{r=1}^{R} \sum_{n=0}^{M-1} s(t_n) \left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j} \right\rangle. \tag{15}$$

Note that the material properties ρ, μ, and λ do not depend on p_j. By differentiating the difference scheme for $\mathbf{u}$ with respect to p_j, we see that $\partial \mathbf{u}/\partial p_j$ could be calculated with the same finite difference scheme as used for computing $\mathbf{u}$, if the source term $\mathbf{F}$ were replaced by $\partial \mathbf{F}/\partial p_j$. However, to compute the gradient of $\mathcal{X}$ with this technique, it would be necessary to solve the elastic wave equation with 11 different forcing functions, where each forcing corresponds to one component of $\partial \mathcal{X}/\partial p_j$.

A more efficient way of computing the gradient of the misfit is based on solving the adjoint wave equation. In this approach, we define the adjoint source in (12) as

$$\mathbf{G}_{\mathbf{i}}^n = \sum_{r=1}^{R} s(t_n) \left(\mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n) \right) \frac{\delta_{\mathbf{i},\mathbf{i}_r}}{h^3 a_{\mathbf{i}}}, \tag{16}$$

where $a_{\mathbf{i}}$ is the weight coefficient in the scalar product (8) and

$$\delta_{\mathbf{i},\mathbf{j}} = \begin{cases} 1, & \mathbf{i} = \mathbf{j}, \\ 0, & \text{otherwise.} \end{cases}$$

Inserting (16) into (15) shows that the gradient of the misfit can be written

$$\frac{\partial \mathcal{X}}{\partial p_j} = \sum_{n=0}^{M-1} \left(\mathbf{G}^n, \frac{\partial \mathbf{u}^n}{\partial p_j} \right)_h.$$

Because $\partial \mathbf{u}/\partial p_j$ satisfies the forward finite difference scheme with source term $\partial \mathbf{F}/\partial p_j$, we can apply Theorem 1 to obtain

$$\frac{\partial \mathcal{X}}{\partial p_j} = \sum_{n=0}^{M-1} \left(\boldsymbol{\kappa}^n, \frac{\partial \mathbf{F}^n}{\partial p_j} + \frac{\Delta_t^2}{12} \frac{\partial \mathbf{F}_{tt}^n}{\partial p_j} \right)_h + \frac{\Delta_t^2}{12} \sum_{n=0}^{M-1} \left(\boldsymbol{\zeta}^n, \frac{\partial \mathbf{F}^n}{\partial p_j} \right)_h. \tag{17}$$

Equation (17) allows us to calculate all components of the gradient from the adjoint wave field $\boldsymbol{\kappa}_{\mathbf{i}}^n$. The scalar products involving the gradients of $\mathbf{F}$ can be assembled during the time stepping of the adjoint scheme. Because the forcing function $\mathbf{F}$ only is non-zero at a few grid points near $\mathbf{x}_s$, the computational cost of evaluating these scalar products is insignificant compared to solving the adjoint wave equation.

3.2 Calculating the Hessian and $\mathbf{q}^T H \mathbf{q}$

The Hessian matrix plays an important role in gradient based optimization. For example, the condition number of the Hessian governs the convergence rate of the conjugate gradient algorithm, and the Hessian can be used to construct a preconditioner.

To compute the Hessian, we differentiate (15) with respect to p_k to obtain

$$H_{k,j} := \frac{\partial}{\partial p_k}\left(\frac{\partial \mathcal{X}}{\partial p_j}\right) = \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\frac{\partial}{\partial p_k}\left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}\right\rangle$$
$$= \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k}, \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}\right\rangle + \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial^2 \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k \partial p_j}\right\rangle. \quad (18)$$

We decompose the Hessian into two parts, $H = H^{(1)} + H^{(2)}$, where

$$H_{j,k}^{(1)} := \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k}, \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}\right\rangle, \quad (19)$$

$$H_{j,k}^{(2)} := \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle \mathbf{u}_{\mathbf{i}_r}^n - \mathbf{d}_r(t_n), \frac{\partial^2 \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k \partial p_j}\right\rangle. \quad (20)$$

By noting the similarities between (15) and (20), we see that the matrix $H^{(2)}$ can also be computed using the adjoint wave field $\boldsymbol{\kappa}$. We arrive at the formula

$$H_{j,k}^{(2)} = \sum_{n=0}^{M-1}\left(\boldsymbol{\kappa}^n, \frac{\partial^2 \mathbf{F}^n}{\partial p_k \partial p_j} + \frac{\Delta_t^2}{12}\frac{\partial^2 \mathbf{F}_{tt}^n}{\partial p_k \partial p_j}\right)_h + \frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}\left(\boldsymbol{\zeta}^n, \frac{\partial^2 \mathbf{F}^n}{\partial p_k \partial p_j}\right)_h. \quad (21)$$

Note that this formula is similar to (17), except that the first derivative of the forcing has been replaced by its second derivative. Hence, we can obtain $H^{(2)}$ by accumulating additional scalar products during the time stepping of the adjoint wave equation. Therefore, the computation of $H^{(2)}$ does not require any additional elastic wave equations to be solved. However, calculating $H^{(1)}$ requires the quantities $\partial \mathbf{u}_{\mathbf{i}}^n / \partial p_j$ to be known, which satisfy the elastic wave equation with the forcing term $\partial \mathbf{F}/\partial p_j$. Hence, an additional 11 elastic wave equations must be solved to assemble the matrix $H^{(1)}$.

The higher computational cost of calculating the Hessian makes it prohibitively expensive to evaluate in each iteration of Algorithm 3. However, as we will see below, it is highly advantageous to compute the Hessian at least once, and use it as a preconditioner throughout the iteration.

The step length calculation (14) for α_s requires the computation of the scalar quantity $\mathbf{q}^T H \mathbf{q}$, where $\mathbf{q}$ is a vector with $P = 11$ components. As before, we decompose the Hessian into $H = H^{(1)} + H^{(2)}$. The second term, $\mathbf{q}^T H^{(2)} \mathbf{q}$, is directly available after $H^{(2)}$ has been calculated, as described above. For the first term, we note that

$$\mathbf{q}^T H^{(1)} \mathbf{q} = \sum_{j=1,k=1}^{P}\sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle q_j \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_j}, \frac{\partial \mathbf{u}_{\mathbf{i}_r}^n}{\partial p_k} q_k\right\rangle.$$

Let $\tilde{\mathbf{u}}_{\mathbf{i}}^n$ denote the solution obtained by solving the discretized elastic wave equation with the forcing term $\sum_j q_j \frac{\partial \mathbf{F}(t_n)}{\partial p_j}$. It then holds that $\tilde{\mathbf{u}}_{\mathbf{i}}^n = \sum_j q_j \frac{\partial \mathbf{u}_{\mathbf{i}}^n}{\partial p_j}$, and hence,

$$\mathbf{q}^T H^{(1)} \mathbf{q} = \sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left\langle \tilde{\mathbf{u}}_{\mathbf{i}_r}^n, \tilde{\mathbf{u}}_{\mathbf{i}_r}^n\right\rangle, \quad (22)$$

can be assembled during the time stepping calculation of $\tilde{\mathbf{u}}^n$. The cost for calculating $\mathbf{q}^T H \mathbf{q}$ therefore amounts to solving one additional elastic wave equation, which also is needed when the step length is estimated by an approximate difference quotient, such as in [7]. The advantage of using a step length based on (22) is that the errors in this difference approximation are avoided.

4 Discretizing the singular source term

The gradient of the Dirac distribution in the seismic source term (4) can discretized based on the discretization of a one-dimensional Dirac distributions $\delta(x - x_s)$, and its derivative $\delta'(x - x_s)$. For all smooth, compactly supported functions of one variable $\varphi(x)$, we have

$$\int \varphi(x)\delta(x - x_s)\, dx = \varphi(x_s) \qquad \int \varphi(x)\frac{d\delta}{dx}(x - x_s)\, dx = -\frac{d\varphi}{dx}(x_s). \tag{23}$$

Our approach is based on the technique in [13], which approximates the singular sources numerically by grid functions that satisfy (23) in a discrete scalar product for all polynomial functions up to order $q > 0$, leading to $q + 1$ moment conditions. The required order is related to the order of accuracy in the approximation of the differential equation. Because we use gradient based optimization to solve for the source location, the source term discretization must be twice continuously differentiable with respect to $\mathbf{x}_s$. As a result, the moment conditions must be augmented by additional continuity conditions.

For a fourth order accurate scheme, the moment conditions for δ should be satisfied for the functions $\varphi(x) = x^k$, $k = 0, \ldots, 3$, and the moment conditions for δ' should be satisfied for $k = 0, \ldots, 4$. Details are given in [13]. To make the technique easier to implement, we use discretizations that satisfy the moment conditions for $k = 0, \ldots, 4$, both for δ and δ'.

We describe the discretization of δ and its derivative in one space dimension. The multi-dimensional approximation can be obtained in a straightforward way by Cartesian products of the one-dimensional discretizations. Let the one-dimensional grid be $x_j = jh$, $j = 0, \ldots, N + 1$, and define the scalar product by $(u, v)_{h1} = h \sum_{j=1}^{N} u_j v_j$. Furthermore, let the grid function $\tilde{b}_j = \tilde{b}(x_s, j_s)_j$ denote a preliminary approximation of $\delta(x - x_s)$, which is centered at grid point j_s. We make the straightforward choice $\tilde{b}_j = 0$ for $j < j_s - 2$ or $j > j_s + 2$, and determine the five coefficients in $\tilde{b}_j$, $j_s - 2 \le j \le j_s + 2$, by solving the system formed by the five moment conditions

$$\left(x^k, \tilde{b}\right)_{h1} = (x_s)^k, \quad k = 0, \ldots, 4. \tag{24}$$

The moment conditions on $\tilde{b}$ do not impose any specific relation between j_s and x_s. However, for accuracy reasons, we want to center the stencil near x_s. For example, we may choose j_s such that $x_{j_s} - h/2 \le x_s < x_{j_s} + h/2$. Within this interval, $\tilde{b}$ is infinitely differentiable with respect to x_s, because the elements $\tilde{b}_j$ are either zero, or depend on x_s through the right hand side of (24), which is a polynomial in x_s. However, if $x_s = x_{j_s} + h/2 + \varepsilon$, the stencil will be centered around grid point x_{j_s} for $\varepsilon < 0$, but around grid point x_{j_s+1} for $\varepsilon \ge 0$. Unfortunately, the elements of $\tilde{b}$ are not continuously differentiable with respect to x_s at $\epsilon = 0$, where the stencil switches center point.

The lack of continuity with respect to x_s has been observed to hamper the convergence of the non-linear conjugated gradient algorithm. We will therefore replace $\tilde{b}(x_s, j_s)$ by a smoother source discretization, denoted by $b(x_s, j_s)$. The properties of $b(x_s, j_s)$ are given in the following theorem.

Theorem 2. *Let j_s be defined by $x_{j_s} \le x_s < x_{j_s+1}$, set $\nu = (x_s - x_{j_s})/h$, and let $\psi(\nu)$ be an m times continuously differentiable function having the properties*

$$\psi(0) = 0, \quad \psi(1) = 1, \quad \frac{d^l\psi}{d\nu^l}(0) = \frac{d^l\psi}{d\nu^l}(1) = 0, \quad l = 1, \ldots, m. \tag{25}$$

Furthermore, assume that $\tilde{b}(x_s, j_s)$ satisfies (24). Then, the grid function

$$b(x_s, j_s) = (1 - \psi(\nu))\, \tilde{b}(x_s, j_s) + \psi(\nu)\tilde{b}(x_s, j_s + 1), \tag{26}$$

is m times continuously differentiable with respect to x_s, and satisfies the moment conditions (24).

Proof. Because $d/d\nu = h d/dx_s$, differentiability with respect to ν is equivalent with differentiability with respect to x_s. Also, $x_{j_s} \le x_s < x_{j_s+1}$ implies $0 \le \nu < 1$. As noted above, the elements of $\tilde{b}(x_s, j_s)$ and $\tilde{b}(x_s, j_s + 1)$ are polynomials in x_s, and are thus infinitely differentiable with respect to x_s. Because ψ is m times continuously differentiable, we conclude that b is m times continuously differentiable for $0 < \nu < 1$. At the stencil switching point $x_s = x_{j_s+1}$, the continuity conditions become

$$\frac{\partial^l b(x_s, j_s)}{\partial x_s^l}|_{x_s \to x_{j_s+1}} = \frac{\partial^l b(x_s, j_s + 1)}{\partial x_s^l}|_{x_s = x_{j_s+1}}, \quad l = 0, \ldots, m. \tag{27}$$

Leibniz's product rule gives

$$\begin{aligned}\frac{\partial^l b(x_s, j_s)}{\partial x_s^l} = (1 - \psi(\nu))\frac{\partial^l \tilde{b}(x_s, j_s)}{\partial x_s^l} + \psi(\nu)\frac{\partial^l \tilde{b}(x_s, j_s + 1)}{\partial x_s^l} \\ + \sum_{q=1}^{l} \binom{l}{q} \frac{1}{h^q} \frac{d^q\psi}{d\nu^q}(\nu) \left(\frac{\partial^{l-q} \tilde{b}(x_s, j_s + 1)}{\partial x_s^{l-q}} - \frac{\partial^{l-q} \tilde{b}(x_s, j_s)}{\partial x_s^{l-q}} \right).\end{aligned} \tag{28}$$

The properties of ψ in (25) give, for $\nu = 0$,

$$\frac{\partial^l b(x_s, j_s)}{\partial x_s^l}|_{x_s = x_{j_s}} = \frac{\partial^l \tilde{b}(x_s, j_s)}{\partial x_s^l}|_{x_s = x_{j_s}}, \quad l = 0, 1, \ldots, m, \tag{29}$$

and for $\nu \to 1$,

$$\frac{\partial^l b(x_s, j_s)}{\partial x_s^l}|_{x_s \to x_{j_s+1}} = \frac{\partial^l \tilde{b}(x_s, j_s + 1)}{\partial x_s^l}|_{x_s = x_{j_s+1}}, \quad l = 0, 1, \ldots, m. \tag{30}$$

Hence, by applying (29) to the source discretization centered at $j_s + 1$, we get

$$\frac{\partial^l b(x_s, j_s + 1)}{\partial x_s^l}|_{x_s = x_{j_s+1}} = \frac{\partial^l \tilde{b}(x_s, j_s + 1)}{\partial x_s^l}|_{x_s = x_{j_s+1}}, \quad l = 0, 1, \ldots, m. \tag{31}$$

The continuity conditions (27) now follow from (30) and (31).

The scalar product in the moment conditions (24) is computed by summation over the elements of the grid function b. This summation is clearly independent of j_s and x_s, so that we have

$$(x^k, b)_{h1} = (1-\psi(\nu))\left(x^k, \tilde{b}(x_s, j_s)\right)_{h1} + \psi(\nu)\left(x^k, \tilde{b}(x_s, j_s+1)\right)_{h1} \\ = (1-\psi(\nu))(x_s)^k + \psi(\nu)(x_s)^k = (x_s)^k,$$

for $k = 0, \ldots, 4$. Therefore, $b(x_s, j_s)$ satisfies the moment condition (24). □

We want to construct a source discretization with two continuous derivatives and apply Theorem 2 to the case $m = 2$. We start from the blending function

$$\psi(\nu) = \begin{cases} 0, & \nu < 0, \\ 10\nu^3 - 15\nu^4 + 6\nu^5, & 0 \le \nu < 1, \\ 1, & \nu \ge 1, \end{cases}$$

which is monotonically increasing for $0 < \nu < 1$ and has two continuous derivatives at the break points $\nu = 0$ and $\nu = 1$. Since $\tilde{b}(x_s, j_s)$ is non-zero at five points, the grid function b has six non-zero elements. After some algebra, we find that the coefficients in the stencil (26) are given by

$$b(x_s, j_s)_{j_s-2} = \frac{1}{h}\left(\frac{1}{12}\nu - \frac{1}{24}\nu^2 - \frac{1}{12}\nu^3 - \frac{19}{24}\nu^4 + P(\nu)\right), \tag{32}$$

$$b(x_s, j_s)_{j_s-1} = \frac{1}{h}\left(-\frac{2}{3}\nu + \frac{2}{3}\nu^2 + \frac{1}{6}\nu^3 + 4\nu^4 - 5P(\nu)\right), \tag{33}$$

$$b(x_s, j_s)_{j_s} = \frac{1}{h}\left(1 - \frac{5}{4}\nu^2 - \frac{97}{12}\nu^4 + 10P(\nu)\right), \tag{34}$$

$$b(x_s, j_s)_{j_s+1} = \frac{1}{h}\left(\frac{2}{3}\nu + \frac{2}{3}\nu^2 - \frac{1}{6}\nu^3 + \frac{49}{6}\nu^4 - 10P(\nu)\right), \tag{35}$$

$$b(x_s, j_s)_{j_s+2} = \frac{1}{h}\left(-\frac{1}{12}\nu - \frac{1}{24}\nu^2 + \frac{1}{12}\nu^3 - \frac{33}{8}\nu^4 + 5P(\nu)\right), \tag{36}$$

$$b(x_s, j_s)_{j_s+3} = \frac{1}{h}\left(\frac{5}{6}\nu^4 - P(\nu)\right), \tag{37}$$

where

$$P(\nu) = \frac{5}{3}\nu^5 - \frac{7}{24}\nu^6 - \frac{17}{12}\nu^7 + \frac{9}{8}\nu^8 - \frac{1}{4}\nu^9,$$

and $b(x_s, j_s)_j = 0$ for all other j.

Let $e(x_s, j_s)_j$ denote the grid function approximating the derivative of the Dirac distribution, $\delta'(x - x_s)$. Following the same approach as above, we arrive at the six point

stencil

$$e(x_s, j_s)_{j_s-2} = \frac{1}{h^2}\left(-\frac{1}{12} + \frac{1}{12}\nu + \frac{1}{4}\nu^2 + \frac{2}{3}\nu^3 + R(\nu)\right), \tag{38}$$

$$e(x_s, j_s)_{j_s-1} = \frac{1}{h^2}\left(\frac{2}{3} - \frac{4}{3}\nu - \frac{1}{2}\nu^2 - \frac{7}{2}\nu^3 - 5R(\nu)\right), \tag{39}$$

$$e(x_s, j_s)_{j_s} = \frac{1}{h^2}\left(\frac{5}{2}\nu + \frac{22}{3}\nu^3 + 10R(\nu)\right), \tag{40}$$

$$e(x_s, j_s)_{j_s+1} = \frac{1}{h^2}\left(-\frac{2}{3} - \frac{4}{3}\nu + \frac{1}{2}\nu^2 - \frac{23}{3}\nu^3 - 10R(\nu)\right), \tag{41}$$

$$e(x_s, j_s)_{j_s+2} = \frac{1}{h^2}\left(\frac{1}{12} + \frac{1}{12}\nu - \frac{1}{4}\nu^2 + 4\nu^3 + 5R(\nu)\right), \tag{42}$$

$$e(x_s, j_s)_{j_s+3} = \frac{1}{h^2}\left(-\frac{5}{6}\nu^3 - R(\nu)\right). \tag{43}$$

Here, the definition of ν and the relation between j_s and x_s are the same as for the grid function $b(x_s, j_s)$ above. The polynomial R is given by

$$R(\nu) = -\frac{25}{12}\nu^4 - \frac{3}{4}\nu^5 + \frac{59}{12}\nu^6 - 4\nu^7 + \nu^8,$$

and $e(x_s, j_s)_j = 0$ for $j < j_s - 2$ or $j > j_s + 3$.

It can be verified that the grid function e satisfies the moment conditions for a fourth order accurate discretization of $\delta'(x - x_s)$,

$$(1, e)_{h1} = 0 \qquad (x^k, e)_{h1} = -k(x_s)^{k-1} \quad k = 1, \ldots, 4,$$

and is twice continuously differentiable with respect to the source location x_s.

5 Estimating initial source parameters

Figure 1 shows contour levels of $\mathcal{X}$ in two planes of the 11-dimensional parameter space, where the remaining nine parameters are held at their minimizing values. This example is taken from the layer over half space problem called LOH.1, see Section 6 for details. The minimum is clearly visible at $x_s = y_s = 15000$ and $z_s = 2000$. Gradient based minimization algorithms assume that the objective function is close to quadratic in parameter space. Figure 1 shows that this assumption only holds close to the minimum. Furthermore, the local minima in these cross-sections of parameter space indicate that $\mathcal{X}$ may have several local minima. To make the minimization algorithm converge to the global minimum, it follows that the initial parameter guess must be fairly accurate. We proceed by describing an approach for establishing initial parameter values for the source estimation problem.

5.1 Initial estimate for the source location and start time

Our initial estimate for the source location is based on first arrival times. Assume that the first wave arrives at time t_r at receiver location (x_r, y_r, z_r). If the material has homogeneous properties with compressional wave velocity c_p, the travel time from source location

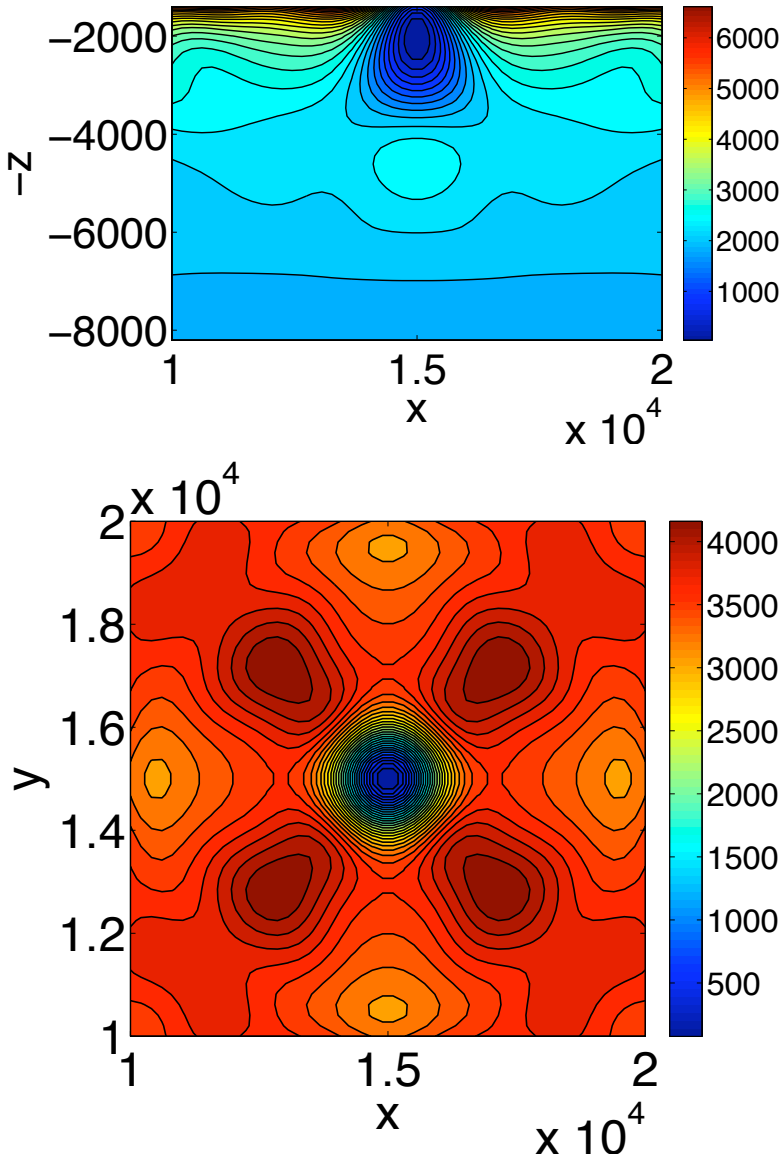

Figure 1: Contour plots of $\mathcal{X}$ as function of the source location for the LOH.1 problem. Top: $\mathcal{X}(x_s, z_s)$ for $y_s = 15000$. Bottom: $\mathcal{X}(x_s, y_s)$ for $z_s = 2000$. All other parameters are held at their correct values.

(x_s, y_s, z_s) to receiver 'r' satisfies

$$\hat{T}_r(x_s, y_s, z_s) = \frac{1}{c_p}\sqrt{(x_r - x_s)^2 + (y_r - y_s)^2 + (z_r - z_s)^2}.$$

The source starting time, t_s, is related to the first arrival time, t_r, through $\hat{T}_r(x_s, y_s, z_s) + t_s = t_r$. Hence, we consider solving

$$I_r(x_s, y_s, z_s, t_s) := \hat{T}_r(x_s, y_s, z_s) + t_s - t_r = 0, \quad r = 1, \ldots, R. \tag{44}$$

Each receiver results in one equation for the four unknowns (x_s, y_s, z_s, t_s), so we need at least four receivers. Usually we have more than four receivers, which makes (44) an overdetermined system. It can be solved in the least squares sense using the Gauss-Newton method.

The first arrival time at receiver r is defined as the smallest value of t_m for which

$$|u^m_{\mathbf{i}_r}| > \eta \max_n |u^n_{\mathbf{i}_r}| \text{ or } |v^m_{\mathbf{i}_r}| > \eta \max_n |v^n_{\mathbf{i}_r}| \text{or } |w^m_{\mathbf{i}_r}| > \eta \max_n |w^n_{\mathbf{i}_r}|, \quad 0 < \eta \ll 1, \tag{45}$$

i.e., the earliest time for which the amplitude of any component of the signal reaches a fraction of its peak value. In the numerical experiments below we use $\eta = 10^{-6}$.

If the material is not homogeneous, the above procedure can still be applied if c_p is replaced by an appropriate average of the compressional wave speed. However, we have found that such an approach can be sensitive to the value of c_p. In some cases it even makes the Gauss-Newton iteration diverge. For heterogeneous materials, a better approach is to estimate the initial source position by ray tracing, which takes the variable material properties into account. We here describe the approach for a layered material model. Consider a material with piecewise constant compressional wave speed in m horizontal layers, depending only on the depth below the free surface,

$$c_p(z) = \begin{cases} c_1, & 0 \le z < Z_1, \\ c_2, & Z_1 \le z < Z_2, \\ & \cdots \\ c_m, & Z_{m-1} \le z \le Z_m = z_{max}. \end{cases}$$

Assume that the source is located at $(x_s,\ y_s,\ z_s)$ with $Z_{n-1} < z_s < Z_n$, and that the receiver is located at $(x_r,\ y_r,\ z_r)$ with $0 \le z_r < Z_1$. The case $n = 1$ correspond to a homogeneous material, and we assume $n \ge 2$. The piecewise linear path between source and receiver with break points at (X_i, Y_i, Z_i), $i = 1, \ldots, n-1$ has the travel time

$$T_r = \sum_{i=1}^{n} \frac{\xi_i}{c_i}, \quad \xi_i = \begin{cases} \sqrt{(X_i - x_r)^2 + (Y_i - y_r)^2 + (Z_i - z_r)^2}, & i = 1, \\ \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2 + (Z_i - Z_{i-1})^2}, & 2 \le i \le n-1, \\ \sqrt{(x_s - X_{i-1})^2 + (y_s - Y_{i-1})^2 + (z_s - Z_{i-1})^2}, & i = n. \end{cases}$$

The travel time is a function of the locations of the break points (X_i, Y_i, Z_i), $i = 1, \ldots, n-1$. Since Z_i are assumed to be fixed, the ray from source to receiver is determined by the values of (X_i, Y_i) that minimize T_r.

At the minimum, $\partial T_r/\partial X_i = 0$ and $\partial T_r/\partial Y_i = 0$, for $i = 1, 2, \ldots, n-1$. This results in the system

$$A(\mathbf{x}, \mathbf{y})\mathbf{x} = \mathbf{r}^{(1)}, \quad A(\mathbf{x}, \mathbf{y})\mathbf{y} = \mathbf{r}^{(2)}, \tag{46}$$

where $\mathbf{x} = (X_1, \ldots, X_{n-1})^T$, $\mathbf{y} = (Y_1, \ldots, Y_{n-1})^T$. Note that equation (46) is non-linear because the tridiagonal matrix A depends on $\mathbf{x}$ and $\mathbf{y}$. We have found that (46) can be solved by the fix point iteration

$$A(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\mathbf{x}^{(k+1)} = \mathbf{r}^{(1)}, \quad A(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})\mathbf{y}^{(k+1)} = \mathbf{r}^{(2)}, \quad k = 0, 1, \ldots.$$

This iteration usually converges quickly, at least for cases where the number of layers is moderate.

By solving the above minimization problem, we can calculate the minimum travel time from the source to receiver 'r', which we denote by $\hat{T}_r$. Clearly, it is a function of the source location, and similar to the case with a homogeneous material, we have $\hat{T}_r = \hat{T}_r(x_s, y_s, z_s)$.

As before, we can estimate the source location by solving (44) for the unknowns (x_s, y_s, z_s, t_s). We assume the more than four receivers are available ($R > 4$), which makes (44) an overdetermined system. The Jacobian of (44) is needed by the Gauss-Newton method. Here we approximate the derivative of I_r with respect to the source location by numerical differentiation, i.e.,

$$\frac{\partial I_r}{\partial x_s} \approx \frac{\hat{T}_r(x_s + \tilde{h}, y_s, z_s) - \hat{T}_r(x_s, y_s, z_s)}{\tilde{h}},$$

for a small fixed number $\tilde{h}$. The y_s- and z_s-derivatives are computed similarly. Thus when solving (44) by the Gauss-Newton method, (46) must be solved four times per iteration.

5.2 Estimating the source frequency

It has turned out to be difficult to automatically estimate the source frequency, ω_0. For this reason, we require an initial guess for ω_0 to be provided by the user. However, in practice this might not be a serious problem, because in realistic applications the observed ground motions must be filtered in time to remove waves that can not be resolved on the computational grid. This is a preprocessing step that is performed before the optimization is started. The corner frequency of the filter is then related to the effective source frequency.

5.3 Initial estimate for the moment tensor

Once initial estimates for the source location, frequency, and starting time have been established, we can use the linearity of the elastic wave equation to estimate the matrix $\mathcal{M}$ in the source term (4). Let $\mathbf{u}^{(xx)}$, $\mathbf{u}^{(xy)}$, $\mathbf{u}^{(xz)}$, $\mathbf{u}^{(yy)}$, $\mathbf{u}^{(yz)}$, and $\mathbf{u}^{(zz)}$ denote solutions of the elastic wave equation with the matrix $\mathcal{M}$ set to

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

respectively. The solution for a general $\mathcal{M}$ is then obtained as the linear combination

$$\mathbf{w}(m_{xx}, m_{xy}, m_{xz}, m_{yy}, m_{yz}, m_{zz}) := \\ m_{xx}\mathbf{u}^{(xx)} + m_{xy}\mathbf{u}^{(xy)} + m_{xz}\mathbf{u}^{(xz)} + m_{yy}\mathbf{u}^{(yy)} + m_{yz}\mathbf{u}^{(yz)} + m_{zz}\mathbf{u}^{(zz)}.$$

The elements of $\mathcal{M}$ are determined by minimizing the wave form misfit

$$\mathcal{X} = \frac{1}{2}\sum_{r=1}^{R}\sum_{n=0}^{M-1} s(t_n)\left|\mathbf{w}_{\mathbf{i}_r}^n(m_{xx}, m_{xy}, m_{xz}, m_{yy}, m_{yz}, m_{zz}) - \mathbf{d}_{\mathbf{i}_r}^n\right|^2. \tag{47}$$

Because $\mathbf{w}$ is linear in m_{ij}, $\mathcal{X}$ is a quadratic function of m_{ij}. Its minimum can be computed directly by solving the 6×6 linear system $\partial\mathcal{X}/\partial m_{ij} = 0$.

6 Numerical experiments

To verify our implementation and gain understanding of the performance of the suggested approach, it is convenient to conduct numerical experiments on synthetic data. To generate the synthetic data, we first solve the discretized elastic wave equation with given source parameters, and use the resulting motions at the receiver stations as 'measured' data. In this way, the exact solution is known, and we can easily evaluate the convergence properties of the minimization algorithm.

A standard test problem for elastic wave modeling is the layer over half space problem called LOH.1, see [1]. In this test, a point moment tensor forcing with a Gaussian time function is applied in a layered isotropic elastic material. The Gaussian time function,

$$g(t; t_0, \omega_0) = \frac{\omega_0}{\sqrt{2\pi}} e^{-\omega_0^2(t-t_0)^2/2},$$

is parametrized by the frequency ω_0 and the center time t_0. The material velocities are $c_p = 4000$ and $c_s = 2000$ in a top layer extending over $0 \le z \le 1000$, with $c_p = 6000$ and $c_s = 3464$ in the half-space $z > 1000$. The densities are $\rho = 2600$ in the top layer and $\rho = 2700$ in the half-space. The computational domain in the standard LOH.1 problem is a box of size $30000 \times 30000 \times 17000$. In order to make the computations run faster, we reduce the depth of the computational domain from 17000 to 8500, but keep the thickness of the top layer unchanged. All computations use the grid spacing $h = 120$ and the elastic wave equation is integrated to time $T = 9$. The spatial grid has 4.5 million points.

In the following numerical experiments, the synthetic data was generated with the source parameter vector $\mathbf{p}_*$, with components $x_s^* = y_s^* = 15000$, $z_s^* = 2000$, $m_{xy}^* = 10^{18}$, $m_{xx}^* = m_{xz}^* = m_{yy}^* = m_{yz}^* = m_{zz}^* = 0$, $t_0^* = 1.45$, and $\omega_0^* = 6.0$. Note that we have reduced the value of the frequency parameter ω_0 to allow for a coarser grid spacing than what normally is used when solving the LOH.1 problem.

The solution is recorded at 25 receiver stations placed on a coarse 5×5 grid, at

$$(x_r, y_r, z_r) = (9000 + 3000(j-1), 9000 + 3000(k-1), 0), \quad r = j + 5(k-1),$$

for $1 \le j \le 5$, and $1 \le k \le 5$.

The center time t_0 in the Gaussian time function follows from the source start time t_s as $t_0 = t_s + t_\delta$, where the half-duration t_δ satisfies

$$\eta = \frac{\omega_0}{\sqrt{2\pi}} e^{-(\omega_0 t_\delta)^2/2} \quad \Rightarrow \quad t_\delta = \frac{1}{\omega_0}\sqrt{-2\log((\eta\sqrt{2\pi})/\omega_0)}.$$

Here, $0 < \eta \ll 1$ is the same constant that is used in (45) for estimating the first arrival time.

The automated initial guess described in Section 5, with the user choice $\omega_0 = 6.3$, resulted in the estimated source position $x_s = 14980.8$, $y_s = 15039.5$, $z_s = 2352.73$, with moment tensor components $m_{xx} = 3.754 \times 10^{14}$, $m_{xy} = 9.622 \times 10^{17}$, $m_{xz} = -8.937 \times 10^{15}$, $m_{yy} = 3.758 \times 10^{14}$, $m_{yz} = 4.351 \times 10^{15}$, $m_{zz} = -5.313 \times 10^{12}$, and $t_0 = 1.358$. The estimate for the location and start time was obtained with the ray tracing algorithm applied to the above layer over half-space material model. It is a sufficiently good approximation to make the Fletcher-Reeves algorithm converge to the global minimum.

In our implementation of the source inversion algorithm, the user is given the choice to enter an initial parameter guess, or have the solver estimate it automatically. The computational cost of the automatic estimate is somewhat high because the moment tensor components are calculated by solving six elastic wave equations, see Section 5.3. Hence, computational time can be saved if a fairly accurate approximation of the source parameters is known in some other way. In some of the numerical experiments below, we use the following initial parameter values:

$$\begin{aligned} x_s = 16000, \;\; y_s = 14000, \;\; z_s = 2200, \;\; m_{xy} = 1.2 \cdot 10^{18}, \\ m_{xx} = m_{xz} = m_{yy} = m_{yz} = m_{zz} = 0, \;\; t_0 = 1.54, \;\; \omega_0 = 6.3. \end{aligned} \tag{48}$$

As long as the initial parameter guess is reasonably close to the global minimum, our practical experience is that the number of iterations required to reach convergence is not sensitive to this choice.

6.1 Choosing the preconditioner

The sizes of the parameters in the source estimation problem span many orders of magnitude. In SI-units, $\mathbf{x}_s$ is of the order $\mathcal{O}(10^4)$, the moment tensor components m_{xx}, $m_{xy}, \ldots$ are of the order $\mathcal{O}(10^{15}) - \mathcal{O}(10^{18})$. The parameters t_0 and ω_0 are both between $\mathcal{O}(1)$ and $\mathcal{O}(10)$. Because there is such a large difference in size between the smallest and largest parameter values, the original minimization problem is very poorly scaled and the condition number of the Hessian is a very large.

For optimal convergence of the Fletcher-Reeves algorithm, the parameters should be scaled such that the Hessian at the solution, $H_* := H(\mathbf{p}_*)$, has condition number one. The change of parameters $\hat{\mathbf{p}} = S\mathbf{p}$ gives the scaled Hessian $\hat{H}_s = (S^{-1})^T H_s S^{-1}$, i.e., the scaling corresponding to $\hat{H}_s = I$ satisfies

$$S^T S = H_*. \tag{49}$$

Hence, S could be computed by a Cholesky factorization of H_*. However, H_* is in general not computable because it requires the solution of the minimization problem to be known. Instead we can use a Cholesky factorization of the Hessian at the initial parameter guess. Since this scaling is not optimal and the implementation of the scaled algorithm is more straight forward when S is diagonal, we restrict us to this case. As we shall see, a significant reduction of the condition number of the Hessian can still be achieved. When S is diagonal, (49) can not be satisfied exactly. Instead we minimize the residual, $||H_* - S^2||_F$, which gives $S_{jj} = \sqrt{H_{jj}}$, $j = 1, \ldots, P$, i.e., the scaling matrix should equal the square root of the diagonal of the Hessian. The Hessian at the minimum is positive definite, which implies that the diagonal elements of H_* are positive. Since H_* is not known until the minimization problem has been solved, we define S as the square root of the diagonal of the Hessian evaluated at the initial guess. One difficulty with this definition is that there is no guarantee that the Hessian is positive definite at the initial guess. If there are negative diagonal elements in the Hessian, we instead use the square root of the diagonal elements of the matrix H_1, see (19), evaluated at the initial guess. It is obvious from (19) that H_1 always has non-negative diagonal elements. This approach has turned out to work well in practice.

The computation of the Hessian, which is described in Section 3.1, requires the elastic wave equation to be solved 11 times. However, this computation only needs to be done once, before the Fletcher-Reeves iteration starts.

6.2 Condition number of the scaled Hessian

Table 1 shows the influence of different scaling matrices for the LOH.1 source inversion problem. The bottom row shows the condition number of the Hessian at the exact minimum, scaled by the given S. Here, the condition number was computed by the Matlab function `cond`. The diagonal variable transformation $\hat{\mathbf{p}} = S\mathbf{p}$ implies that the inverse of the diagonal elements of the scaling matrix correspond to reference sizes of the parameters. However, only their relative sizes matter because multiplying S by a constant factor does not change the condition number of the scaled Hessian.

The unscaled Hessian has condition number $cond(H_*) = 4.73 \cdot 10^{38}$. The second column of Table 1 shows the scaling obtained as the square root of the diagonal elements of the Hessian, evaluated at the initial parameter guess (48). Not all diagonal elements of the Hessian are positive at this point in parameter space. For this reason, only the first part of the Hessian, H_1, as described in Section 6.1 was used in this scaling. It is interesting to note that the scaling obtained from the square root of the diagonal of the Hessian at the minimum, shown in column three, leads to a slightly larger condition number. As was mentioned above, the Hessian at the minimum is in general not computable because this scaling assumes that the solution of the minimization problem is known. The fourth column, labeled ref. sizes 1, shows the scaling based on estimated sizes of the parameters. These numbers are based on the size of the domain, which is in the 10's of kilometers, and the fact that we know that t_0 is of order $\mathcal{O}(1)$, $\omega_0 = \mathcal{O}(10)$, and the moment tensor components are of the order $\mathcal{O}(10^{18})$. Table 1 shows that the scalings based on the Hessian give significantly smaller condition numbers compared with the unscaled case and

	Hes., guess	Hes., exact	Ref. sizes 1	Ref. sizes 2	Ref. sizes 3
$1/s_{1,1}$ (x_s)	10.8	11.6	$1.00 \cdot 10^4$	$1.00 \cdot 10^3$	$5.00 \cdot 10^3$
$1/s_{2,2}$ (y_s)	10.8	11.6	$1.00 \cdot 10^4$	$1.00 \cdot 10^3$	$5.00 \cdot 10^3$
$1/s_{3,3}$ (z_s)	12.2	20.6	$1.00 \cdot 10^4$	$1.00 \cdot 10^3$	$5.00 \cdot 10^3$
$1/s_{4,4}$ (m_{xx})	$2.68 \cdot 10^{16}$	$2.70 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{5,5}$ (m_{xy})	$1.68 \cdot 10^{16}$	$1.65 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{6,6}$ (m_{xz})	$1.39 \cdot 10^{16}$	$1.29 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{7,7}$ (m_{yy})	$2.67 \cdot 10^{16}$	$2.70 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{8,8}$ (m_{yz})	$1.39 \cdot 10^{16}$	$1.29 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{9,9}$ (m_{zz})	$2.28 \cdot 10^{16}$	$1.85 \cdot 10^{16}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$	$1.00 \cdot 10^{18}$
$1/s_{10,10}$ (t_0)	$2.13 \cdot 10^{-3}$	$2.55 \cdot 10^{-3}$	1.00	0.10	0.50
$1/s_{11,11}$ (ω_0)	$5.65 \cdot 10^{-2}$	$6.24 \cdot 10^{-2}$	10.0	1.00	5.00
$cond(S^{-1}H_*S^{-1})$	28.1	31.8	$6.18 \cdot 10^3$	80.4	$1.55 \cdot 10^3$

Table 1: *Scaling factors and their influence on the condition number of the scaled Hessian. The condition number of the unscaled Hessian is* $2.77 \cdot 10^{38}$

ref. sizes 1. After inspecting the Hessian based scalings, we modified the reference size scaling to be 10^3 for the length scale, 10^{18} for the moment scale, 0.1 for the time scale, and 1 for the frequency scale. This scaling, given as ref. sizes 2 in Table 1, gave the much improved condition number 80.4. The last column of Table 1 shows a scaling that is in between ref. sizes 1 and ref. sizes 2, labeled ref. sizes 3. It lead to a condition number close to that of ref. sizes 1, indicating how sensitive the condition number is to the scaling matrix. Hence, even though it is possible to design a favorable parameter scaling by order of magnitude arguments, it is very difficult to find the optimal values. We conclude that using the Hessian for scaling the parameters is a much more reliable way to achieve a small condition number.

6.3 Convergence rates

Figure 2 shows convergence properties of the Fletcher-Reeves algorithm for different scaling matrices. These computations were run for up to a maximum of 10 restarts ($m = 10$ in Algorithm 3), with each restart cycle consisting of $P = 11$ inner iterations. The iteration is terminated when the maximum norm of the scaled gradient of the misfit becomes smaller than 10^{-12}, or when the maximum number of iterations is reached. The magenta curve shows the convergence history when S is taken as the square root of the diagonal elements of H_*. Using the square root of the diagonal elements of the Hessian at the initial guess, shown by the red curve in Figure 2, makes the iteration converge in almost the same number of iterations. The results plotted in cyan and blue colors were

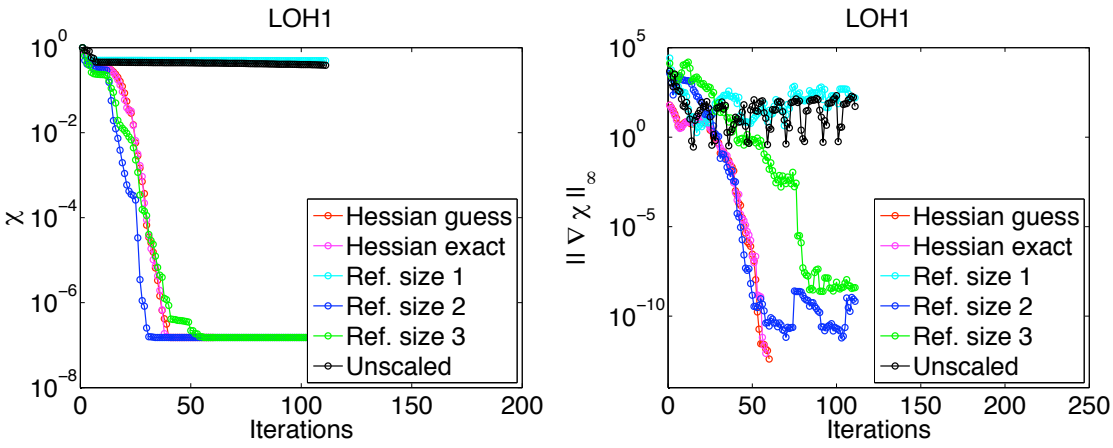

Figure 2: Convergence of the misfit (left) and the maximum norm of the scaled gradient (right) for the different scalings given in Table 1.

obtained with scalings corresponding to the cases "ref. sizes 1" and "ref. sizes 2" in Table 1. The improvement in convergence rate when switching from "ref. sizes 1" to "ref. sizes 2" is remarkable, and prompted us to try the intermediate scaling labeled "ref. sizes 3", shown in green color.

The Hessian based scalings always perform well, and the solution is obtained in 40-50 iterations. The reference size scalings can be made almost as efficient, but are very sensitive to the exact values in the scaling matrix. The convergence rate of the unscaled method, shown in black, is very slow and should not be used in practical computations. Figure 3 displays the evolution of the source parameters during the iterations, when the scaling is computed from the Hessian at the initial guess. The left figure shows the source position x_s, y_s, z_s vs. the number of iterations. Here, y_s is offset by 1000 to distinguish it from x_s and z_s is offset by 10000 to make it fit into the same plot. The circles to the right of the curves indicate the exact value of the parameter. Similarly, the middle subplot of Figure 3 show the evolution of the six components of the matrix $\mathcal{M}$, and the right subplot shows the time shift and the frequency with an offset. It is clear that already after about 33 iterations (2 restarts with the Fletcher-Reeves algorithm), all parameter values have converged in "picture" norm.

Figure 4 gives another illustration of the convergence process. The ground motions at the receiver stations, $(x, y, z) = (9000, 21000, 0)$ and $(x, y, z) = (9000, 12000, 0)$ are displayed as functions of time. The red curves were computed by solving the elastic wave equation with the initial source parameter values (48). The curves plotted in black are the motions due to solving the elastic wave equation with source parameter values $\mathbf{p}_*$, corresponding to the solution of the minimization problem. Since this example uses synthetic data, the solution of the minimization problem results in motions that are identical to the observations, modulus roundoff errors.

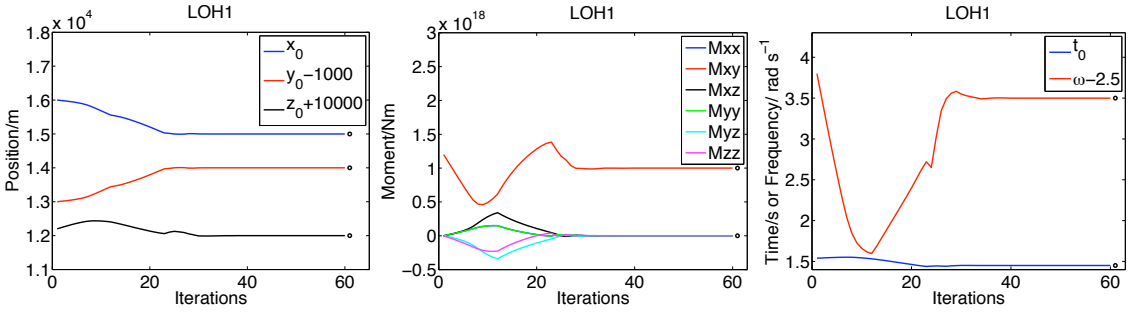

Figure 3: Convergence of source parameters. Location (left), moment tensor components (middle), and time shift and frequency (right).

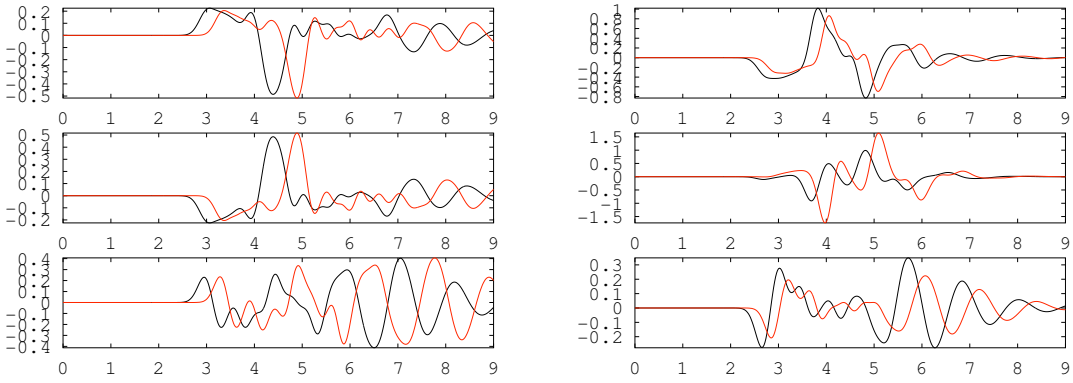

Figure 4: The (u, v, w) components of the motions vs. time, at the receivers $(x, y, z) = (9000, 21000, 0)$ (left) and $(x, y, z) = (9000, 12000, 0)$ (right). Curves in red were generated with the initial source parameter values (48) and curves in black correspond to $\mathbf{p}_*$, i.e., the solution of the minimization problem.

6.4 Computational cost

In each iteration of Fletcher-Reeves algorithm, the computation of the misfit and its gradient can be obtained by solving one elastic and one adjoint wave equation. The computation of the step length (14) adds one more elastic wave equation to be solved, and the test for acceptance of the initial step length in the line search algorithm requires the solution of yet another elastic wave equation. Hence, each iteration requires a minimum of three elastic wave equations and one adjoint wave equation to be solved. This count holds if the line search algorithm accepts the initial step length, which usually is the case. When the line search algorithm needs to perform step length reduction (backtracking), each reduction incurs one additional solve of an elastic wave equation.

Figure 3 shows that a highly accurate solution can be obtained after about 40 iterations. Since the adjoint wave equation needs the same amount of computational work as solving the elastic wave equation, this corresponds to approximately 160 solutions of the elastic wave equation. The total computation times for the 60 iterations shown in Figure 3 was about 15 minutes using 256 cores of a Linux cluster of Intel Xenon processors.

7 Conclusions

We have presented an algorithm for estimating the seismic source parameters from recorded time dependent motions at a number of receiver stations. The solution of this inverse problem is obtained by minimizing the full waveform misfit using a non-linear conjugate gradient method. The key features of the proposed technique are an adjoint discretization of the fourth order accurate method in [16], a source discretization that leads to a twice continuously differentiable misfit function, and a parameter scaling that makes the minimization problem well conditioned. Numerical experiments with the LOH.1 problem indicate good convergence properties of the proposed algorithm.

Several practical problems must be solved before we can apply our approach to estimate source parameters in realistic seismic events. Seismographic recordings must be deconvolved and band-pass filtered to compensate for instrument response characteristics. Furthermore, additional filtering of the recordings is often needed to remove frequencies that can not be resolved on the computational grid. We are currently working on incorporating the filters into the source estimation problem. Because the material properties do not depend on time, the elastic wave equation describes a linear time-invariant system. Filtering the solution is therefore equivalent to using a filtered source time function. The resulting minimization problem will therefore depend on only 10 parameters, since the frequency parameter is fixed by the filter.

Another interesting extension of the current approach is the inverse problem for estimating the material wave speeds and density. One additional difficulty with this problem is to find a suitable parametrization of the material. To limit the dimensionality of parameter space, we expect that some degree of smoothness must be imposed on the material model, perhaps by using piecewise smooth basis functions to represent the material properties. Preliminary computations in two space dimensions with a very simple material parametrization are showing encouraging results.

A Proof of Theorem 1

We start by expanding the predictor into the corrector in Algorithm 1. Then rewrite the resulting expression as

$$\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^{n}+\mathbf{u}^{n-1}}{\Delta_t^2}=\mathbf{L}_h(\mathbf{u}^n)+\mathbf{F}(t_n)+\frac{\Delta_t^2}{12}\big(\mathbf{L}_h(\mathbf{v}^n)+\mathbf{F}_{tt}(t_n)\big)+\mathbf{S}_G(\mathbf{u}^n-\mathbf{u}^{n-1}). \quad (50)$$

Next, take the scalar product between (50) and $\boldsymbol{\kappa}^n$, and sum over all time steps,

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^{n}+\mathbf{u}^{n-1}}{\Delta_t^2})_h=\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{u}^n))_h+\sum_{n=0}^{M-1}\left(\boldsymbol{\kappa}^n,\mathbf{F}(t_n)+\frac{\Delta_t^2}{12}\mathbf{F}_{tt}(t_n)\right)_h$$
$$+\frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{v}^n))_h+\sum_{n=0}^{M-1}\big(\boldsymbol{\kappa}^n,\mathbf{S}_G(\mathbf{u}^n-\mathbf{u}^{n-1})\big)_h. \quad (51)$$

The sum on the left hand side of (51) can be rewritten as

$$\sum_{n=0}^{M-1}\left(\boldsymbol{\kappa}^n,\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^{n}+\mathbf{u}^{n-1}}{\Delta_t^2}\right)_h=\sum_{n=0}^{M-1}\left(\rho\frac{\boldsymbol{\kappa}^{n+1}-2\boldsymbol{\kappa}^{n}+\boldsymbol{\kappa}^{n-1}}{\Delta_t^2},\mathbf{u}^n\right)_h$$
$$+\frac{1}{\Delta_t^2}\left((\rho\mathbf{u}^{-1},\boldsymbol{\kappa}^0)_h-(\rho\mathbf{u}^0,\boldsymbol{\kappa}^{-1})_h+(\rho\mathbf{u}^M,\boldsymbol{\kappa}^{M-1})_h-(\rho\mathbf{u}^{M-1},\boldsymbol{\kappa}^M)_h\right), \quad (52)$$

where the initial data $\mathbf{u}^0=\mathbf{u}^{-1}=\boldsymbol{\kappa}^M=\boldsymbol{\kappa}^{M-1}=\mathbf{0}$ make

$$\sum_{n=0}^{M-1}\left(\boldsymbol{\kappa}^n,\rho\frac{\mathbf{u}^{n+1}-2\mathbf{u}^{n}+\mathbf{u}^{n-1}}{\Delta_t^2}\right)_h=\sum_{n=0}^{M-1}\left(\rho\frac{\boldsymbol{\kappa}^{n+1}-2\boldsymbol{\kappa}^{n}+\boldsymbol{\kappa}^{n-1}}{\Delta_t^2},\mathbf{u}^n\right)_h.$$

The first sum on the right hand side of (51) is treated by the self-adjoint property,

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{u}^n))_h=\sum_{n=0}^{M-1}(\mathbf{L}_h(\boldsymbol{\kappa}^n),\mathbf{u}^n)_h.$$

The second last sum of (51) can be rewritten

$$(\boldsymbol{\kappa}^n,\mathbf{L}_h(\mathbf{v}^n))_h=(\mathbf{L}_h(\boldsymbol{\kappa}^n),\mathbf{v}^n)_h=\left(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}(\mathbf{L}_h(\mathbf{u}^n)+\mathbf{F}(t_n))\right)_h$$
$$=\left(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}\mathbf{L}_h(\mathbf{u}^n)\right)_h+\left(\mathbf{L}_h(\boldsymbol{\kappa}^n),\frac{1}{\rho}\mathbf{F}(t_n)\right)_h$$
$$=\left(\mathbf{L}_h\left(\frac{1}{\rho}\mathbf{L}_h(\boldsymbol{\kappa}^n)\right),\mathbf{u}^n\right)_h+\left(\frac{1}{\rho}\mathbf{L}_h(\boldsymbol{\kappa}^n),\mathbf{F}(t_n)\right)_h=(\mathbf{L}_h(\boldsymbol{\zeta}^n),\mathbf{u}^n)_h+(\boldsymbol{\zeta}^n,\mathbf{F}(t_n))_h. \quad (53)$$

The supergrid damping term can be written

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n, \mathbf{S}_G(\mathbf{u}^n - \mathbf{u}^{n-1}))_h = \sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n, \mathbf{S}_G(\mathbf{u}^n))_h \\ - \sum_{n=0}^{M-1}(\boldsymbol{\kappa}^{n+1}, \mathbf{S}_G(\mathbf{u}^n))_h - (\boldsymbol{\kappa}^0, \mathbf{S}_G(\mathbf{u}^{-1}))_h + (\boldsymbol{\kappa}^M, \mathbf{S}_G(\mathbf{u}^{M-1}))_h. \quad (54)$$

The boundary terms are zero because of the initial data $\mathbf{u}^{-1} = \boldsymbol{\kappa}^M = 0$, and we use the symmetry (9) to obtain

$$\sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n, \mathbf{S}_G(\mathbf{u}^n - \mathbf{u}^{n-1}))_h = \sum_{n=0}^{M-1}(\boldsymbol{\kappa}^n - \boldsymbol{\kappa}^{n+1}, \mathbf{S}_G(\mathbf{u}^n))_h = -\sum_{n=0}^{M-1}(\mathbf{S}_G(\boldsymbol{\kappa}^{n+1} - \boldsymbol{\kappa}^n), \mathbf{u}^n)_h$$

Collecting terms gives that (51) is equivalent to

$$\sum_{n=0}^{M-1}\left(\rho\frac{\boldsymbol{\kappa}^{n+1} - 2\boldsymbol{\kappa}^n + \boldsymbol{\kappa}^{n-1}}{\Delta_t^2}, \mathbf{u}^n\right)_h = \sum_{n=0}^{M-1}(\mathbf{L}_h(\boldsymbol{\kappa}^n), \mathbf{u}^n)_h + \sum_{n=0}^{M-1}\left(\boldsymbol{\kappa}^n, \mathbf{F}(t_n) + \frac{\Delta_t^2}{12}\mathbf{F}_{tt}(t_n)\right)_h \\ + \frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}(\mathbf{L}_h(\boldsymbol{\zeta}^n), \mathbf{u}^n)_h + \frac{\Delta_t^2}{12}\sum_{n=0}^{M-1}(\boldsymbol{\zeta}^n, \mathbf{F}(t_n))_h - \sum_{n=0}^{M-1}(\mathbf{S}_G(\boldsymbol{\kappa}^{n+1} - \boldsymbol{\kappa}^n), \mathbf{u}^n)_h. \quad (55)$$

Expanding the predictor (11) into the corrector (12) gives

$$\rho\frac{\boldsymbol{\kappa}^{n+1} - 2\boldsymbol{\kappa}^n + \boldsymbol{\kappa}^{n-1}}{\Delta_t^2} = \mathbf{L}_h(\boldsymbol{\kappa}^n) + \mathbf{G}(t_n) + \frac{\Delta_t^2}{12}\mathbf{L}_h(\boldsymbol{\zeta}^n) - \mathbf{S}_G(\boldsymbol{\kappa}^{n+1} - \boldsymbol{\kappa}^n). \quad (56)$$

Identity (13) of Theorem 1 is obtained by inserting (56) into the left hand side of (55).

References

[1] S. M. Day, J. Bielak, D. Dreger, S. Larsen, R. Graves, A. Pitarka, and K. B. Olsen. Test of 3D elastodynamic codes: Lifelines project task 1A01. Technical report, Pacific Earthquake Engineering Center, 2001.

[2] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.

[3] Z. Duputel, L. Rivera, Y. Fukahata, and H. Kanamori. Uncertainty estimations for seismic source inversions. *Geophys. J. Int.*, 190:1243–1256, 2012.

[4] L. Eisner and R. W. Clayton. A reciprocity method for multiple-source simulations. *Bull. Seism. Soc. Am.*, 91:553–560, 2001.

[5] M. Fink. Time reversal in acoustics. *Contemp. Phys.*, 37:95–109, 1996.

[6] E. Haber. Quasi-Newton methods for large-scale electromagnetic inverse problems. *Inverse Problems*, 21:305–323, 2005.

[7] Y. H. Kim, Q. Liu, and J. Tromp. Adjoint centroid-moment tensor inversions. *Geophys. J. Int.*, 186(1):264–278, 2011.

[8] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

[9] C. Larmat, J.-P. Montagner, M. Fink, Y. Capdeville, A. Tourin, and E. Clévédé. Time-reversal imaging of seismic sources and application to the great Sumatra earthquake. *Geophys. Res. Lett.*, 33:L19312, 2006.

[10] P. Liu, S. Hartzell, and W. Stephenson. Non-linear multiparameter inversion using a hybrid global search algorithm: applications in reflection seismology. *Geophys. J. Int.*, 122:991–1000, 1995.

[11] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, 1973.

[12] S. Nilsson, N. A. Petersson, B. Sjögreen, and H.-O. Kreiss. Stable difference approximations for the elastic wave equation in second order formulation. *SIAM J. Numer. Anal.*, 45:1902–1936, 2007.

[13] N. A. Petersson and B. Sjögreen. Stable grid refinement and singular source discretization for seismic wave simulations. *Comm. Comput. Phys.*, 8(5):1074–1110, 2010.

[14] N. A. Petersson and B. Sjögreen. A fourth order supergrid damping layer for the elastic wave equation. Technical report, Lawrence Livermore National Laboratory, 2012. To be submitted.

[15] R. E. Plessix and W. A. Mulder. Resistivity imaging with controlled-source electromagnetic data: depth and data weighting. *Inverse Problems*, 24:034012, 2008.

[16] B. Sjögreen and N. A. Petersson. A fourth order accurate finite difference scheme for the elastic wave equation in second order formulation. *J. Scient. Comput.*, 52:17–48, 2012.

[17] C. Tape, Q. Liu, and J. Tromp. Finite-frequency tomography using adjoint methods - Methodology and examples using membrane surface waves. *Geophys. J. Int.*, 168:1105–1129, 2007.

[18] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation.* SIAM, 2005.

[19] J. Tromp, C. Tape, and Q. Liu. Seismic tomography, adjoint mehtods, time reversal and banana-doughnut kernels. *Geophys. J. Int.*, 160:195–216, 2005.