Computers & Geosciences 151 (2021) 104751

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

A general approach to seismic inversion with automatic differentiation

Weiqiang Zhu [a,*,1], Kailai Xu [b,1], Eric Darve [b,c], Gregory C. Beroza [a]

[a] Department of Geophysics, Stanford University, Stanford, CA, 94305, USA
[b] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, 94305, USA
[c] Mechanical Engineering, Stanford University, Stanford, CA, 94305, USA

ARTICLE INFO

Keywords:
Computational methods
Algorithms
Inverse problems
Parallel and highperformance computing
Software engineering

ABSTRACT

Imaging Earth structure or seismic sources from seismic data involves minimizing a target misfit function, and is commonly solved through gradient-based optimization. The adjoint-state method has been developed to compute the gradient efficiently; however, its implementation can be time-consuming and difficult. We develop a general seismic inversion framework to calculate gradients using reverse-mode automatic differentiation. The central idea is that adjoint-state methods and reverse-mode automatic differentiation are mathematically equivalent. The mapping between numerical PDE simulation and deep learning allows us to build a seismic inverse modeling library, ADSeismic, based on deep learning frameworks, which supports high performance reverse-mode automatic differentiation on CPUs and GPUs. We demonstrate the performance of ADSeismic on inverse problems related to velocity model estimation, rupture imaging, earthquake location, and source time function retrieval. ADSeismic has the potential to solve a wide variety of inverse modeling applications within a unified framework.

1. Introduction

Inverse modeling is used in seismology to recover physical parameters such as earthquake location, magnitude, and Earth's interior structure. Such inverse problems are usually solved by minimizing a misfit function that measures the discrepancy between predictions and observations. Derivative-based optimization requires calculation of the gradient of the misfit function with respect to the physical parameters. The adjoint-state method is a commonly used technique for computing the gradient efficiently (Plessix, 2006). This method solves an adjoint linear system, which involves solutions of the forward problem usually implemented in partial differential equations (PDEs). The drawback of the adjoint-state method is that the derivation and implementation can be very challenging. For PDE-constrained optimization, the adjoint equation to calculate the partial derivatives of specific physical equation and its PDE discretization with respect to the model parameters needs to be derived on a case-by-case basis for different systems and physical parameters (Bradley, 2013). Although many frameworks exist for specific inverse modeling applications (Cockett et al., 2015; Rücker et al., 2017), to our knowledge general frameworks that can estimate physical parameters without case-by-case gradient derivation and implementation are lacking.

Automatic differentiation (AD) (Baydin et al., 2017; Paszke et al., 2017), where the gradients are computed automatically based on the computational graph of the forward simulation, provides an alternative approach. In AD, a computational graph of the forward simulation keeps track of arithmetic operational dependencies, stores intermediate results, and computes the gradient using the chain rule. AD has been the dominant approach for training deep neural networks, which is known as "back-propagation" in the deep learning community (Rumelhart et al., 1986). Both deep neural networks and PDE simulations can be viewed as a series of linear or nonlinear operators (Hughes et al., 2019). Moreover, reverse-mode automatic differentiation has been shown to be mathematically equivalent to the adjoint-state method (LeCun et al., 1988; Li et al., 2019; Ren et al., 2020). This correspondence allows us to develop a flexible and general seismic inversion framework, ADSeismic, based on current deep learning frameworks such as TensorFlow (Abadi et al., 2016). We note that AD has already been applied to velocity estimation in exploration seismology (Sambridge et al., 2007; Cao and Liao, 2015; Vlasenko et al., 2016; Richardson, 2018). Compared to

* Corresponding author.
E-mail addresses: zhuwq@stanford.edu (W. Zhu), kailaix@stanford.edu (K. Xu), darve@stanford.edu (E. Darve), beroza@stanford.edu (G.C. Beroza).

[1] The two authors contributed equally to this paper. Weiqiang Zhu and Kailai Xu developed the method, performed the inversion experiments, and co-wrote the paper. Eric Darve and Gregory C. Beroza supervised the research and edited the manuscript. All authors reviewed the final manuscript. The package ADSeismic.jl is available at https://github.com/kailaix/ADSeismic.jl.

https://doi.org/10.1016/j.cageo.2021.104751
Received 11 March 2020; Received in revised form 23 February 2021; Accepted 5 March 2021
Available online 26 March 2021
0098-3004/

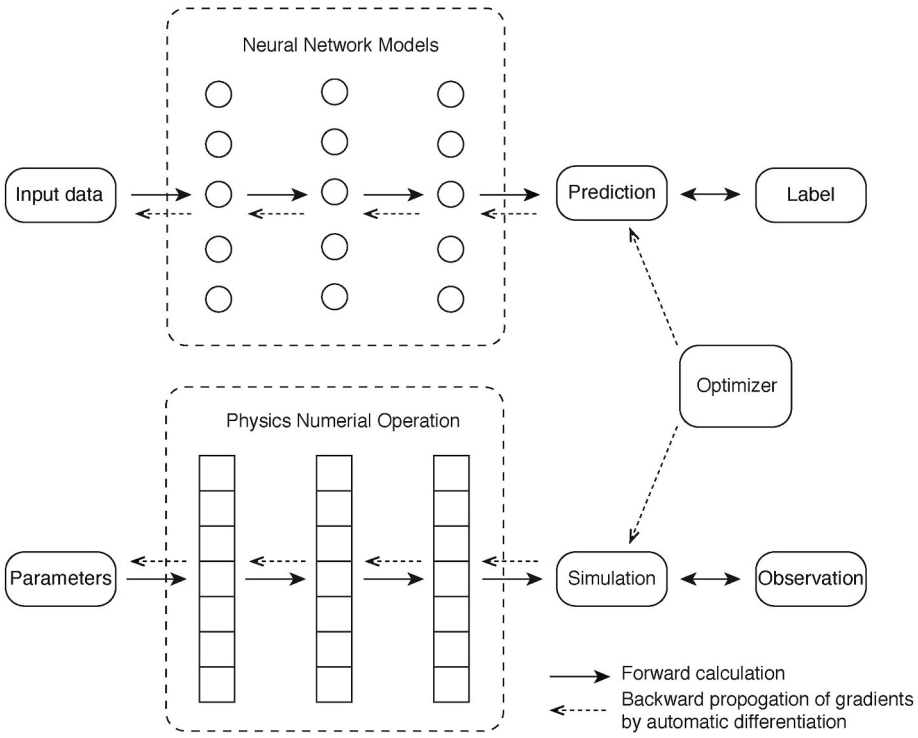

Fig. 1. The similarity between neural networks training and PDE-based physical simulation and optimization.

existing open-source seismic inversion software, such as IWAVE (Symes, 2015), JUDI4Flux.jl (Witte et al., 2020), and Devito (Louboutin et al., 2019), ADSeismic has some distinct features for seismic inversion: it allows for flexibly experimenting with different inversion targets, leverages specialized hardware designed for deep learning, and executes numerical simulations on heterogeneous computing platforms. ADseismic provides a high performance environment with easily accessible gradients on CPUs and GPUs. It also supports parallel computing based on MPI (Message Passing Interface) (Clarke et al., 1994) for solving inversion tasks on large-scale super computers.

We demonstrate several applications, including velocity estimation, fault rupture imaging, earthquake location, and source time function retrieval. AD yields the same inversion results as adjoint-state methods. The advantage is that while we need to derive and implement a specific gradient for each parameter in different cases with the adjoint-state method, these inversion problems can be solved similarly by specifying a different parameter as the inversion target with ADSeismic because the gradient for each parameter is automatically calculated by AD. Moreover, both the forward simulation and inversion can be accelerated by GPUs and large-scale CPU clusters. Since deep learning hardware and frameworks are continuously improving, ADSeismic provides seismic inverse modeling with increasingly powerful automatic differentiation techniques for a wide range of applications.

2. Method

2.1. Automatic differentiation

Automatic differentiation (AD) is a general and efficient method to compute gradients based on the chain rule. By tracing the forward-pass computation, the gradient at the final step propagates back to each operator and parameter in a computational graph. AD is mainly used for training neural network models that consist of a sequence of linear transforms and non-linear activation functions. AD calculates the gradients of every variable by propagating the gradients back from the loss function to the trainable parameters. These gradients are then used in a gradient-based optimizer, such as the gradient descent (GD) method, to update the parameters and minimize the differences between the model predictions and the ground-truth labels. Numerical simulations based on PDEs are similar to neural network models in that they are both sequences of linear/non-linear transformations (Fig. 1). For example, the Finite-Difference Time-Domain (FDTD) method (Yee, 1966), applies a finite difference operator to consecutive time steps to solve time-dependent PDEs (Hughes et al., 2019). In seismic problems, we specify parameters, such as wave velocity, source location, or source time functions, in forward simulations to generate predicted seismic signals. The gradients of the observational differences over these parameters can be computed automatically in ADSeismic and thus used in a gradient-based optimizer in the same way as when training neural networks.

2.2. Relationship to the adjoint method

The adjoint-state method is an efficient technique for computing the gradient of the misfit function with respect to the physical parameters of interest. For example, the adjoint-state method is commonly used to compute the gradient in full-waveform inversion (Plessix, 2006). To clarify the connection between the adjoint-state method and reverse-mode automatic differentiation, we provide a derivation based on Lagrange multipliers.

Consider the explicit discretization of the wave equation, which can be written as

$$\begin{aligned} U_1 &= A(\theta)U_0 + F_0 \\ U_2 &= A(\theta)U_1 + F_1 \\ &\vdots \\ U_{n-1} &= A(\theta)U_{n-2} + F_{n-2} \\ U_n &= A(\theta)U_{n-1} + F_{n-1} \end{aligned} \tag{1}$$

where U_k is the seismic wavefield at k-th time step, F_k is the source term at k-th time step, $A(\theta)$ is the associated coefficient matrix, and θ is the physical parameter of interest, e.g., the wave velocity. $A(\theta)$ indicates that the entries in the matrix depend on θ. To simplify the notation, we

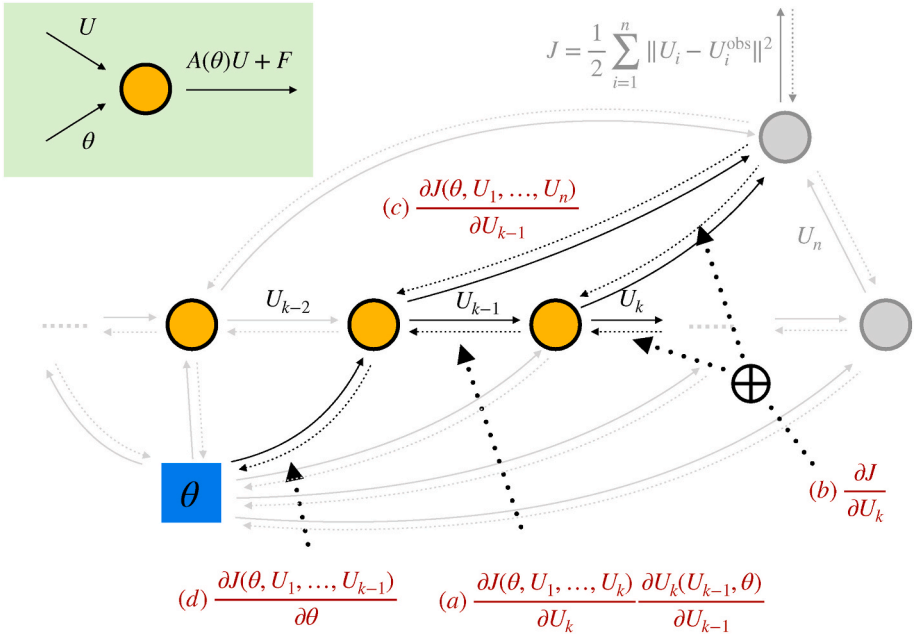

Fig. 2. Computational graph and gradient back-propagation by automatic differentiation

let the misfit function be

$$J(\theta)=\frac{1}{2}\sum_{k=1}^{n}\left|\left|U_k(\theta)-U_k^{\text{obs}}\right|\right|^2$$

where U_k^{obs} is the observation at k-th step. The corresponding Lagrangian functional is

$$L(\theta,U_1,\ldots,U_n)=\frac{1}{2}\sum_{k=1}^{n}\left|\left|U_k-U_k^{\text{obs}}\right|\right|^2+\sum_{k=1}^{n}\lambda_k^T(A(\theta)U_{k-1}+F_{k-1}-U_k) \quad (2)$$

where λ_k is the adjoint variable. The Karush-Kuhn-Tucker (KKT) condition (Luenberger and Ye, 1984) for Eq. (2) reads

$$\begin{aligned}
&\frac{\partial L}{\partial U_n}=U_n-U_n^{\text{obs}}-\lambda_n=0\\
&\frac{\partial L}{\partial U_{n-1}}=U_{n-1}-U_{n-1}^{\text{obs}}-\lambda_{n-1}+A(\theta)^T\lambda_n=0\\
&\vdots\\
&\frac{\partial L}{\partial U_2}=U_2-U_2^{\text{obs}}-\lambda_2+A(\theta)^T\lambda_3=0\\
&\frac{\partial L}{\partial U_1}=U_1-U_1^{\text{obs}}-\lambda_1+A(\theta)^T\lambda_2=0
\end{aligned} \quad (3)$$

Rearranging (3) we obtain

$$\begin{aligned}
&\lambda_n=U_n-U_n^{\text{obs}}\\
&\lambda_{n-1}=A(\theta)^T\lambda_n+U_{n-1}-U_{n-1}^{\text{obs}}\\
&\vdots\\
&\lambda_2=A(\theta)^T\lambda_3+U_2-U_2^{\text{obs}}\\
&\lambda_1=A(\theta)^T\lambda_2+U_1-U_1^{\text{obs}}
\end{aligned} \quad (4)$$

Note that we can compute all the adjoint variables $\lambda_k, k=1,2,\ldots,n$ sequentially from $k=n$ to $k=1$. In this process, we need to perform matrix multiplication with the coefficient matrix $A(\theta)^T$, which is why we call λ_k adjoint variables.

Finally, the gradients of L with respect to θ can be extracted using the computed $\lambda_k, k=1,2,\ldots,n$

$$\frac{\partial L}{\partial \theta}=\sum_{k=1}^{n}\lambda_k{}^T\frac{\partial A(\theta)}{\partial \theta}U_{k-1} \quad (5)$$

In the following text, we describe how reverse-mode AD is used for computing the gradient $\frac{\partial J}{\partial \theta}$ and show that AD calculates the adjoint variables and gradients in the same way as the adjoint-state method (Eq. (4) and (5)). A straightforward way to view AD is to consider a specific operator in the computational graph from $k-1$ to k step:

$$\begin{aligned}
&\text{Forward Computation:} && U_k(U_{k-1},\theta)=A(\theta)U_{k-1}+F_{k-1}\\
&\text{Backward Gradient:} && \frac{\partial U_k(U_{k-1},\theta)}{\partial U_{k-1}}=A(\theta)^T\\
& && \frac{\partial U_k(U_{k-1},\theta)}{\partial \theta}=\frac{\partial A(\theta)}{\partial \theta}U_{k-1}
\end{aligned} \quad (6)$$

We assume that the gradient of J with respect to U_k has already been calculated at the k-th time step. We then back-propagate the gradients to the previous time step (Fig. 2). For convenience we define the gradients as

$$\begin{aligned}
&\mu_k:=\left(\frac{\partial J(\theta,U_1,\ldots,U_k)}{\partial U_k}\right)^T,\quad k=1,2,\ldots,n-1\\
&\mu_n=\left(U_n-U_n^{\text{obs}}\right)^T
\end{aligned} \quad (7)$$

Here, $J(\theta,U_1,\ldots,U_k)$ can be recursively defined as

$$J(\theta,U_1,\ldots,U_n):=\frac{1}{2}\sum_{k=1}^{n}\left|\left|U_k-U_k^{\text{obs}}\right|\right|^2 \quad (8)$$

$$J(\theta,U_1,\ldots,U_k):=J(\theta,U_1,\ldots,U_k,A(\theta)U_k+F_k),\quad k=1,2,\ldots,n-1 \quad (9)$$

where we define $J(\theta,U_1,\ldots,U_k)$ by substituting U_{k+1} in $J(\theta,U_1,\ldots,U_k,U_{k+1})$ with $A(\theta)U_k+F_k$.

We now focus on one specific step shown in bold in Fig. 2. In AD, we need to compute the gradients $\frac{\partial J}{\partial U_{k-1}}$ and $\frac{\partial J}{\partial \theta}$ given the so-called "top" gradients $\frac{\partial J}{\partial U_k}$ (noted by the symbol "b" in Fig. 2). The gradient back-propagation rule for $\frac{\partial J}{\partial U_{k-1}}$ reads

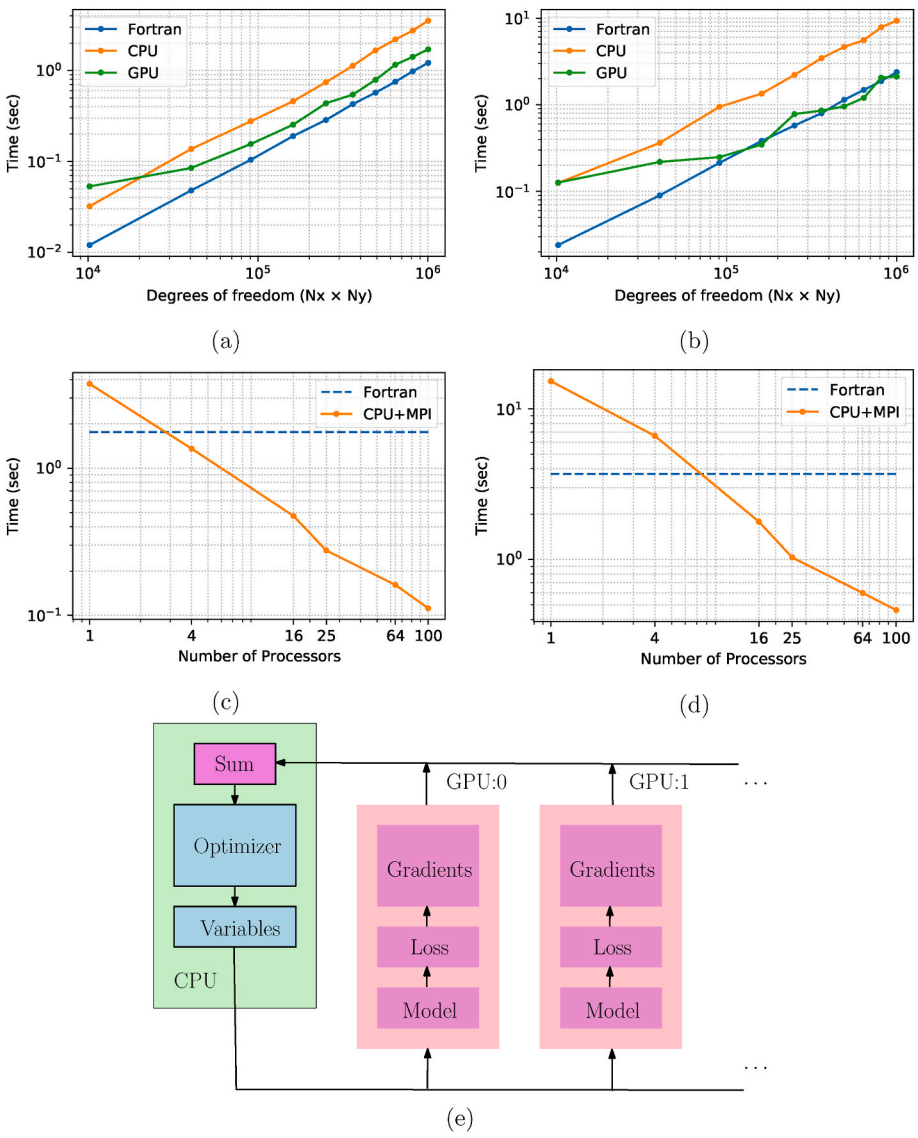

Fig. 3. High-performance computing of ADSeismic: We benchmark the computation times of ADSeismic on CPU and GPU and against the FORTRAN package SEISMIC_CPML on CPU for the acoustic wave equation (a) and the elastic wave equation (b). The computation time of ADSeismic can be further reduced with more CPU processors for the acoustic wave equation (c) and the elastic wave equation (d), because ADSeismic supports MPI for distributed and parallel computing. ADSeismic also inherits multi-GPU computing from Tensorflow (e).

$$\mu_{k-1}^T = \frac{\partial J(\theta, U_1, \ldots, U_{k-1})}{\partial U_{k-1}} = \overbrace{\frac{\partial J(\theta, U_1, \ldots, U_k)}{\partial U_k}}^{(b)} \underbrace{\frac{\partial U_k(U_{k-1}, \theta)}{\partial U_{k-1}}}_{(a)} + \overbrace{\frac{\partial J(\theta, U_1, \ldots, U_n)}{\partial U_{k-1}}}^{(c)}$$
$$= A(\theta)^T \mu_k + \left(U_{k-1} - U_{k-1}^{\text{obs}}\right), \quad k = 2, \ldots, n \tag{10}$$

$$g_{k-1} := \overbrace{\frac{\partial J(\theta, U_1, \ldots, U_{k-1})}{\partial \theta}}^{(d)} = \overbrace{\frac{\partial J(\theta, U_1, \ldots, U_k)}{\partial U_k}}^{(b)} \frac{\partial U_k(U_{k-1}, \theta)}{\partial \theta} = \mu_k^T \frac{\partial A(\theta)}{\partial \theta} U_{k-1} \tag{11}$$

Note J on the left hand side and on the right hand side have different arguments (Fig. 2).

The gradient back-propagation rule for $\frac{\partial J(\theta, U_1, U_2, \ldots, U_{k-1})}{\partial \theta}$ reads

The gradient $\frac{\partial J(\theta)}{\partial \theta}$ is computed by accumulating g_k from all steps

$$\frac{\partial J(\theta)}{\partial \theta} = \sum_{k=1}^{n} g_k = \sum_{k=1}^{n} \mu_k^T \frac{\partial A(\theta)}{\partial \theta} U_{k-1} \tag{12}$$

We now demonstrate the equivalence of the gradients (Eq. (5)) computed using AD and the gradients (Eq. (12)) computed using the adjoint-state method.

Proposition 1. *Assume that* $\{\mu_k\}_{k=1}^n$ *satisfies Eq. (7) and Eq. (10), and* $\{\lambda_k\}_{k=1}^n$ *satisfies Eq. (3), then*

$$\lambda_k = \mu_k, \quad k = 1, 2, \ldots, n \tag{13}$$

And therefore,

$$\sum_{k=1}^{n} \mu_k^T \frac{\partial A(\theta)}{\partial \theta} U_{k-1} = \sum_{k=1}^{n} \lambda_k^T \frac{\partial A(\theta)}{\partial \theta} U_{k-1} \tag{14}$$

Proof. Note $\lambda_n = \mu_n = U_n - U_n^{\text{obs}}$ *and the recursive relations* Eq. (4) *and* Eq. (10) *are the same. Thus, we have* $\lambda_k = \mu_k, \quad k = 1, 2, \ldots, n$. *Therefore,* Eq. (14) *holds.* ∎

Proposition 1 implies that reverse-mode automatic differentiation is mathematically equivalent to the adjoint-state method, and the intermediate gradient $\mu_k = \frac{\partial J}{\partial U_k}$ is exactly the adjoint variable λ_k. In the following text, we describe our general approach for seismic inversion based on the connection between the automatic differentiation and the adjoint state method.

2.3. Implementation

In this section we describe how automatic differentiation assists computing the gradient of the misfit function with respect to the physical parameters in ADSeismic. We use a staggered grid finite difference method for discretizing both the acoustic wave equation and the elastic wave equation with perfectly matched layer (PML) (Roden and Gedney, 2000; Komatitsch and Martin, 2007; Grote and Sim, 2010). The governing equation for the acoustic wave equation is

$$\frac{\partial^2 u}{\partial t^2} = \nabla \cdot \left(c^2 \nabla u\right) + f \tag{15}$$

where u is displacement, f is the source term, and c is the spatially varying acoustic velocity. The inversion parameters of interest are c or f. The governing equation for the elastic wave equation is

$$\begin{aligned} \rho \frac{\partial v_i}{\partial t} &= \sigma_{ij,j} + \rho f_i \\ \frac{\partial \sigma_{ij}}{\partial t} &= \lambda v_{k,k} + \mu\left(v_{i,j} + v_{j,i}\right) \end{aligned} \tag{16}$$

where v is velocity, σ is the stress tensor, ρ is density, and λ and μ are the Lamé's constants. The inversion parameters in the elastic wave equation case are λ, μ, ρ or f.

The finite difference discretization leads to a system of linear equations Eq. (1) for both Eq. (15) and Eq. (16). For the adjoint-state method, we also need to derive and implement Eq. (3) to compute the gradient Eq. (5). This step is unnecessary in ADSeismic since the gradient is extracted automatically from the computational graph. We emphasize that only the forward simulation code is required for building a computational graph and the gradient automatically computed by AD is the same as that for the adjoint-state method.

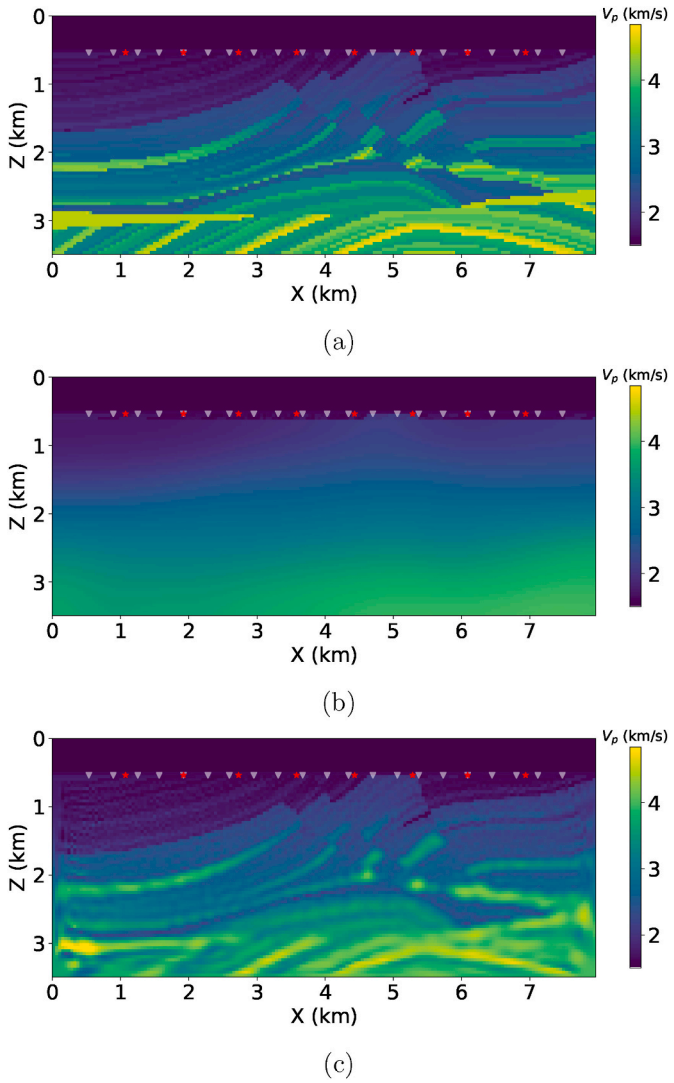

Fig. 4. The Marmousi benchmark model: (a) the true P-wave velocity model; (b) the initial velocity model; (c) the inverted velocity model. The white triangles at the top represent the receiver locations and the red stars represent the source locations. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

We use the Julia package, ADCME,[2] for our implementation since it provides an interface to TensorFlow for automatic differentiation and intuitive Julia syntax for expressing mathematical formulae in numerical simulation. ADCME also provides built-in optimization solvers such as L-BFGS-B (Zhu et al., 1997) for minimizing the misfit function. ADCME allows us to easily extend ADSeismic to other equations or models in seismic applications.

3. Applications

In this section, we first benchmark the performance of ADSeismic on CPU, GPU, and computer clusters for acoustic and elastic wave equations. We then present three applications of ADSeismic to seismic problems including: velocity model estimation, earthquake location, source time function estimation, and earthquake rupture imaging. The

[2] https://github.com/kailaix/ADCME.jl.

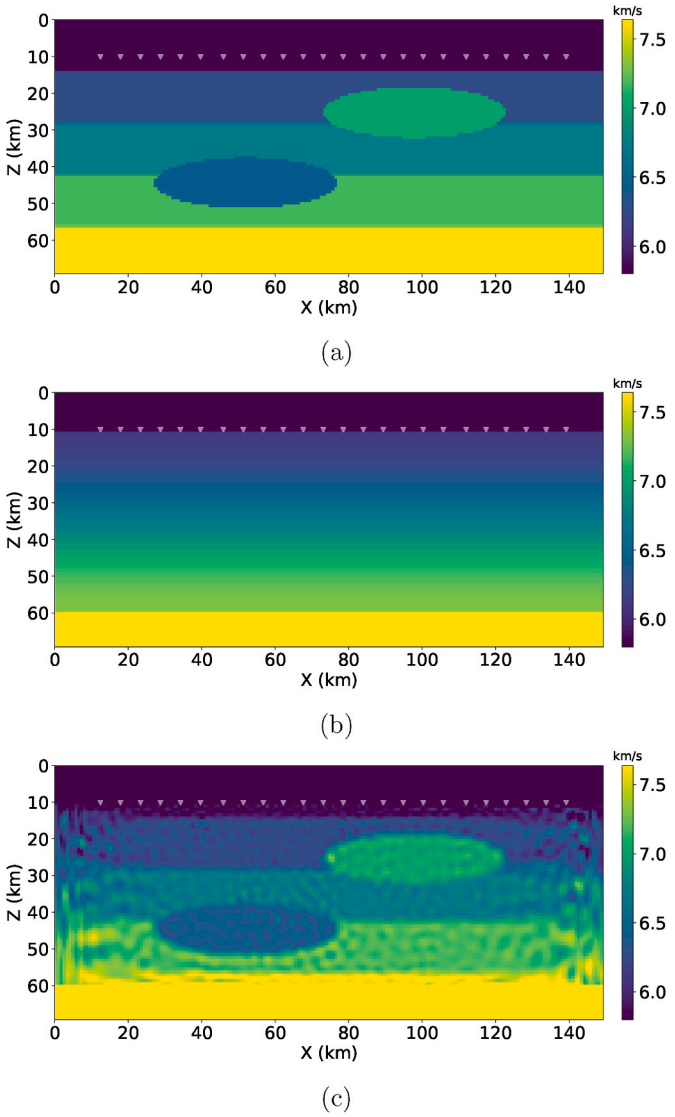

Fig. 5. The layered model with inclusions: (a) the true P-wave velocity model; (b) the initial velocity model; (c) the estimated velocity model. Here we use seven plane waves propagating from the bottom to the surface with incidence angles ranging from -45° to 45° and an interval of 15°.

applications are built with the same forward simulation code (acoustic or elastic wave equations) with only minor changes to specify the inversion parameters to be recovered.

3.1. Performance Benchmarking

Because the backend of ADSeismic is TensorFlow, the same forward simulation code runs on both the CPU and GPU. We benchmark the performance of ADSeismic on the Intel(R) Xeon(R) CPU E5-2698 and the Tesla V100-SXM2 GPU. For comparison, we also report the speed of a dedicated FORTRAN package SEISMIC_CPML[3] for simulating seismic waves. The comparisons between the CPU and GPU times of ADSesmic and the CPU time of the FORTRAN package for the acoustic and elastic equations are shown in Fig. 3a and b. The computation times are averaged over three repeated runs. Because FORTRAN is well optimized for scientific computing, such as vectorization for array-based computing (Loveman, 1993), the FORTRAN package runs 3–5 times faster than

[3] https://github.com/geodynamics/seismic_cpml.

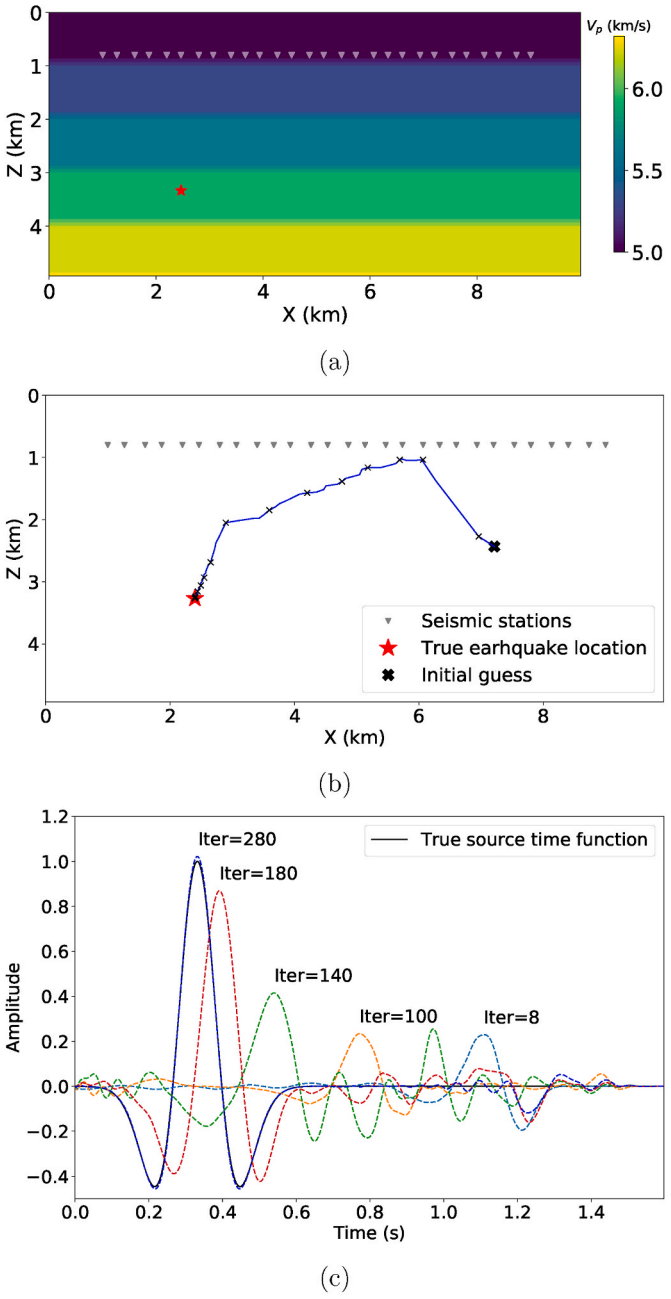

Fig. 6. Inversion of earthquake location and source time function: (a) the velocity model and true source location; (b) the evolution of earthquake location represented by the black "$\times$ " symbol; (c) the evolution of the source time function from a flat initial state.

ADSeismic on CPUs. With GPU acceleration, ADSeismic achieves similar performance to the FORTRAN code.

ADSeismic also inherits the multi-GPU support from TensorFlow such that we can split the source onto different GPUs so that the forward simulation and the associated gradient are computed using AD in parallel across the GPUs. Then, the gradients are assembled on the CPU and fed to the L-BFGS optimizer to update the inversion parameters. Finally, the updated inversion parameters are distributed to all GPUs for the next integration (Fig. 3e).

To extend ADSeismic's applications on large-scale supercomputers, we add MPI into ADSeismic to support distributed and parallel computing. For seismic wave simulation, we can divide the computational domain into several sub-domains and assign the computation of each sub-domain to one of the CPU processors. MPI is used to exchange

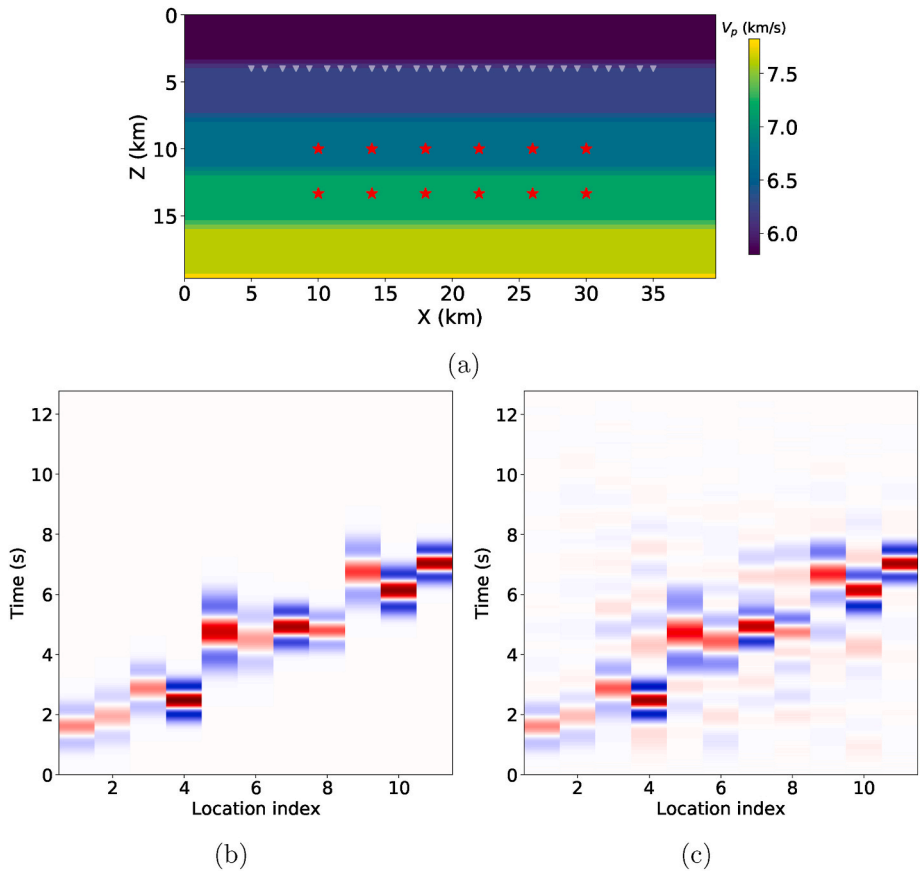

Fig. 7. Inversion of the whole rupture history: (a) the velocity model, receivers (white triangles), and simplified rupture locations (red stars); (b) true slip waveforms; (c) inverted slip waveforms. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the updated wavefield values between adjacent sub-domain at each time step. In this way we can both speed up the computation and utilize the massive memory space of supercomputers for large-scale wavefield simulation and parameter inversion. Fig. 3c and d shows the speedup with the increasing number of CPU processors (Intel(R) Xeon(R) CPU E5-2670).

3.2. Full-waveform inversion

Classic full-waveform inversion (FWI) is based on the adjoint-state method (Tarantola, 1984; Fichtner et al., 2006; Plessix, 2006; Virieux and Operto, 2009). As shown above, AD is mathematically equivalent to the adjoint-state method so that we can apply AD directly to full-waveform inversion without manual derivation of the adjoint-state equations. We demonstrate our method using two cases: the well-known and geometrically complex Marmousi benchmark model (Versteeg, 1994; Martin et al., 2002) (Fig. 4) and a layered model of the Earth's crust with embedded anomalies of elliptical shape (Fig. 5). We place eight active sources on the surface with a spacing of 850 m for the Marmousi benchmark and seven plane waves with incident angles from -45^o to 45^o from the bottom to mimic incoming teleseismic waves for the layered model. We use a Ricker wavelet as the source time function for both cases. Similar to common FWI applications, we choose L-BFGS optimization and a $L2-$ norm loss function for all inversions. We note

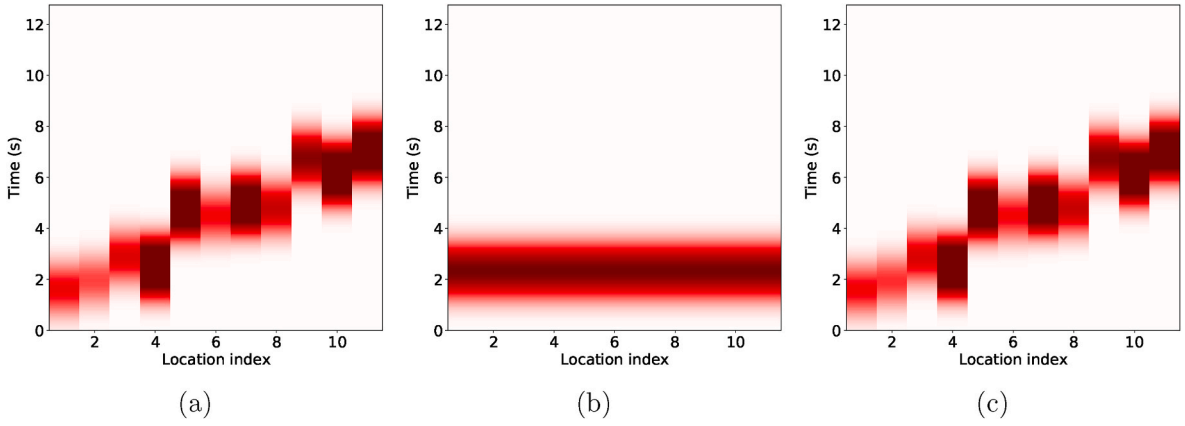

Fig. 8. Inversion of slip time and amplitude: (a) true earthquake slip with the shape of a Gaussian; (b) initial inversion state with a same slip time and amplitude; (b) estimated earthquake slip.

that ADSeismic supports other optimization techniques such as the stochastic gradient descent (SGD) method (Bottou, 2010; Richardson, 2018) although the application and comparison of these optimizers is beyond the scope of this paper. The inversion results in Figs. 4c and 5c show good recovery of the complex velocity structure and anomalies, demonstrating that AD accurately estimates the velocity models.

3.3. Earthquake location and source time function retrieval

Determining earthquake location is a routine but essential earthquake monitoring task for which commonly used methods include 1) linearized inversion for absolute earthquake location (Lienert et al., 1986; Kissling et al., 1994, 1995; Klein, 2002) and relative earthquake location (Waldhauser and Ellsworth, 2000; Schaff et al., 2004; Rubinstein and Beroza, 2007); 2) non-linear inversion methods (Thurber, 1985; Lomax et al., 2000, 2009); and 3) migration-based or time-reversal methods (Nakata and Beroza, 2016; Nakata et al., 2016). The migration-based method produces a focused wavefield that is the same as the gradient in the first iteration of the adjoint-state method (Fichtner, 2010); however, this method does not explicitly give the source location but requires post-processing to extract potential earthquake locations from the focused wavefield.

We use a new non-linear earthquake location method based on full waveforms. The inversion target, the source term $f(x,t)$ in equation (15), is a delta function in space, whose gradient at zero is not well defined, making the direct application of the adjoint-state method difficult. With AD, we can flexibly re-parameterize the optimization target $f(x,t)$ with a continuous Gaussian form

$$f(x,t)=\frac{g(t)}{2\pi\sigma^2}\exp\left(-\frac{||x-x_0||^2}{2\sigma^2}\right) \tag{17}$$

where $g(t)$ is the source time function, x_0 is the earthquake location, and σ is the standard deviation of the Gaussian function, which in our test is set to half of the grid size. In this test, we simultaneously estimate the earthquake location x_0 and the source time function $g(t)$ by fitting the recorded waveforms. Fig. 6 shows the evolution of the earthquake location and source time function during optimization from an initial state of a random selected earthquake location and a flat source time function. The inversion results agree well with the true earthquake location and source time function.

3.4. Earthquake rupture imaging

The rupture process of large earthquakes has resolvable spatial and temporal extent. Imaging this rupture process from observed seismic data contributes to the understanding the complexity behind the evolution of earthquakes. The linearized kinematic inversion method using elastodynamic Green's functions (Kikuchi and Kanamori, 1982; Hartzell and Heaton, 1983; Beroza and Spudich, 1988; Wald et al., 1990; Beroza, 1991; Zhang et al., 2009; Suzuki et al., 2011) and direct imaging methods, such as back-projection (Ishii et al., 2005; Krüger and Ohrnberger, 2005; Walker et al., 2005; Xu et al., 2009; Lay et al., 2010; Simons et al., 2011; Meng et al., 2012), are the two most commonly used methods for imaging the earthquake rupture process. The adjoint-state method has also been tested for rupture process inversion (Kremers et al., 2011; Somala et al., 2018).

We consider a simplified 2D earthquake rupture case to show the potential applications of ADSeismic for imaging the earthquake rupture process. We mimic a simple rupture process with a group of sources activated from the left to right with different rise times and amplitudes (Fig. 7a and b). We consider two inversion targets: the entire rupture history, or the rupture time and amplitude. To estimate the rupture history, we choose the unknown parameter as the source time function $f(t)$. To estimate the rupture time and amplitude, we choose the parameters of rupture time t_0 and amplitude A_0 by assuming that the shape of the source time function is known as a Gaussian function:

$$f(t)=A_0\exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right) \tag{18}$$

To estimate the entire rupture history, the initial state is set to be zero slip for all locations. To estimate the rupture time and amplitude, the initial state is set to be the same Gaussian function with a constant rupture time and amplitude (Fig. 8b). Imaging the entire rupture history requires many more parameters (the total number of time steps) than estimating only the rupture time and amplitude (two parameters A_0 and t_0), with the result that the former problem is less constrained for the same number of receivers. The final inversion results are shown in Figs. 7c and 8c. Note that we have not incorporated a dynamic rupture model to simulate the rupture propagation in this test. Although deriving the adjoint equation is challenging for the coupled system between dynamic earthquake rupture and seismic waves, AD provides a solution to back-propagate the gradients from the wave equation into the dynamic rupture equation to enable the inversion of fault properties based on seismic wave recordings. The potential applications of AD for complex coupled systems in geophysics is an obvious direction for future research.

4. Limitations

Despite the many strengths of ADSeismic, we note three major limitations:

First, as with any inverse problem, inversion based on AD may suffer from ill-conditioning, among these are the often-encountered problem in seismology of cycle-skipping (Virieux and Operto, 2009; Hu et al., 2018), which produces a local minimum when the predicted signal is shifted more than half a wavelength from the observation due to a poor initial model or lack of low frequency information. Neither AD nor adjoint-state methods can solve the ill-conditioning issue, which is intrinsic to the optimization problem. Nevertheless, there are many techniques for improving the conditioning of the optimization problem (Ma and Hale, 2013; Biondi and Almomin, 2014; Wu et al., 2014; Yang et al., 2018) and these can be applied in our AD framework.

Second, reverse-mode AD has demanding memory requirements. This is a noteworthy constraint when running large simulations on GPUs because GPUs usually have lower available memory than CPUs. Techniques such as check-pointing (Griewank and Walther, 2000; Symes, 2007; Chen et al., 2016) have been used to reduce memory requirements, but at the cost of more computation. In ADSeismic, we add MPI support to use the large amount of memory of supercomputers for large-scale seismic problems.

Third, the numerical schemes we consider in this work are all explicit. In some applications, implicit schemes are desirable for reasons such as stability, accuracy, and non-linearity (Liu and Sen, 2009; Chu and Stoffa, 2012; Richardson, 2018). For implicit schemes it is challenging to apply reverse-mode AD techniques since most AD frameworks only provide explicit differentiable operators. Li et al. (2019) introduce the intelligent automatic differentiation method that implements AD for implicit numerical schemes. This approach could be used for extending ADSeismic to implicit schemes.

5. Conclusions

In this paper we have explained the connection between the

automatic differentiation technique in deep learning and adjoint-state methods in seismic inversion. Based on that correspondence we design a general seismic inversion framework, ADSeismic, based on the AD functionality from deep learning software. ADSeismic shows promising results on a series of seismic inversion problems with demonstrated acceleration on GPUs and large-scale CPU clusters. Since deep learning techniques and frameworks are continuously improving, ADSeismic allows for flexibly experimenting with new models, leverages specialized hardware optimized for deep learning, and executes numerical simulations on heterogeneous computing platforms. This should facilitate general seismic inversion in a high performance computing environment. Furthermore, it opens a pathway for innovation in inverse modeling in geophysics by leveraging AD functionalities in a deep learning framework.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank three anonymous reviewers for their helpful comments. Kailai Xu and Eric Darve are supported by the Applied Mathematics Program within the Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR), through the Collaboratory on Mathematics and Physics-Informed Learning Machines for Multiscale and Multiphysics Problems Research Center (DE-SC0019453). Weiqiang Zhu and Gregory C. Beroza are supported by the Department of Energy (DOE) Office of Basic Energy Sciences (DE-SC0020445).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al., 2016. Tensorflow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2017. Automatic differentiation in machine learning: a survey. J. Mach. Learn. Res. 18 (1), 5595–5637.

Beroza, G.C., 1991. Near-source modeling of the Loma Prieta earthquake: evidence for heterogeneous slip and implications for earthquake hazard. Bull. Seismol. Soc. Am. 81 (5), 1603–1621.

Beroza, G.C., Spudich, P., 1988. Linearized inversion for fault rupture behavior: application to the 1984 Morgan Hill, California, earthquake. J. Geophys. Res.: Solid Earth 93 (B6), 6275–6296.

Biondi, B., Almomin, A., 2014. Simultaneous inversion of full data bandwidth by tomographic full-waveform inversion. Geophysics 79 (3), WA129–WA140.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp. 177–186.

Bradley, A.M., 2013. *PDE-constrained optimization and the adjoint method* Tech. In: Rep. Technical Report. Stanford University. https://cs.stanford.edu/~ambrad.

Cao, D., Liao, W., 2015. A computational method for full waveform inversion of crosswell seismic data using automatic differentiation. Comput. Phys. Commun. 188, 47–58.

Chen, T., Xu, B., Zhang, C., Guestrin, C., 2016. Training Deep Nets with Sublinear Memory Cost arXiv preprint arXiv:1604.06174.

Chu, C., Stoffa, P.L., 2012. Implicit finite-difference simulations of seismic wave propagation. Geophysics 77 (2), T57–T67.

Clarke, L., Glendinning, I., Hempel, R., 1994. The MPI message passing interface standard. In: Programming Environments for Massively Parallel Distributed Systems. Springer, pp. 213–218.

Cockett, R., Kang, S., Heagy, L.J., Pidlisecky, A., Oldenburg, D.W., 2015. SimPEG: an open source framework for simulation and gradient based parameter estimation in geophysical applications. Comput. Geosci. 85, 142–154.

Fichtner, A., 2010. Full Seismic Waveform Modelling and Inversion. Springer Science & Business Media.

Fichtner, A., Bunge, H.-P., Igel, H., 2006. The adjoint method in seismology: I. Theory. Phys. Earth Planet. In. 157 (1–2), 86–104.

Griewank, A., Walther, A., 2000. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Trans. Math Software 26 (1), 19–45.

Grote, M.J., Sim, I., 2010. Efficient PML for the Wave Equation arXiv preprint arXiv: 1001.0319.

Hartzell, S.H., Heaton, T.H., 1983. Inversion of strong ground motion and teleseismic waveform data for the fault rupture history of the 1979 Imperial Valley, California, earthquake. Bull. Seismol. Soc. Am. 73 (6A), 1553–1583.

Hu, W., Chen, J., Liu, J., Abubakar, A., 2018. Retrieving low wavenumber information in FWI: an overview of the cycle-skipping phenomenon and solutions. IEEE Signal Process. Mag. 35 (2), 132–141.

Hughes, T.W., Williamson, I.A., Minkov, M., Fan, S., 2019. Wave physics as an analog recurrent neural network. Science Advances 5 (12).

Ishii, M., Shearer, P.M., Houston, H., Vidale, J.E., 2005. Extent, duration and speed of the 2004 Sumatra–Andaman earthquake imaged by the Hi-Net array. Nature 435 (7044), 933–936.

Kikuchi, M., Kanamori, H., 1982. Inversion of complex body waves. Bull. Seismol. Soc. Am. 72 (2), 491–506.

Kissling, E., Ellsworth, W., Eberhart-Phillips, D., Kradolfer, U., 1994. Initial reference models in local earthquake tomography. J. Geophys. Res.: Solid Earth 99 (B10), 19635–19646.

Kissling, E., Kradolfer, U., Maurer, H., 1995. Program VELEST User's Guide-Short Introduction. Institute of Geophysics, ETH Zurich.

Klein, F.W., 2002. User's guide to HYPOINVERSE-2000, a Fortran program to solve for earthquake locations and magnitudes. Tech. Rep.. US Geological Survey.

Komatitsch, D., Martin, R., 2007. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. Geophysics 72 (5), SM155–SM167.

Kremers, S., Fichtner, A., Brietzke, G., Igel, H., Larmat, C., Huang, L., Käser, M., 2011. Exploring the potentials and limitations of the time-reversal imaging of finite seismic sources. Solid Earth 2 (1), 95–105.

Krüger, F., Ohrnberger, M., 2005. Tracking the rupture of the M_w=9.3 Sumatra earthquake over 1,150 km at teleseismic distance. *Nature 435* (7044), 937–939.

Lay, T., Ammon, C.J., Kanamori, H., Koper, K., Sufri, O., Hutko, A., 2010. Teleseismic inversion for rupture process of the 27 February 2010 Chile (Mw 8.8) earthquake. Geophys. Res. Lett. 37 (13).

LeCun, Y., Touresky, D., Hinton, G., Sejnowski, T., 1988. A theoretical framework for back-propagation. In: Proceedings of the 1988 Connectionist Models Summer School, vol. 1, pp. 21–28.

Li, D., Xu, K., Harris, J.M., Darve, E., 2019. Time-lapse Full Waveform Inversion for Subsurface Flow Problems with Intelligent Automatic Differentiation arXiv preprint arXiv:1912.07552.

Lienert, B.R., Berg, E., Frazer, L.N., 1986. HYPOCENTER: an earthquake location method using centered, scaled, and adaptively damped least squares. Bull. Seismol. Soc. Am. 76 (3), 771–783.

Liu, Y., Sen, M.K., 2009. A practical implicit finite-difference method: examples from seismic modelling. J. Geophys. Eng. 6 (3), 231–249.

Lomax, A., Michelini, A., Curtis, A., 2009. Earthquake location, direct, global-search methods. Encycl.Complex.Syst. Sci. 5, 1–33.

Lomax, A., Virieux, J., Volant, P., Berge-Thierry, C., 2000. Probabilistic earthquake location in 3D and layered models. In: Advances in Seismic Event Location. Springer, pp. 101–134.

Louboutin, M., Lange, M., Luporini, F., Kukreja, N., Witte, P.A., Herrmann, F.J., et al., 2019. Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration. Geosci. Model Dev. (GMD) 12 (3), 1165–1187. https://doi.org/10.5194/gmd-12-1165-2019. https://www.geosci-model-dev.net/12/1165/2019/.

Loveman, D.B., 1993. High performance fortran. IEEE Parallel Distr. Technol. Syst. Appl. 1 (1), 25–42.

Luenberger, D.G., Ye, Y., 1984. Linear and Nonlinear Programming, vol. 2. Springer.

Ma, Y., Hale, D., 2013. Wave-equation reflection traveltime inversion with dynamic warping and full-waveform inversion. Geophysics 78 (6), R223–R233.

Martin, G.S., Marfurt, K.J., Larsen, S., 2002. Marmousi-2: an updated model for the investigation of AVO in structurally complex areas. In: SEG Technical Program Expanded Abstracts 2002. Society of Exploration Geophysicists, pp. 1979–1982.

Meng, L., Ampuero, J.-P., Stock, J., Duputel, Z., Luo, Y., Tsai, V., 2012. Earthquake in a maze: compressional rupture branching during the 2012 Mw 8.6 Sumatra earthquake. Science 337 (6095), 724–726.

Nakata, N., Beroza, G., Sun, J., Fomel, S., 2016. Migration-based passive-source imaging for continuous data. In: SEG Technical Program Expanded Abstracts 2016. Society of Exploration Geophysicists, pp. 2607–2611.

Nakata, N., Beroza, G.C., 2016. Reverse time migration for microseismic sources using the geometric mean as an imaging condition. Geophysics 81 (2). KS51–KS60.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al., 2017. Automatic Differentiation in PyTorch.

Plessix, R.-E., 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. Geophys. J. Int. 167 (2), 495–503.

Ren, Y., Xu, X., Yang, S., Nie, L., Chen, Y., 2020. A physics-based neural-network way to perform seismic full waveform inversion. IEEE Access 8, 112266–112277.

Richardson, A., 2018. Seismic Full-Waveform Inversion Using Deep Learning Tools and Techniques arXiv preprint arXiv:1801.07232.

Roden, J.A., Gedney, S.D., 2000. Convolution PML (CPML): an efficient FDTD implementation of the CFS–PML for arbitrary media. Microw. Opt. Technol. Lett. 27 (5), 334–339.

Rubinstein, J.L., Beroza, G.C., 2007. Full waveform earthquake location: application to seismic streaks on the Calaveras fault, California. J. Geophys. Res.: Solid Earth 112 (B5).
Rücker, C., Günther, T., Wagner, F.M., 2017. pyGIMLi: an open-source library for modelling and inversion in geophysics. Comput. Geosci. 109, 106–123.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323 (6088), 533–536.
Sambridge, M., Rickwood, P., Rawlinson, N., Sommacal, S., 2007. Automatic differentiation in geophysical inverse problems. Geophys. J. Int. 170 (1), 1–8.
Schaff, D.P., Bokelmann, G.H., Ellsworth, W.L., Zanzerkia, E., Waldhauser, F., Beroza, G. C., 2004. Optimizing correlation techniques for improved earthquake location. Bull. Seismol. Soc. Am. 94 (2), 705–721.
Simons, M., Minson, S.E., Sladen, A., Ortega, F., Jiang, J., Owen, S.E., et al., 2011. The 2011 magnitude 9.0 Tohoku-Oki earthquake: mosaicking the megathrust from seconds to centuries. science 332 (6036), 1421–1425.
Somala, S.N., Ampuero, J.-P., Lapusta, N., 2018. Finite-fault source inversion using adjoint methods in 3D heterogeneous media. Geophys. J. Int. 214 (1), 402–420.
Suzuki, W., Aoi, S., Sekiguchi, H., Kunugi, T., 2011. Rupture process of the 2011 Tohoku-Oki mega-thrust earthquake (M9. 0) inverted from strong-motion data. Geophys. Res. Lett. 38 (7).
Symes, W.W., 2007. Reverse time migration with optimal checkpointing. Geophysics 72 (5), SM213–SM221.
Symes, W.W., 2015. IWAVE Structure and Basic Use Cases, vol. 85. THE RICE INVERSION PROJECT.
Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation. Geophysics 49 (8), 1259–1266.
Thurber, C.H., 1985. Nonlinear earthquake location: theory and examples. Bull. Seismol. Soc. Am. 75 (3), 779–790.
Versteeg, R., 1994. The Marmousi experience: velocity model determination on a synthetic complex data set. Lead. Edge 13 (9), 927–936.
Virieux, J., Operto, S., 2009. An overview of full-waveform inversion in exploration geophysics. Geophysics 74 (6), WCC1–WCC26.
Vlasenko, A., Köhl, A., Stammer, D., 2016. The efficiency of geophysical adjoint codes generated by automatic differentiation tools. Comput. Phys. Commun. 199, 22–28.
Wald, D.J., Helmberger, D.V., Hartzell, S.H., 1990. Rupture process of the 1987 Superstition Hills earthquake from the inversion of strong-motion data. Bull. Seismol. Soc. Am. 80 (5), 1079–1098.
Waldhauser, F., Ellsworth, W.L., 2000. A double-difference earthquake location algorithm: method and application to the northern Hayward fault, California. Bull. Seismol. Soc. Am. 90 (6), 1353–1368.
Walker, K.T., Ishii, M., Shearer, P.M., 2005. Rupture details of the 28 March 2005 Sumatra Mw 8.6 earthquake imaged with teleseismic P waves. Geophys. Res. Lett. 32 (24).
Witte, P.A., Loubouting, M., Herrmann, F.J., 2020, December. JUDI4Flux: Seismic modeling for deep learning. Zenodo. https://doi.org/10.5281/zenodo.4301017 https://doi.org/10.5281/zenodo.4301017.
Wu, R.-S., Luo, J., Wu, B., 2014. Seismic envelope inversion and modulation signal model. Geophysics 79 (3), WA13–WA24.
Xu, Y., Koper, K.D., Sufri, O., Zhu, L., Hutko, A.R., 2009. Rupture imaging of the Mw 7.9 12 May 2008 Wenchuan earthquake from back projection of teleseismic P waves. G-cubed 10 (4).
Yang, Y., Engquist, B., Sun, J., Hamfeldt, B.F., 2018. Application of optimal transport and the quadratic Wasserstein metric to full-waveform inversion. Geophysics 83 (1), R43–R62.
Yee, K., 1966. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. Antenn. Propag. 14 (3), 302–307.
Zhang, Y., Feng, W., Xu, L., Zhou, C., Chen, Y., 2009. Spatio-temporal rupture process of the 2008 great Wenchuan earthquake. Sci. China Earth Sci. 52 (2), 145–154.
Zhu, C., Byrd, R.H., Lu, P., Nocedal, J., 1997. Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound-constrained optimization. ACM Trans. Math Software 23 (4), 550–560.