Automating adjoint wave-equation travel-time tomography using scientific workflow

Xiaofeng Zhang · Po Chen · Satish Pullammanappallil

Received: 30 July 2013 / Accepted: 31 October 2013 / Published online: 22 November 2013

© The Seismological Society of China, Institute of Geophysics, China Earthquake Administration and Springer-Verlag Berlin Heidelberg 2013

Abstract Recent advances in commodity high-performance computing technology have dramatically reduced the computational cost for solving the seismic wave equation in complex earth structure models. As a consequence, wave-equation-based seismic tomography techniques are being actively developed and gradually adopted in routine subsurface seismic imaging practices. Wave-equation travel-time tomography is a seismic tomography technique that inverts cross-correlation travel-time misfits using full-wave Fréchet kernels computed by solving the wave equation. This technique can be implemented very efficiently using the adjoint method, in which the misfits are back-propagated from the receivers (i.e., seismometers) to produce the adjoint wave-field and the interaction between the adjoint wave-field and the forward wave-field from the seismic source gives the gradient of the objective function. Once the gradient is available, a gradient-based optimization algorithm can then be adopted to produce an optimal earth structure model that minimizes the objective function. This methodology is conceptually straightforward, but its implementation in practical situations is highly complex, error-prone and computationally demanding. In this study, we demonstrate the feasibility of automating wave-equation travel-time tomography based on the adjoint method using Kepler, an open-source software package for designing, managing and executing scientific workflows. The

workflow technology allows us to abstract away much of the complexity involved in the implementation in a manner that is both robust and scalable. Our automated adjoint wave-equation travel-time tomography package has been successfully applied on a real active-source seismic dataset.

Keywords Adjoint tomography · Scientific workflow · Seismic inversion · Kepler

1 Introduction

Seismic waves generated by natural earthquakes or man-made seismic sources and recorded by modern digital seismometers carry important information about the internal geological structure of the Earth. Seismic tomography, an imaging technique that resembles the CT scan used for imaging the human body, has been the most important tool for imaging the deep structure of the Earth's interior using seismic waves.

The propagation of seismic waves inside the Earth is governed by the elastodynamic equation. Conventional ray-theoretic seismic tomography techniques are based on the Eikonal equation, which is a high-frequency approximation of the elastodynamic equation. The resolution of ray-theoretic methods is inherently limited to the size of the first Fresnel zone due to the high-frequency approximation adopted in its formulation. Another class of tomography techniques known as the “wave-equation” methods is based directly on the elastodynamic equation itself. This type of wave-equation tomography techniques has the potential to substantially improve the resolution and the accuracy of the tomographic images of the Earth's interior.

The theoretical foundation of wave-equation tomography techniques was laid out more than two decades ago based on

X. Zhang · P. Chen (✉)
Department of Geology and Geophysics, University of Wyoming, Laramie, WY 82071, USA
e-mail: pchen@uwyo.edu

S. Pullammanappallil
Optim Software and Data Solutions LLC, 200 S. Virginia St., Suite 560, Reno, NV 89501, USA
e-mail: satish@optimsoftware.com

the adjoint state method (e.g., Bamberger et al. 1982; Tarantola 1984, 1988). Nevertheless, the wide adoption of the wave-equation methodology in routine seismic tomography practices has been faced with two major obstacles. Wave-equation tomography often requires numerical solutions of the elastodynamic equation (forward and adjoint), which can be computationally demanding, especially for three-dimensional problems. The procedure for wave-equation tomography techniques can be relatively complex and error-prone compared with conventional ray-theoretic tomography techniques. In this article, we explore the possibility of completely automating a particular wave-equation tomography technique, the wave-equation travel-time tomography, using scientific workflow based on Kepler, an open-source software package for designing, managing, and executing scientific workflows. Kepler allows us to encapsulate complex procedures, which often involve running multiple computationally intensive Fortran and/or C/C++ codes, into simple modules with clearly defined interfaces. It also provides a unified interface for harnessing different types of computational resources located on local computer clusters and/or remote computing service centers. By liberating users from the repetitive tasks of designing, managing, and monitoring large-scale parallel calculations, we expect that our workflow implementation will allow seismic tomography practitioners to focus more on the scientific application rather than data and process management and eventually accelerate the wide adoption of wave-equation-based tomography techniques.

2 Wave-equation travel-time tomography based on the adjoint method

The term “wave-equation travel-time inversion” was introduced to the seismic tomography community in the seminal paper of Luo and Schuster (1991). In that paper, cross-correlation travel-time misfits between observed waveforms and the corresponding synthetic waveforms computed using a reference earth structure model were inverted using the adjoint method. The finite-difference solution of the 2D acoustic wave equation was used to carry out both the forward simulations from the seismic sources to the receivers and the adjoint simulations from the receivers. We briefly summarize some of the important mathematical formulas in the following. The notation we adopt here follows that in Chen et al. (2007), which is slightly more general than that in Luo and Schuster (1991) and is applicable to both acoustic and elastic wave equations in 3D.

We begin our formulation with the definition of a misfit measurement, indexed by n , that measures the misfit between the i -th component observed seismogram $\bar{u}_s^i(x_r, t)$ and the corresponding model-predicted (i.e., synthetic) seismogram $u_s^i(x_r, t)$,

$$d_{sm}^i = D_n [\bar{u}_s^i(x_r, t), u_s^i(x_r, t)], \quad (1)$$

where s and r are indexes used to reference the different sources and receivers in a dataset. We assume that the misfit measurement operator D_n is constructed to satisfy

$$D_n [u_s^i(x_r, t), u_s^i(x_r, t)] = 0. \quad (2)$$

The measurement process generally involves nonlinear operations both on the observed seismograms and on the corresponding synthetics. The first-order Fréchet derivative of the misfit measurement d_{sm}^i with respect to the synthetic waveform $u_s^i(x_r, t)$ can be expressed using an integral equation,

$$\delta d_{sm}^i = \int_0^T dt J_{sm}^i(t) \delta u_s^i(x_r, t), \quad (3)$$

where the integration kernel $J_{sm}^i(t)$ is termed the “seismogram perturbation kernel” (Chen et al. 2007) and the time interval $[0, T]$ is the length of the complete synthetic seismogram. Equation (3) thereby provides a linear relation between the perturbation of the waveform and the perturbation of the misfit measurement. Examples of the misfit measurement operator D_n and the corresponding seismogram perturbation kernel $J_{sm}^i(t)$ are given in Chen et al. (2007). The seismogram perturbation kernel for broadband cross-correlation travel-time misfit measurement is proportional to the synthetic velocity (e.g., Luo and Schuster 1991; Zhao and Jordan 1998), i.e.,

$$J_{sm}^i(t) = - \frac{\partial_t u_s^i(x_r, t) [H(t - t_n) - H(t - t'_n)]}{\int_{t_n}^{t'_n} dt |\partial_t u_s^i(x_r, t)|^2}, \quad (4)$$

where $H(t)$ is the Heaviside function, $[t_n, t'_n]$ is the time window that isolates the target seismic phase used for making the misfit measurement. The first-order Fréchet derivative of the waveform with respect to the earth structure model is given by the first-order term in the Born series (Dahlen and Tromp 1998; Zhao et al. 2000),

$$\begin{aligned} \delta u_s^i(x_r, t) = & - \int dV(x) \\ & \int d\tau \left[\sum_j G_{ij}(x_r, t - \tau; x) \partial_\tau^2 u_s^j(x, \tau) \delta \rho(x) \right. \\ & \left. + \sum_{jklm} \partial_k G_{ij}(x_r, t - \tau; x) \partial_l u_s^m(x, \tau) \delta c_{jklm}(x) \right], \end{aligned} \quad (5)$$

where ∂_k is the partial derivative with respect to x_k and ∂_τ is the partial derivative with respect to τ , G is the Green's tensor, and $\delta \rho(x)$ and $\delta c_{jklm}(x)$ are perturbations to the mass density and the elastic moduli.

We consider an objective function defined as the following

$$\chi = \frac{1}{2} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \sum_{i,n} |d_{srn}^i|^2, \quad (6)$$

where N_s and N_r are the total number of seismic sources and receivers involved in a dataset. The objective function defined here actually is the inner product of the misfit vector with itself. This inner product can also be weighted using a data covariance matrix, which we omit here for notational simplicity. The first-order Fréchet derivative of the objective function with respect to the earth structure model, i.e., the mass density and the elastic moduli, can then be expressed as an integral equation, i.e.,

$$\begin{aligned} \delta\chi &= \sum_{srn} d_{srn}^i \delta d_{srn}^i = \sum_{srn} d_{srn}^i \int_0^T dt J_{srn}^i(t) \delta u_s^i(x_r, t) \\ &= \int dV(x) K_{\chi}^{\rho}(x) \delta\rho(x) \\ &\quad + \sum_{jklm} \int dV(x) K_{\chi}^{c_{jklm}}(x) \delta c_{jklm}(x), \end{aligned} \quad (7)$$

where the integration kernels $K_{\chi}^{\rho}(x)$ and $K_{\chi}^{c_{jklm}}(x)$ give the exact gradients of the objective function χ with respect to the earth structure model. Considering Eq. (5) and the reciprocity principle (Aki and Richards 2002),

$$G_{ji}(x_r, t - \tau; x) = G_{ji}(x, t - \tau; x_r), \quad (8)$$

the gradients of the objective function with respect to mass density and elastic moduli can then be expressed as

$$\begin{aligned} K_{\chi}^{\rho}(x) &= - \sum_{srn} \int dt \int d\tau d_{srn}^i J_{srn}^i(t) \\ &\quad \sum_j G_{ji}(x, t - \tau; x_r) \partial_{\tau}^2 u_s^j(x, \tau), \end{aligned} \quad (9)$$

$$\begin{aligned} K_{\chi}^{c_{jklm}}(x) &= - \sum_{srn} \int dt \int d\tau d_{srn}^i J_{srn}^i(t) \partial_k \\ &\quad G_{ji}(x, t - \tau; x_r) \partial_l u_s^m(x, \tau). \end{aligned} \quad (10)$$

At this point, there are two physically equivalent but computationally different (Chen et al. 2007) approaches to proceed. The scattering-integral (SI) method (e.g., Zhao et al. 2005) is based on explicitly calculating and storing the receiver Green's tensors (RGTs) $G_{ji}(x, t - \tau; x_r)$ for every receiver used in a dataset. The SI method allows construction of not only the gradient of the objective function, but also its Hessian (i.e., the second-order derivative of the objective function with respect to the earth structure model), so that the Gauss–Newton algorithm can be adopted to minimize the objective function. The availability of RGTs also allows rapid inversion of seismic source parameters by using the reciprocity principle (e.g., Zhao et al. 2006; Lee et al. 2011). The disadvantage of the SI method is that the disk

space needed for storing the RGTs is quite large, especially when the number of receivers used in a dataset is large. The adjoint method (e.g., Tarantola 1988; Tromp et al. 2005; Chen 2011; Liu and Gu 2012) does not allow explicit construction of the full Hessian but it only requires a very small amount of disk spaces. In the adjoint method, the adjoint wave-field for seismic source s can be expressed as,

$$[u_s^j]^{\dagger}(x, \tau) = \int dt \sum_{rin} G_{ji}(x, \tau - t; x_r) J_{srn}^i(-t) d_{srn}^i. \quad (11)$$

This adjoint wave-field can be constructed without explicitly computing the receiver Green's tensor. In fact, if we solve the adjoint wave equation, which turns out to be identical to the forward wave equation under certain circumstances (e.g., Tarantola 1988), using the adjoint source field, which is defined as

$$[f_s^i]^{\dagger}(x, t) = \sum_{rn} J_{srn}^i(-t) d_{srn}^i \delta(x - x_r), \quad (12)$$

the result is identical to the adjoint wave-field as expressed in Eq. (11). The adjoint source field is obtained by placing the time-reversed and misfit-weighted seismogram perturbation kernels at the receiver locations. Once the adjoint wave-field for the source s is obtained, the source-specific kernels can be computed as

$$\begin{aligned} {}_s K_{\chi}^{\rho}(x) &= - \int d\tau \sum_j \partial_{\tau}^2 u_s^j(x, \tau) [u_s^j]^{\dagger}(x, -\tau) \\ &= - \int d\tau \sum_j \partial_{\tau} u_s^j(x, \tau) \partial_{\tau} [u_s^j]^{\dagger}(x, -\tau), \end{aligned} \quad (13)$$

$${}_s K_{\chi}^{c_{jklm}}(x) = - \int d\tau \partial_l u_s^m(x, \tau) \partial_k [u_s^j]^{\dagger}(x, -\tau). \quad (14)$$

The gradients of the objective function can be obtained by summing the source-specific kernels over all sources used in a dataset,

$$K_{\chi}^{\rho}(x) = \sum_{s=1}^{N_s} {}_s K_{\chi}^{\rho}(x), \quad (15)$$

$$K_{\chi}^{c_{jklm}}(x) = \sum_{s=1}^{N_s} {}_s K_{\chi}^{c_{jklm}}(x). \quad (16)$$

In practice, the gradients computed in Eqs. (15) and (16) are smoothed using a model covariance operator (e.g., Tape et al. 2010; Fichtner et al. 2009; Chen 2011) before used in a conjugate-gradient algorithm or any other types of gradient-based optimization algorithms to minimize the objective function.

The gradients, $K_{\chi}^{\rho}(x)$ and $K_{\chi}^{c_{jklm}}(x)$, are functions of space x and can be discretized on a spatial mesh. We denote the discretized and smoothed gradients as column vectors K^{ρ} and $K^{c_{jklm}}$. The gradient vectors for different types of model

parameters can be concatenated into a single column vector, which we denote as K . In a typical conjugate-gradient algorithm, the descent direction at the current iteration (i.e., iteration k), which we denote as P_k , is computed from the descent direction at the previous iteration P_{k-1} and the gradient of the current iteration K_k as

$$P_k = -K_k + \beta_k P_{k-1}, \quad (17)$$

where β_k is a scalar computed from the gradients of the previous and current iterations. There are a variety of formulas for computing β_k (Press et al. 1992). One of the most widely used formulas, the Fletcher–Reeves' formula (Press et al. 1992), which is also used in our implementation, is

$$\beta_k = \frac{K_k^T K_k}{K_{k-1}^T K_{k-1}}, \quad (18)$$

where superscript T represents the transpose. Once the descent direction is obtained, a line search is performed to find the optimal step length α_k to walk in the descent direction. In practice, a quadratic interpolation is sufficient for finding the optimal step length. Once the line search is completed, the earth structure model vector $\mathbf{m}$ can be updated to the next iteration,

$$m_{k+1} = m_k + \alpha_k P_k, \quad (19)$$

and the algorithm proceeds until a pre-specified convergence criterion is met.

3 Scientific workflows

Advancements in computing technology have significantly reduced the cost for numerical simulations of natural systems, including those involved in seismic wave propagations. With the new opportunities for scientists also come new challenges, which involve managing the ever-increasing, large volume of data generated by the simulations and the constantly evolving, highly complex computational environments provided by modern computer clusters, the distributed Grid and the Cloud computational platforms. Scientific workflows aim to address many of the challenges by providing a formal description of a process for accomplishing a scientific objective using abstract tasks and their interdependencies.

Kepler is one of the free software packages for designing, managing, and executing scientific workflows. It is built on top of the mature Ptolemy II platform, which is an open-source software framework for heterogeneous, concurrent modeling and design. Unlike many other workflow management systems, Kepler separates the structure of the workflow model from its model of computation (MoC, see below) and allows different MoCs to be bound to the same

workflow graph. It also supports hierarchy in workflows and allows nested workflows, which enables workflow designers to build reusable, modular sub-workflows.

A workflow can be considered as a directed graph W with its nodes representing *actors* (i.e., tasks and workflow steps) and its directed edges representing the *connections* among actors. We can associate a set of parameters p , an input data state x and an output data state y with the workflow W . The model of computation (MoC) M describes how to execute a parameterized workflow W_p on x to uniquely determine the output y , i.e., $y = M(W_p(x))$. An actor can contain a collection of other actors. This type of actors is called *composite actors*. An entire workflow can be encapsulated into a composite actor, which can be included as a component within a larger workflow. A nested workflow can become a reusable component and a collection of well-designed reusable components can significantly reduce the amount effort for building a wide range of workflows. Kepler provides a collection of pre-defined commonly used set of MoCs, which are called *directors*. In a nested workflow, it is possible to have different directors at different levels. A sub-workflow (i.e., a composite actor) that inherits the director from the upper level is called a *transparent composite*; A sub-workflow that has its own director is called an *opaque composite*. The directors that were used in implementing our adjoint wave-equation travel-time tomography workflow are the Synchronous Dataflow (SDF) and the Dynamic Dataflow (DDF). The SDF director is used where the workflow execution can be synchronous, with processing occurring one component at a time in a pre-determined sequence. The DDF director is used when there are looping and/or branching structures in the workflow.

3.1 The composite actor for managing simulation jobs

We build our workflow for the adjoint wave-equation travel-time tomography using a “bottom-up” approach and start with the most elementary sub-workflow for submitting and monitoring the execution status of wave-propagation simulation jobs. On a computer cluster that is shared by many users, the execution of a parallel program is usually handled by a job manager. The user requests for computational resources for executing a parallel program by submitting a job script to the job manager. The job manager then places the job in a queue, together with the jobs submitted by other users. When the resources needed for executing a job become available, the job manager allocates those resources for the job and starts its execution. When the job finishes, the job writes out a log file reporting the status of the execution.

In our implementation, the 3D elastic wave equation is solved using an MPI-parallelized 4th-order staggered-grid

Fig. 1 The composite actor “SimJob” and the workflow contained inside it. In this and the following two figures, *boxes* with rounded corners are normal Kepler actors and *boxes* with rectangular corners are either data objects or Kepler expression actors, which evaluate mathematical and logical expressions. The *arrows* connecting the actors are used loosely to indicate both the execution order and the flow of data items (tokens). Here we did not use the exact Kepler workflow diagram, which not only has a more rigorous format but also requires a more in-depth understanding of the Kepler system

finite-difference code (Olsen 1994). To prepare, submit, and monitor the status of our parallel job, we built a composite actor named “SimJob,” which uses Kepler’s DDF director and pre-defined actors: “JobManager,” “JobCreator,” “JobSubmitter,” and “JobStatus.” A diagram of our composite actor “SimJob” is shown in Fig. 1. Kepler is capable of interfacing with different types of cluster job managers. The internal actor “JobManager” allows users to select the specific type of job manager that is used on the cluster. The internal actor “JobCreator” prepares for the working directory on the cluster’s file system and generates the job script using the name of the parallel program provided by the user and in a format conforming to the job manager selected by the “JobManager” actor. The internal actor “JobSubmitter” simply submits the job script to the job manager and records the status of the job submission process. If it is not successful, the actor can throw an exception based on the user’s preference. After the job is successfully submitted, the internal actor “JobStatus” can be used in a loop to constantly check the status of the job. If the job is not finished, it will idle for 60 s and check the status again. If the job is finished, the “CheckOutput” actor, which is actually a modified version of the “ExternalExecution” actor

Fig. 2 The composite actor “SrcKer” and the workflow inside it. We note that the output files generated by “SimJob_FW” are used both by the “SimJob_Misfit” actor and the “SimJob_Ker” actor; therefore, in the diagram we drew an *arrow* from “SimJob_FW” to “SimJob_Ker.” The “Observed Seismograms” box is a data object used as an input to the “SimJob_Misfit”

provided by Kepler, will launch a script that checks the output from the finite-difference code. Under normal circumstances, the finite-difference code should generate a set of binary files containing the space–time volume of either the forward wave-field or the adjoint wave-field, depending upon the source used in the simulation. Under some rare circumstances, the normal execution of the finite-difference code can be interrupted due to hardware failure, network problems, or some other reasons and some of the binary output files may not be generated or can become corrupted. The script simply checks the total number of the binary output files and the size of each file. If they do not match exactly with the expected values, it will clean up the work directory and the workflow will go back to re-create and re-submit the job. If after 3 trials the problem still exists, the workflow will abort. If all the output files are correctly generated, the workflow will come to the end and finish.

3.2 The composite actor for computing source-specific kernels

As shown in Eqs. (13) and (14), the calculation of the source-specific kernels ${}_s K_{\chi}^{\rho}(x)$ and ${}_s K_{\chi}^{c_{jklm}}(x)$ for source s requires the velocity (i.e., temporal derivative of displacement) and strain (i.e., spatial derivative of displacement) fields of the forward and the adjoint wave-fields. These velocity and strain fields are contained in the binary output files generated by the finite-difference wave-propagation simulations. To calculate the adjoint wave-field, we

need the adjoint source field as defined in Eq. (12), which requires all the cross-correlation travel-time misfit measurements and their corresponding seismogram perturbation kernels, as defined in Eq. (4), for source s . To calculate the misfit measurements and their seismogram perturbation kernels, we need access to the forward velocity wave-field from the source s , which is also generated by the finite-difference wave-propagation simulation for source s . We construct a composite actor named “SrcKer” for all the operations needed for the calculation of the source-specific kernels. We use the SDF director for this composite actor and the composite actor “SimJob” constructed in the previous section to submit and manage parallel jobs. Figure 2 shows a diagram of the workflow inside the “SrcKer” composite actor.

The workflow starts by the “SimJob_FW” actor that submits and manages the job for the finite-difference calculation of the forward wave-field. This actor is adapted from the generic “SimJob” actor constructed in the previous section by setting the parallel executable to be the finite-difference wave-propagation simulation code and the input source field to the one specified by the source parameters of the seismic source s . As shown in the previous section, the workflow inside the “SimJob_FW” composite actor will ensure that the finite-difference simulation finishes correctly and produces the correct output files, which will be needed by the next actor “SimJob_Misfit.” The “SimJob_Misfit” actor is adapted from the “SimJob” actor by setting the parallel executable to be an MPI-parallelized C++ code for making broadband cross-correlation travel-time misfit measurements and generating the corresponding seismogram perturbation kernels according to Eq. (4). The parallel misfit measurement code takes two sets of input files: the synthetic seismograms generated by the forward finite-difference simulation completed by the previous actor “SimJob_FW” and the observed seismograms for source s . The time windows $[t_n, t'_n]$ needed for making the misfit measurement had been picked and stored in the header fields of the observed seismograms. The corresponding time window on the synthetic seismogram was re-picked automatically in each iteration. These time windows were used as input for the “SimJob_Misfit” actor. The output of this actor is the adjoint source field as defined in Eq. (12) and is used as input for the “SimJob_AW” actor, which performs the finite-difference simulation using the adjoint source field and generates the adjoint velocity and strain wave-fields. SimJob_Misfit also calculates the objective function using the misfit measurement made on each seismogram. The calculation of the source-specific kernels is performed by the “SimJob_Ker” actor, which is adapted from the “SimJob” actor by setting the executable to be an MPI-

parallelized Fortran code that implements Eqs. (13) and (14). This parallel Fortran code reads the forward velocity and strain wave-fields generated by the “SimJob_FW” actor and the adjoint velocity and strain wave-fields generated by the “SimJob_AW” actor, reverses the temporal axis of the adjoint fields, and calculates the zero-lag temporal correlation between the forward and the time-reversed adjoint fields on every spatial grid point. The output of this actor is a set of binary files storing the source-specific kernels on the spatial mesh used in the finite-difference wave-propagation simulations. Once the calculation of the source-specific kernels for the source s is completed, the “CleanUp_Wav” actor, which is adapted from Kepler’s “ExternalExecution” actor, launches a script that deletes all the forward and adjoint velocity and strain fields from the disk.

3.3 The top-level workflow

The top-level workflow handles the conjugate-gradient iterations and we denote the iteration number as k . Within each conjugate-gradient iteration, we need to iterate the “SrcKer” composite actor N_s times with each iteration using a different source index s . Once the source-specific kernels are completed for all sources in the dataset, the gradients of the objective function, $K_\chi^p(x)$ and $K_\chi^{c_{jklm}}(x)$, can be computed by simply summing all the source-specific kernels according to Eqs. (15) and (16). The gradients can then be smoothed using different techniques. In our implementation, we used a simple spatial averaging operator. In the conjugate-gradient algorithm, the descent direction is a linear combination of the gradient and the descent direction of the previous iteration as shown in Eq. (17). We first compute the coefficient β_k using Eq. (18) and then compute the descent direction P_k using Eq. (17). Once the descent direction is obtained, a line search for determining the optimal step length α_k in the descent direction needs to be performed.

Once the descent direction is fixed at P_k , the objective function can be considered as a function of only one independent variable, which is the step length α . At $\alpha = 0$, we already know the value of the objective function, which is simply a summation of the squared misfit measurements for all sources, and its derivative with respect to α , which is the inner product between the gradient of the objective function χ and the descent direction, i.e., $d\chi(m_k + \alpha P_k)/d\alpha = \nabla_m \chi \cdot P_k = K_k \cdot P_k$. To perform a quadratic interpolation, we need the value of the objective function at a different α . In our implementation, the trial step length can be adjusted by the user. For instance, we can compute the value of the objective function at $\alpha = 1$, which requires a forward finite-difference simulation for

Fig. 3 The top-level workflow and the components inside it. It contains the outside conjugate-gradient loop controlled by “Ramp_k” and an inside loop over source index s controlled by “Ramp_s.” The “LineSearch” actor is a composite actor with a nested loop over source index s embedded in it

every source in the dataset using the earth structure model $m_k + P_k$. Once we have the values for $\chi(0)$, $\chi(1)$, and the derivative $d\chi/d\alpha$ at $\alpha = 0$, we can fit the function $\chi(\alpha)$ with a 2nd-order polynomial of α and the location of the minimum of this polynomial is chosen to be the optimal step length α_k . Once the optimal step length is obtained, the earth structure model is updated using Eq. (19).

A diagram for the top-level workflow is shown in Fig. 3. Kepler provides the “Ramp” actor for controlling iterations. This actor keeps track of the iterations and increments its index every time an iteration is completed. We use the actor “Ramp_k” to control the outermost conjugate-gradient iteration with the iteration index k . The actor “Ramp_s” is used for controlling the iterations over the source index s , which increases from 1 to N_s . For every iteration, the composite actor “SrcKer” is executed with a different s . Once the source-specific kernels for all sources are completed (i.e., the iterations of “Ramp_s” are all finished), the actor “SimJob_Ker_sum” performs a summation of all source-specific kernels. This actor is adapted from the generic “SimJob” actor by setting the executable to be a parallel Fortran code, which sums all the source-specific kernels to produce the gradients and performs a simple spatial averaging to smooth the gradients. The actor “SimJob_Descent” is also adapted from the “SimJob” actor, with the executable set to be a parallel Fortran code that computes the descent direction P_k using Eqs. (17) and (18). It takes the smoothed gradients generated by

“SimJob_Ker_sum” and the smoothed gradients and the descent direction from the previous iteration as input and generates the descent direction for the current iteration. The “LineSearch” actor is a composite actor, within which there is a loop over the source index s . Within each iteration, a “SimJob_FW” and a “SimJob_Misfit” are performed using the earth structure model $m_k + P_k$, where P_k has been generated by “SimJob_Descent.” The misfits for all sources are then summed to calculate the value of the objective function $\chi(1)$. Once $\chi(1)$ is obtained, the optimal step length α_k is computed using the quadratic interpolation. The actor “SimJob_Update” simply computes the updated earth structure model using the current descent direction, the optimal step length, and the current earth structure model. At this point, a convergence criterion, which is based on the norm of the model perturbation (i.e., $\|\alpha_k P_k\|$) and the estimated reduction of the objective function, is tested. If the convergence criterion is met, the top-level workflow terminates, otherwise more iterations in “Ramp_k” are executed.

4 An example application

The observed seismograms are from an active-source seismic survey at an undisclosed location in central Nevada, the United States. The region is dominated by typical Basin and Range geology with range fronts next to large valleys. The extension and crustal stretching produce large normal faults extending deep in the crust. Under favorable conditions, these deep fault structures can act as conduits for migration of geothermal fluids. The objective of the seismic survey is to image the fault structures that are often buried and do not have surface expressions. An initial velocity model is obtained using the travel-times of the first-arriving P -waves through wave-equation travel-time tomography based on the adjoint method. This initial velocity model is then used in depth migration to map detailed subsurface fault structures. The seismic line is about 5.36 km long. Eighty inline dynamite shots with 220-feet (67.1 m) spacing were recorded by the full spread of 160 110-feet-spaced (33.55 m) receivers.

The modeling volume, which is about 6 km long, 1 km wide, and 3.4 km deep with its length parallel to the seismic line and its width perpendicular to the seismic line, is discretized into a uniform mesh with 5.6-m grid-spacing, which results in about 122 million grid points. The synthetic seismograms were computed for 3.5 s with a sampling interval of 0.00046 s and have a dominant frequency at around 20 Hz. Both the synthetic and the observed seismograms were band-pass filtered using a 4th-order Butterworth filter with corner frequencies at 1 Hz and 20 Hz. Each finite-difference wave-propagation simulation

Fig. 4 An example of adjoint wave-equation travel-time tomography for a seismic line in central Nevada, the United States. **a** A cross section view of the *P*-velocity of the 1D starting model. The color bar indicates the range of the *P*-velocity. The seismic velocities increase smoothly with depth. **b** A histogram of the first-arrival travel-time misfit measurements for the starting model. The objective function for the starting model is $\chi = 70.2$ (s^2). **c** A cross section view of the *P*-velocity after 21 iterations of the adjoint method. **d** A histogram of the first-arrival travel-time misfit measurements for the improved model. The objective function for this model is $\chi = 1.4$ (s^2)

uses 2,048 computing cores and takes around 15 min on an IBM Blue Gene/P supercomputer. Without any spatial and temporal decimation, the disk space needed for storing the spatial–temporal volumes of the forward and adjoint strain fields is about 10 TB per shot. Since these strain fields were deleted by the “CleanUp_Wav” actor immediately after the source-specific kernels were obtained, the maximum amount of disk space used by the whole calculation never exceeded around 10 TB. The amount of disk space needed for the calculation can be further reduced by introducing some spatial–temporal decimation or using more sophisticated data compression algorithms. A one-dimensional seismic velocity model with *P*-wave speed increasing smoothly with depth is used as the starting model. A cross section of the starting model along the seismic survey line is shown in Fig. 4a. Synthetic seismograms are computed in the reference seismic velocity model using the finite-difference method, and cross-correlation travel-time shifts

between the observed and synthetic first-arriving *P*-waves are measured. The cross-correlation time-windows were automatically determined to be from the first-arrival picks to 0.05 s after. The first-arrival picks were provided to us with the observed seismograms. In each iteration, the synthetic seismograms and the cross-correlation travel-time measurements were re-computed for the reference seismic velocity model of the current iteration. Each iteration requires 4 simulations per shot, which amounts to around 0.164 million core-hours. After 21 iterations (around 3.4 million core-hours) the objective function reduced about 98 % and the *P*-wave velocity model is shown in Fig. 4c. Due to the high sampling density of this particular dataset, we did not apply any spatial smoothing to the gradients during each iteration, which causes the rapid variation close to the surface in the inverted velocity model. This problem can be removed by applying some spatial smoothing to the gradients. The maximum *P*-wave velocity

perturbation relative to the starting model is more than 65 %. Histograms of the travel-time misfits for the starting model and for the final model are shown in Fig. 4b, d. This example demonstrates that through multiple iterations, the adjoint method can effectively account for the nonlinearity in travel-time tomography.

5 Discussion and conclusions

In this study, we have successfully automated the wave-equation travel-time tomography based on the adjoint method using the “Kepler” scientific workflow package. We gave a brief summary about the adjoint method and its application on wave-equation travel-time tomography. The complete calculation process can be organized into different components, which can then be encapsulated using Kepler actors. Composite actors were used to encapsulate nested workflows. Different types of Kepler directors were used to coordinate the actions of different actors at different levels. We also demonstrated the capability of our automated wave-equation travel-time tomography package using a real active-source dataset collected in central Nevada, US. In this example, only first-arriving P-waves were used in the inversion because of the simplicity in windowing the first-arriving P-waves from the seismograms. Our package is also applicable to any seismic phases on the seismograms once a more sophisticated automatic waveform selection algorithm (e.g., Lee and Chen 2013) is integrated into our package.

In addition to providing automation for complex scientific calculations, Kepler and some other scientific workflow software packages can also help to optimize the calculation process by exposing and exploiting various forms of parallelism inherent in data-driven applications. For example, in our application the calculation of source-specific kernels for different sources can be carried out in parallel on multiple computer clusters, as the source-specific kernels are independent of each other. The workflow environment also encourages the reuse of composite actors when automating a scientific calculation. In our application, the “SimJob” composite actor was reused by other composite actors at different levels. With some modifications, it can also be reused to automate job submission and management in other scientific applications. The entire design of the workflow for an application actually also serves as an explicit documentation of the entire calculation process, which can facilitate better communication and collaboration among researchers and also improve the reproducibility of calculation results.

References

- Aki K, Richards PG (2002) Quantitative seismology. University Science Books, Sausalito, p 700
- Bamberger A, Chavent G, Hemons Ch, Lailly P (1982) Inversion of normal incidence seismograms. *Geophysics* 47(5):757–770
- Chen P (2011) Full-wave seismic data assimilation: theoretical background and recent advances. *Pure Appl Geophys* 168(10):1527–1552
- Chen P, Jordan TH, Zhao L (2007) Full three-dimensional tomography: a comparison between the scattering-integral and adjoint-wavefield methods. *Geophys J Int* 170(1):175–181
- Dahlen FA, Tromp J (1998) Theoretical global seismology. Princeton University Press, Princeton, p 1025
- Fichtner A, Kennett BLN, Igel H, Bunge H-P (2009) Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods. *Geophys J Int* 179:1703–1725
- Lee E, Chen P (2013) Automating seismic waveform analysis for full 3-D waveform inversions. *Geophys J Int* 194(1):572–589
- Lee E, Chen P, Jordan T, Wang L (2011) Rapid full-wave centroid moment tensor (CMT) inversion in a three-dimensional earth structure model for earthquakes in Southern California. *Geophys J Int* 186:311–330
- Liu Q, Gu R (2012) Seismic imaging: from classical to adjoint tomography. *Tectonophysics*. doi:10.1016/j.tecto.2012.07.006
- Luo Y, Schuster GT (1991) Wave equation travel-time inversion. *Geophysics* 56:645–653
- Olsen KB (1994) Simulation of 3-D elastic wave propagation in the Salt Lake Basin. PhD Dissertation, University of Utah, Salt Lake City, USA
- Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C, the art of scientific computing, 2nd edn. Cambridge University Press, New York, p 994
- Tape C, Liu Q, Maggi A, Tromp J (2010) Seismic tomography of the southern California crust based on spectral-element and adjoint methods. *Geophys J Int* 180:433–462. doi:10.1111/j.1365-246X.2009.04429.x
- Tarantola A (1984) Inversion of seismic reflection data in the acoustic approximation. *Geophysics* 49(8):1259–1266
- Tarantola A (1988) Theoretical background for the inversion of seismic waveforms, including elasticity and attenuation. *Pure Appl Geophys* 128(1/2):365–399
- Tromp J, Tape C, Liu Q (2005) Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophys J Int* 160:195–216
- Zhao L, Jordan TH (1998) Sensitivity of frequency-dependent traveltimes to laterally heterogeneous, anisotropic Earth structure. *Geophys J Int* 133:683–704
- Zhao L, Jordan TH, Chapman CH (2000) Three-dimensional Fréchet differential kernels for seismic delay times. *Geophys J Int* 141:58–576
- Zhao L, Jordan TH, Olsen KB, Chen P (2005) Fréchet kernels for imaging regional earth structure based on three-dimensional reference models. *Bull Seismol Soc Am* 95(6):2066–2080. doi:10.1785/0120050081
- Zhao L, Chen P, Jordan TH (2006) Strain Green’s tensors, reciprocity, and their applications to seismic source and structure studies. *Bull Seismol Soc Am* 96(5):1753–1763. doi:10.1785/0120050253