ARTICLE IN PRESS

SoftwareX ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

SoftwareX

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

ASKI: A modular toolbox for scattering-integral-based seismic full waveform inversion and sensitivity analysis utilizing external forward codes

Florian Schumacher*, Wolfgang Friederich

Institute of Geology, Mineralogy and Geophysics, Ruhr-University Bochum, Germany

ARTICLE INFO

Article history:
Received 7 July 2016
Accepted 31 October 2016

Keywords:
ASKI
Seismic full waveform inversion
Waveform sensitivity kernels
Object-oriented programming

ABSTRACT

Due to increasing computational resources, the development of new numerically demanding methods and software for imaging Earth's interior remains of high interest in Earth sciences. Here, we give a description from a user's and programmer's perspective of the highly modular, flexible and extendable software package ASKI – Analysis of Sensitivity and Kernel Inversion – recently developed for iterative scattering-integral-based seismic full waveform inversion. In ASKI, the three fundamental steps of solving the seismic forward problem, computing waveform sensitivity kernels and deriving a model update are solved by independent software programs that interact via file output/input only. Furthermore, the spatial discretizations of the model space used for solving the seismic forward problem and for deriving model updates, respectively, are kept completely independent. For this reason, ASKI does not contain a specific forward solver but instead provides a general interface to established community wave propagation codes. Moreover, the third fundamental step of deriving a model update can be repeated at relatively low costs applying different kinds of model regularization or re-selecting/weighting the inverted dataset without need to re-solve the forward problem or re-compute the kernels. Additionally, ASKI offers the user sensitivity and resolution analysis tools based on the full sensitivity matrix and allows to compose customized workflows in a consistent computational environment. ASKI is written in modern Fortran and Python, it is well documented and freely available under terms of the GNU General Public License (http://www.rub.de/aski).

1. Motivation and significance

In the context of Earth sciences, researchers as well as industrial companies have a natural interest in more accurate imaging methods which can be applied to the increasing amounts of available seismic data. Software implementing such new methods have an increased demand of computational resources on high-performance computing systems which, however, become available more easily nowadays.

The imaging method of seismic full waveform inversion (FWI) aims at utilizing the complete information content of measured seismic waveforms for deriving an earth model. Established methods iteratively derive a series of models $\mathbf{m}^1, \mathbf{m}^2, \dots, \mathbf{m}^n, \dots$ converging to the solution of the inverse problem by minimizing a waveform misfit criterion. Starting off with an initial model $\mathbf{m}^0$ of sufficient quality, in each iteration $n \geq 1$ first the seismic forward problem is solved, i.e. seismic wave propagation is simulated with respect to model $\mathbf{m}^{n-1}$ assuming the mechanisms of the involved seismic sources as known (or inverting for source properties jointly). On the basis of the observed residual between the measured seismic waveforms and the synthetic ones computed with respect to model $\mathbf{m}^{n-1}$, then a model $\mathbf{m}^n$ is derived which best reduces the misfit criterion in use. One group of currently used methods are based on the (pre-conditioned) conjugate gradient of the misfit functional with respect to the model parameters [1–4]. Another group of currently used methods minimize the misfit criterion by Newton-like [5–7] or Gauss–Newton methods [8–11] which utilize (approximations of) higher order derivatives of the misfit functional with respect to the model parameters for deriving a model update. These generally have faster convergence properties than gradient-based methods but can be subject to higher computational costs. Established FWI codes (for gradient-based as well as Newton-like or Gauss–Newton methods) infer derivatives of the misfit criterion by combination of the wavefield

* Corresponding author.
E-mail address: florian.schumacher@rub.de (F. Schumacher).

http://dx.doi.org/10.1016/j.softx.2016.10.005
2352-7110/

Please cite this article in press as: Schumacher F, Friederich W. ASKI: A modular toolbox for scattering-integral-based seismic full waveform inversion and sensitivity analysis utilizing external forward codes. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.10.005

originating from the seismic source with the wavefield of back-propagated residuals originating from the receiver positions. Thus, solving the forward problem, i.e. simulating seismic wave propagation, is strongly interwoven with computing the derivatives and is usually implemented in the same code which thereby has a rather monolithic character.

Seismic FWI is a complex problem that requires demanding numerical computations as well as handling of large amounts of data on high-performance computing systems. Thereby, complex workflows arise that need to be handled by researchers in a consistent and flexible way. From a geophysical point of view, FWI applications may have a wide range in terms of scale (from global to ultra-sonic), considered wave types and frequencies. Hence, it is desirable to have modular and extendable, thus efficient, solutions to FWI. Nowadays, new developments follow this approach and try to establish the above stated inversion strategies within integrated systems or toolboxes providing flexibility in choosing inversion methods and in general follow modularized approaches to solving the seismic inverse problem [12–16].

As one variety of Gauss–Newton FWI, the scattering-integral (SI) method [9,17] is particularly suitable for modularization, but is not considered by one of the above stated modular approaches. The fundamental steps of solving the forward problem and deriving a model update can be naturally decoupled, since the computation of the involved derivatives of waveform data with respect to the model parameters (called *waveform kernels*) is done by combination of the wavefield originating from the seismic source with Green's functions originating from the receiver positions that are independent of actual measured seismograms. Green's functions can be re-used in different source–receiver combinations serving as generalized backpropagations. This motivates to pre-compute the required wavefields and store them to hard disk before computing the waveform kernels. As a consequence, it becomes possible to solve the forward problem independently using established wave propagation codes, which are connected to the inversion algorithm by a suitable interface. Hence, this approach allows to independently develop inversion concepts and regularization methods on one hand, and to develop the in general demanding forward codes, e.g. with the objective of computational performance, on the other hand.

Furthermore, the general separation of solving the forward problem and computing the waveform kernels/deriving a model update, strongly suggest to introduce independent spatial model descriptions for solving the forward problem and for approaching the inversion step, as this is highly beneficial for the overall regularization of the inverse problem and hence the convergence of the iterative solution (also compare [18, sec. 3.2]). In Schumacher et al. [11] we chose this novel approach also in order to make scattering-integral-based FWI more computationally feasible. Naturally, a very modular inversion process arises that we implemented in the software package ASKI in an accordingly modular object-oriented fashion. ASKI stands for *Analysis of Sensitivity and Kernel Inversion* and offers the user a platform to solve various seismic FWI problems as well as resolution and sensitivity analysis within a modular, internally consistent, flexible and extendable computational environment.

In this paper, we describe the functionalities that ASKI offers, how these are implemented, how a researcher may use and possibly extend ASKI, and which benefits and challenges arise from the modular structure of ASKI for both, users and developers.

2. ASKI in general

ASKI is a toolbox for sensitivity and resolution analysis as well as for solving FWI problems in an iterative fashion by the SI method based on waveform sensitivity kernels. These kernels constitute a connection between waveform data samples and model values by quantifying how a certain data sample changes if a certain model parameter value is perturbed. For more details on the waveform sensitivity kernels used by ASKI and formulae how to compute them, we refer to Schumacher et al. [11, esp. appx. A2]. The computation of the kernels requires spectral wavefields originating from seismic sources and, independently, Green's functions originating from the receiver components. The scattering-integral-based waveform inversion implemented by ASKI is conceptually of very modular nature due to a very strict organizational separation of the three basic steps of solving the forward problem (called "stage I" in [11, sec. 3]), computing waveform sensitivity kernels ("stage II") and deriving a model update ("stage III"). Based on the sensitivity kernels computed at stage II, any sensitivity and resolution analysis can be conducted, having the full sensitivity matrix at hand. These three stages are illustrated in Fig. 1.

ASKI does not solve the seismic forward problem internally, but instead provides interfaces to existing forward codes to compute the required wavefields. Supported forward codes are, at the moment, the 1D semi-analytical code Gemini [21] and the 3D spectral-element code SPECFEM3D [22] for both, Cartesian and spherical framework, as well as the 3D nodal discontinuous-Galerkin code NEXD [23]. Extension to other forward codes is planned.

In order to make scattering-integral-based waveform inversion computationally more feasible and to approach the inverse problem in a more natural way based on the resolving power of the inverted seismic data, ASKI uses a volumetric spatial representation of the model space (called *inversion grid* in ASKI) that is independent of the model description for solving the forward problem, which is assumed by ASKI to be a point grid and is called *wavefield points* (cp. [11, sec. 3.1]). Very different kinds of inversion grids are provided by ASKI, accounting for complexity and geometrical scale of the particular inverse problem to solve. Additionally, we suggest in Schumacher et al. [11, sec. 3.2] to do the inversion step in the frequency domain, which is why ASKI computes frequency-domain sensitivity kernels from spectral wavefields provided by the forward codes.

At stage III, the inversion procedure allows to account for regularization terms of the misfit criterion to be optimized and to discard particular data samples of the data set or apply a specific weight to each datum. It is even possible to alter the misfit criterion as a whole, at this stage. ASKI, therefore, provides options to apply any regularization conditions to the inversion step that are representable as linear equations of the model update values, in particular smoothing and damping. At relatively low costs the computation of a model update can be repeated applying different regularization or data weighting/selection.

3. ASKI from a user's perspective

Fig. 2 shows a simplified workflow of main ASKI operations. The software package ASKI consists of numerous independent executables and scripts that communicate by input/output of files and can be composed to customized workflows of iterative FWI as well as sensitivity and resolution analysis. ASKI is controlled by input parameter files and operated by calling the executables.

For a particular workflow of FWI or sensitivity/resolution analysis, a user must set a parameter file that specifies all general information that will not change throughout iterations of full waveform inversion (if there are any) and from which locations of all files and directories used by the workflow can be inferred. Therefore, it is called the main parameter file (Fig. 2, Ⓐ) and it is required as input to almost all ASKI executables. Along with some conventions on nomenclature, all files required by an executable

Please cite this article in press as: Schumacher F, Friederich W. ASKI: A modular toolbox for scattering-integral-based seismic full waveform inversion and sensitivity analysis utilizing external forward codes. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.10.005

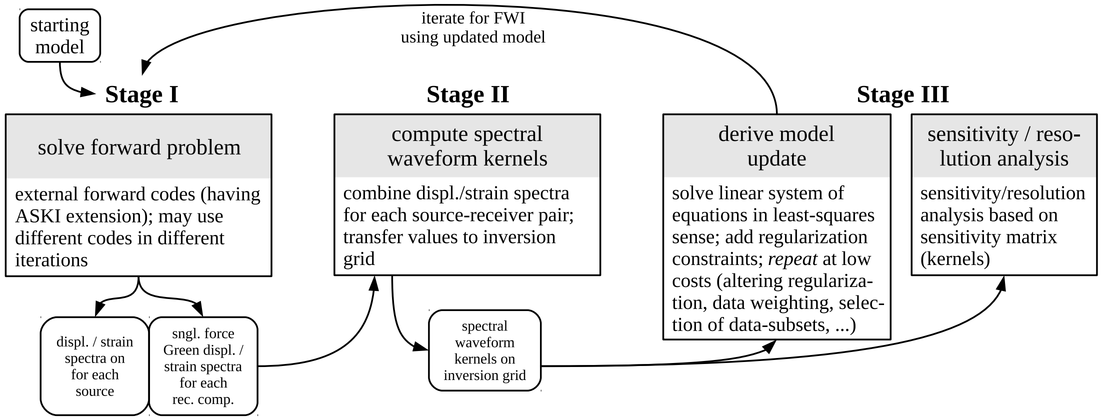

Fig. 1. Simplified scheme of the three separate stages of an iteration of SI FWI, described in more detail in Schumacher et al. [11, sec. 3]. Boxes with sharp corners indicate operations conducted by (a series of) autonomous software programs. Rounded corners indicate independent objects represented by files on hard disk.

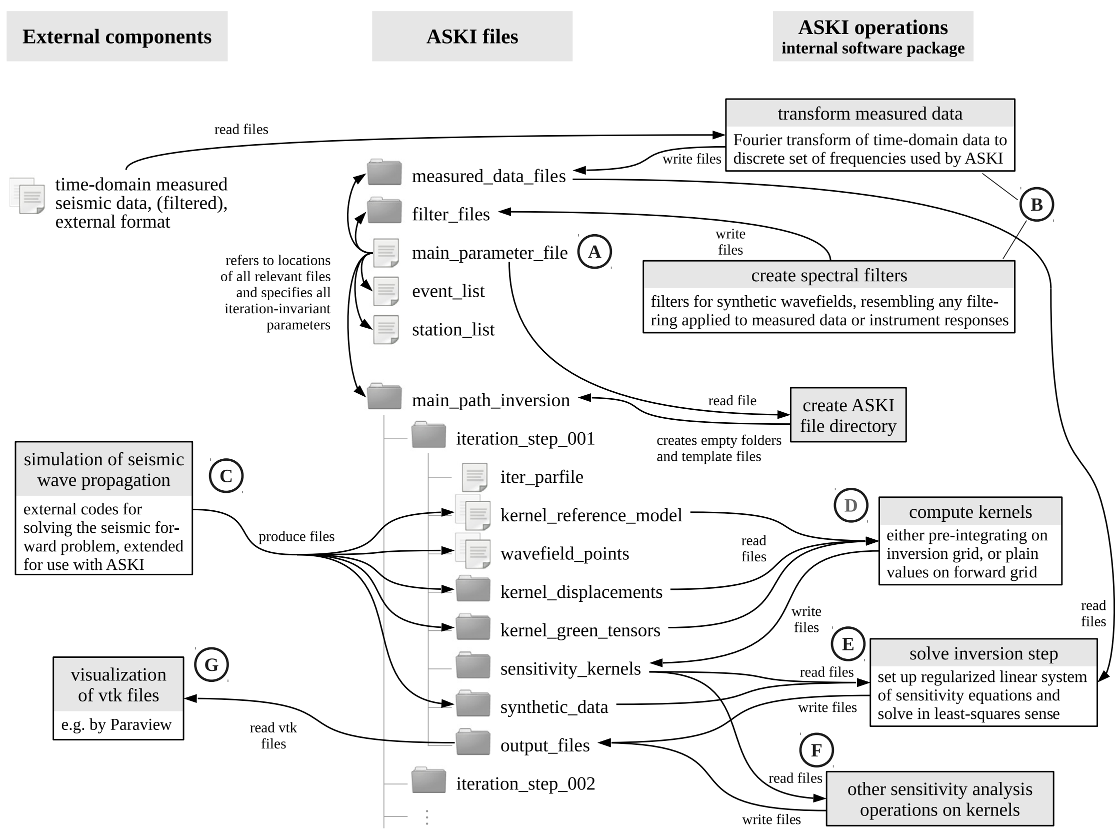

Fig. 2. Simplified workflow of main ASKI operations, displaying the directory tree used by ASKI in an iteration of FWI (center), as well as external (left) and ASKI-internal (right) operations writing/reading files to/from it. Circled letters are referred to in Section 3.

can thus be located on the file system. All seismic sources and receivers involved in this workflow need to be specified by text files in a simple pre-defined format. Since ASKI works in the frequency domain, any waveform data to be inverted or filters used in the operations need to be Fourier-transformed at specifically chosen discrete frequencies. ASKI provides executables for these tasks (Fig. 2, B) and supports basic formats for seismic data such as text trace files and Seismic Unix [24].

The user needs to choose a forward code by which the wavefields required for kernel computation should be computed. The particular choice may depend on the geophysical complexity that should be accounted for in forward modeling (e.g. 1D or 3D acoustic or visco/poro-elastic medium, local Cartesian model domain or accounting for Earth's curvature/gravity/rotation). Sufficient experience is necessary how to operate the forward code in general, as well as its specific features of producing output for ASKI. These output files must be written to a designated directory referred to by the main parameter file (Fig. 2, C). This requires large amounts of storage and significant output operations, but combining the wavefields for different source–receiver pairs when computing kernels by the SI method may result in an overall optimized number of simulations to be done, dependent on the involved number of sources and receivers [9].

ASKI currently supports computation of spectral waveform kernels for isotropically elastic model parametrizations. Kernels can be computed selectively only for those source–receiver combinations for which there are data in the dataset, possibly at individual frequencies. Calling the respective excecutable, the wavefield files are read in and the computed kernel files are stored to their designated directory (Fig. 2, D). By the formulae

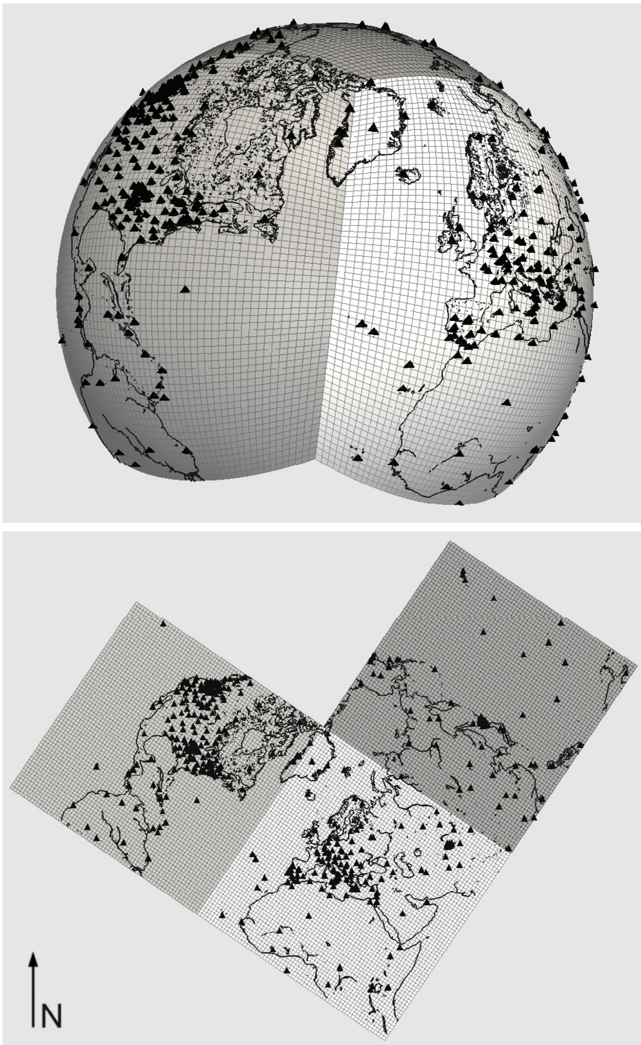

Fig. 3. Paraview screen shots of two different representations of the same spherical inversion grid consisting of 3 90-degree chunks of a cubed sphere (cp. [19], here colored in white, light gray and dark gray, respectively) covering the northern hemisphere. Also plotted are shore lines and an example network of seismic station locations (black triangles). All quantities in ASKI, such as wavefields, kernels or models on point or volume grids, as well as station/earthquake locations or shore lines are plotted consistently in the chosen projection: By inversion-grid-specific transformation routines, namely, all involved point coordinates (e.g. corners of hexahedra, points on line segments, etc.) are transformed from their actual location in space to their coordinates in the requested graphical representation. Note that only the surface of the volumetric inversion grid is visible here, which actually extends to depth. *Top:* global spherical representation, including Earth's curvature. *Bottom:* representation without curvature, northing referring to the center of the white chunk.

for isotropically elastic waveform kernels [11, Eq. A5], eighteen displacement and strain components are superposed to just three kernel values (for two elastic constants and density). Additionally the kernels are pre-integrated onto the coarser inversion grid. That is why they require significantly less disk space than the wavefields on which they are based (only around 3%, for instance, in the example inversion shown by [11, table 1]).

Once kernels are computed, they can be used for deriving an update of the isotropically elastic earth model parameters in an iteration of FWI (Fig. 2, Ⓔ–Ⓕ). Smoothing and/or damping conditions can be added as additional equations to the linear system of sensitivity equations. These equations relate the unknowns (i.e. the model update values) in a specific way, e.g. forcing them to be small (damping) or requesting them to represent some average of their neighboring values (smoothing). For every model update value, the particular constraints are in general allowed to vary with location inside the model domain. Therefore, the intensities of smoothing and damping can be reduced in areas of good data coverage, thus increasing the influence of the seismic data in places where they have more resolutional power. Applying this kind of regularization is independent of the preceding stages of solving the forward problem and computing kernels, which are also much more computationally expensive than deriving a model update by solving an overdetermined system of equations in a least-squares sense. Thus, a user can play with smoothing and damping intensities and derive different model updates at relatively low costs. In addition, particular data can be down-weighted or discarded at this stage in order to reduce unwanted influence by data that can only be fitted insufficiently or to counteract model artifacts evoked e.g. by earthquake clusters. This provides the user with additional flexibility for deriving a satisfactory updated model.

Other sensitivity and resolution analysis operations based on kernel values may as well read in the kernel files, do their analyses and produce output in the ASKI directories, according to the needs of the user (Fig. 2, Ⓕ). ASKI additionally provides executables for minor auxiliary operations, e.g. to generate a starting model in the format required by ASKI or to create vtk files [25] from kernel, wavefield or model values for plotting with external software such as Paraview [26], VisIt [27], MayaVi [28], or VTK [25] (Fig. 2, Ⓖ). Also all involved grids or point sets are generated in form of vtk files for visualization, e.g. wavefield points, station/event coordinates, lines between event and station locations, or shorelines. All vtk output files of different quantities can be generated with respect to different geometrical projections, e.g. applying certain rotations or removing curvature in spherical settings in order to create convenient views to inspected objects in a consistent way. Fig. 3 gives an example of the same plot using two different projections. For plotting numerical data on the inversion grid, it can additionally be decided whether the vtk files should contain data on the volumetric cell geometry or on a point grid only, namely the center points of the inversion grid cells. The latter can be advantageous when applying some kind of display filters/interpolations by the visualization tools and reduces the overall geometry information in the vtk files, hence the file size.

The toolbox-like modular way of applying the software ASKI as a user strongly suggests a form of modularity in the user documentation. The ASKI user manual, therefore, is structured according to a modular concept by providing a top-down approach of information flow: For each workflow (e.g. FWI or transforming kernels to time domain for better human inspection, or other sensitivity/resolution analysis), the manual provides a compact list of operations to be done, referring to later sections of the manual where descriptions of the individual operations can be found. Since these individual basic operations (such as setting up parameter files, transforming data, solving the forward problem, computing (pre-integrated) kernels, etc.) occur in different ASKI workflows, we thus avoid redundancy of documentation. For even more detailed information about file formats or special executables, we refer the interested user to yet later sections of the manual. This way we try to focus the user only on relevant information in a modular piece-by-piece fashion according to the actually conducted workflow. We hope that this way, new users being confronted for the first time with ASKI do not become discouraged, and experienced users benefit from quickly reminding themselves about the basic operations of a particular workflow. Even though hyperlinks are generated within the pdf manual and we use page references in the text by which a user can easily jump around in a printed version, certainly other forms of documentation are better suited for such kind of modular/linked user documentation, such as linked web documents viewed in a web browser.

The ASKI project website http://www.rub.de/aski intends to provide basic information and literature recommendation,

Please cite this article in press as: Schumacher F, Friederich W. ASKI: A modular toolbox for scattering-integral-based seismic full waveform inversion and sensitivity analysis utilizing external forward codes. SoftwareX (2016), http://dx.doi.org/10.1016/j.softx.2016.10.005

as well as a reference to ASKI's source repository (currently https://github.com/seismology-RUB/ASKI). Via the source repository, ASKI and some of its components, as well as documentation and some examples are freely available under terms of the GNU General Public License (version 2 or higher http://www.gnu.org/licenses/gpl).

4. ASKI from a programmer's viewpoint

The toolbox-like nature of ASKI requires the individual operations to communicate via input/output of files on hard disk and encourages the toolbox to be implemented in an object-oriented fashion that enables flexible maintenance of the code and convenient extension of the software package. Fig. 4 sketches modularization aspects of ASKI which are discussed in the following in the context of the implementation of ASKI.

Using external forward codes requires a suitable interface to ASKI. Since in general there are very different (semi-)analytical or fully numerical seismic forward codes operating in time or frequency domain using very different numerical schemes and we do not want to a priori disallow any, we decided to have a very general, thus flexible interface constructed in an object-oriented fashion (Fig. 4, A): From a particular forward code, all forward-code-dependent quantities such as wavefield points (i.e. the forward grid), kernel reference model (i.e. the background model on the forward grid used to solve the seismic forward problem), kernel displacement (i.e. the spectral wavefields originating from the seismic sources) and kernel Green tensor (i.e. the back-propagations in form of single force Green tensor components originating from the seismic station components) are communicated to ASKI through an individual sub-module. In order to extend ASKI to support another forward code, a specific submodule needs to be created for each such quantity (as indicated by Fig. 4, A). This way, ASKI allows the forward codes to define their own grid points on which they provide the wavefields and they can use their own file formats for points, model, wavefields and other meta information they might use, provided the knowledge of how to access the required information, e.g. how to read any files or to calculate certain data, is implemented in the respective sub-modules. Any particular implementational advantages of a forward code can thus be maintained without imposing unnecessary requests on the code. For instance, grid-based forward codes should choose some subset of the simulation grid as wavefield points for ASKI and the forward code's standard (parameter) files for grids and meta information can be re-used by the interface sub-modules. However, all forward codes most likely need to be extended or modified in order to provide the required spectral wavefield output, synthetic data in the required frequency discretization, as well as point/model information.

Using independent spatial descriptions for solving the forward problem and doing the inversion step raises the following challenge. Not only need information to be transferred from the external forward codes into ASKI (as described above), but also information about the new model derived in an iteration of FWI need to be communicated from ASKI to the forward code for solving the seismic forward problem in the beginning of the next iteration of FWI (Fig. 4, G). Since the inversion grid used by ASKI is completely independent of the spatial model description used by the forward code, an additional extension of each forward code is required in form of a suitable interpolation method. For nodal forward codes, 3D-unstructured interpolation methods based on Shepard [29] can solve this problem, which is applied in ASKI for SPECFEM3D and NEXD.

Another aspect of the modularity of SI-based FWI that clearly suggests object-oriented implementation is having different types of inversion grids accounting for different geometrical settings and the generally locally varying resolutional power of the dataset. ASKI provides several types of volumetric inversion grids for spherical/Cartesian and simple/sophisticated applications which may have cells of hexahedral and/or tetrahedral shape (some examples are shown in Fig. 5). Each inversion grid is realized as a submodule implementing a well defined interface to the inversion grid parent module (Fig. 4, B). Thus, support for new types of inversion grids, providing special meshing features for a particular inversion problem can be added easily to the ASKI software package. Even (subsets of) volumetric grids used by element-based forward codes can be re-used as ASKI inversion grids, as we implemented in ASKI for the forward codes SPECFEM3D_Cartesian and SPECFEM3D_GLOBE.

In order to connect arbitrary sets of wavefield points with arbitrary volumetric inversion grids, ASKI first locates all points of the forward grid inside the inversion grid cells. This requires a convention about the coordinates communicated by the forward grid submodule of the used forward code to the chosen inversion grid submodule. ASKI uses global Cartesian coordinates for both, spherical and Cartesian inversion grids. Internally, ASKI uses standardized geometries of inversion grid cells, which are in case of hexahedral cells the cube

$$\left\{(x,y,z) \in \mathbb{R}^3 \,\middle|\, -1 \le x,y,z \le 1\right\} \tag{1}$$

and in case of tetrahedral cells the tetrahedron

$$\left\{(x,y,z) \in \mathbb{R}^3 \,\middle|\, x,y,z \ge 0\,;\; x+y+z \le 1\right\} \tag{2}$$

spanned by the standard unit vectors in $\mathbb{R}^3$. After locating the wavefield points inside a cell, the point coordinates are transformed to the respective hexahedral or tetrahedral standard cell (Fig. 4, C). For the (non-dimensional) point coordinates with respect to the standard cell, quadrature rules are calculated for integration on the cell and the resulting integration weights are multiplied by the (dimensionalized) Jacobian of the transformation to the original cell shape. This way, the sensitivity kernels computed on the wavefield points can be integrated onto the volumetric cells of the inversion grid (Fig. 4, D), evaluating the scattering-integral on disjoint subvolumes of the earth. This constitutes the required transition from forward to inversion grid. Due to the modular implementation, different kinds of quadrature rules are supported by ASKI ranging from simple averaging to sophisticated quadrature after Levin [30]. Easily, further integration rules can be added to the software package. For the forward codes SPECFEM3D_Cartesian and SPECFEM3D_GLOBE, for instance, ASKI supports to use the SPECFEM quadrature rules for integration onto the spectral element grid when using it as an inversion grid. New element-based forward codes may as well support this functionality and re-use their own quadrature rules for pre-integration of the kernels.

The inversion grid module, furthermore, provides the geometry information for writing vtk files, dependent on the chosen visual projection that is independent of the actual absolute location of inversion grid cells in space. To this end, the particular inversion grid sub-module transforms a given set of point coordinates to the desired projection location (Fig. 4, E). Extending a particular inversion grid sub-module, a programmer can easily add support for further geometrical projections.

Extending ASKI to support not only isotropic model parametrizations but anisotropic or even anelastic models for kernel computation and inversion requires additional implementation in different parts of the code. An additional model parametrization needs to be defined in terms of its name, its number of parameters, their names and units. ASKI handles this information by a designated software module. Additionally, and most importantly, new routines in the kernel module need to be provided with formulae for these new parameters. In case the new parameters should also be used for forward modeling in proceeding iterations

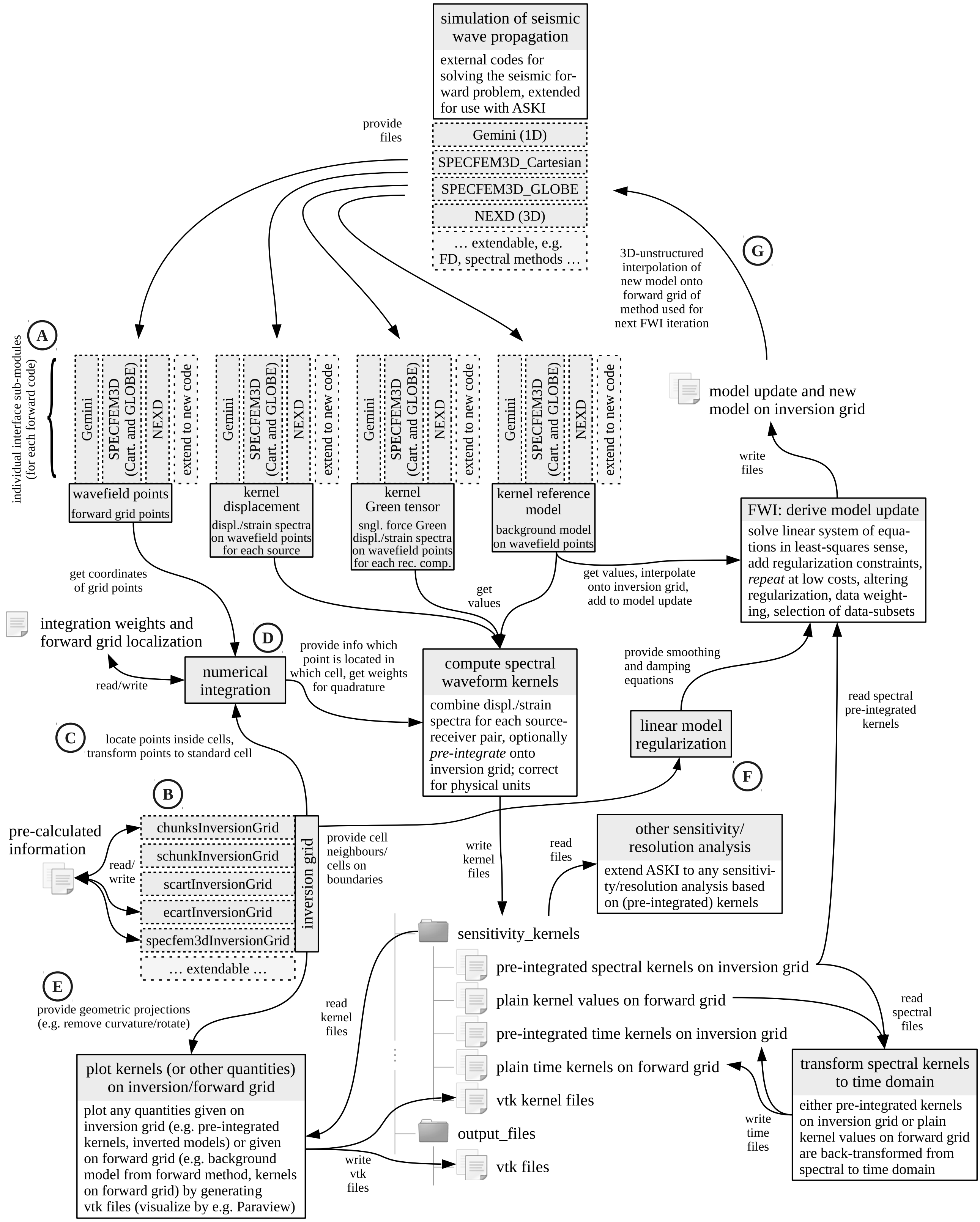

Fig. 4. Simplified structure of software package ASKI with focus on its modular connectivity. Circled letters are referred to in Section 4. Note that certain aspects are not referred to here, e.g. synthetic data (produced by the external wave propagation codes) and measured data both additionally required for deriving a model update in FWI, as well as rather technical modules, e.g. for handling earth model parametrizations, etc.

of FWI, an external forward code must be chosen for FWI which in general supports this parametrization and which in particular has an ASKI interface that is able to communicate model values of this parametrization from ASKI to the forward code. Nonetheless, ASKI's modular code structure well prepares for extending ASKI to model parametrizations other than isotropically elastic ones.

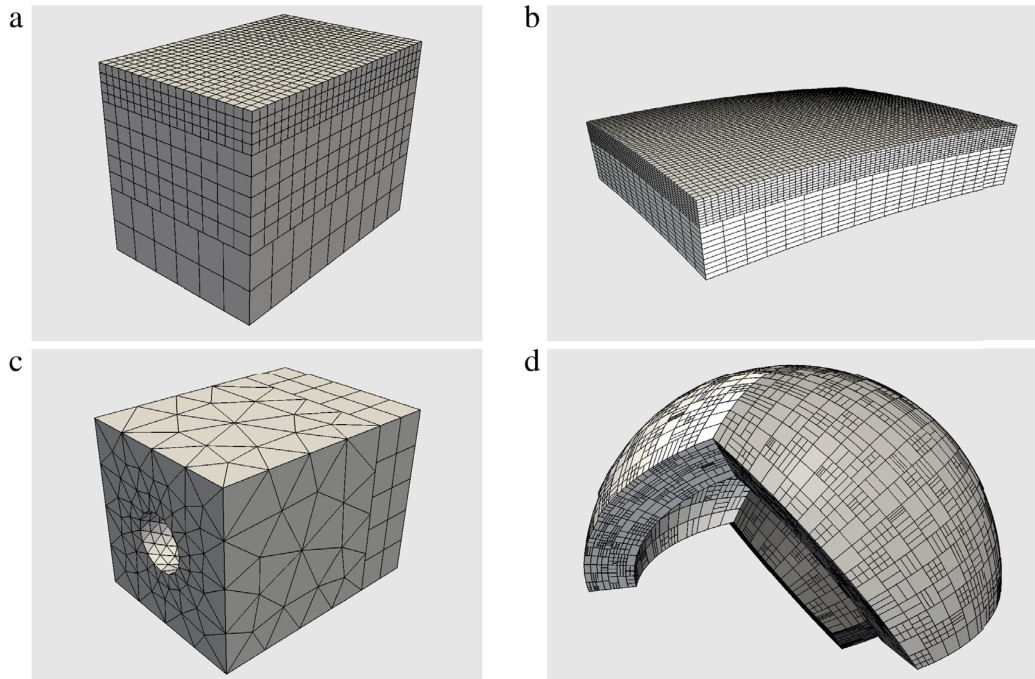

Fig. 5. Examples of 4 different inversion grids: (a) and (b) show a simple Cartesian and spherical inversion grid, respectively, that allow for laterally homogeneous cell sizes and depth-dependent refinement. (c) is an example of a mixed tetrahedral-hexahedral mesh externally generated by Trelis 15.1 [20] which can be used to generate sophisticated Cartesian and spherical inversion grids. (d) is an example of a spherical inversion grid consisting of 3 90-degrees chunks of a cubed sphere [19]. This more sophisticated spherical inversion grid may be built from two, three or six 90-degrees chunks of a cubed sphere, or a single chunk with arbitrary angular extension between 0° and 90° and allows for adaptive refinement of the hexahedral inversion grid cells (here a random refinement, for illustration). Also supported by ASKI, but not shown here, is an external inversion grid consisting of (a subset of) the spectral element grid used by the forward solvers SPECFEM3D_Cartesian and SPECFEM3D_GLOBE. In combination with the complete set of forward grid points inside the selected elements, it is possible to use the SPECFEM3D quadrature rules in ASKI for pre-integration of the kernels.

Having all kernel values available as files on hard disk, it is possible to implement different serial or parallelized realizations of setting up and solving the (large) regularized system of linear sensitivity equations (e.g. by factorization methods or conjugate-gradient solvers) in an iteration of FWI. ASKI manages regularization constraints in a separate software module which also controls the behavior of boundary cells in case of smoothing conditions, i.e. when there are missing neighboring cells on outer (or inner) boundaries of the inversion grid (Fig. 4, Ⓕ). This way, it is easy to add new types of regularization conditions that are representable as linear equations of the model update values. Due to object-oriented and modular implementation of ASKI, more functionality can be added to ASKI conveniently based on the software modules provided by ASKI (seen as a kind of library). Further workflows of sensitivity/resolution analysis based on kernels can be realized this way, such as determining optimal acquisition layouts before acquiring data, determining sub datasets that are best sensitive to a specific model region, or analyzing data and model space based on singular value decomposition of the sensitivity operator.

One challenge arising from the general modularized approach of ASKI is to deal with inconsistencies in FWI regarding the physical units of the involved quantities such as measured and synthetic data, wavefields, volume element of integration and earth model parameters. The external forward codes, namely, might not by default use standardized physical units for displacement wavefield or elastic earth model parameters. Also, measured seismic displacement data may be provided in nanometers or in the SI unit meter and different types of inversion grids for spherical and Cartesian applications may use different units of distance (e.g. meters and kilometers), resulting in different units of the volume element (cubic meters or cubic kilometers, respectively) for the integration weights. In the ASKI application example of Schumacher et al. [11, sec. 4.2], the forward code Gemini is used to invert for shear wave speed v_s, i.e. sensitivity equations of the form

$$\delta u(\mathbf{r}) = \int_{\oplus} K^{v_s}(\mathbf{x};\mathbf{r})\,\delta v_s(\mathbf{x})\,\mathrm{d}^3\mathbf{x} \tag{3}$$

are used by ASKI, relating residuals δu of seismic displacement at receiver positions $\mathbf{r}$ to model updates of shear wave speed δv_s in the model domain $\oplus$ by means of the kernel $K^{v_s}(\mathbf{x};\mathbf{r})$. Common seismological units were used for this inversion, i.e.

$$[\delta u] = \mathrm{nm}, \tag{4}$$

$$[\delta v_s] = \mathrm{km\ s^{-1}}, \tag{5}$$

$$[\mathrm{d}^3\mathbf{x}] = \mathrm{km}^3, \tag{6}$$

where $[\cdot]$ denotes the physical unit of the quantity contained in square brackets. Following Schumacher et al. [11, appx. A], the physical unit of the kernel K^{v_s} can be expressed as

$$\begin{aligned}[K^{v_s}] &= [\rho^0]\,[v_s^0]\,[\gamma]\,[e] \\ &= \frac{\mathrm{g}}{\mathrm{cm}^3}\,\frac{\mathrm{km}}{\mathrm{s}}\,\frac{10^{-12}}{\mathrm{N}}\,10^{-12} = 10^{-18}\,\frac{\mathrm{s}}{\mathrm{m}^3},\end{aligned} \tag{7}$$

where e denotes the strain of the wavefield originating from the seismic source and γ denotes the strain of the Green function originating from the receiver position which are provided by Gemini as displacement in nm per km distance and therefore have a dimensionless unit of 10^{-12}. As γ is computed for a unit force, it should have an additional physical unit of N^{-1}. Looking at Eq. (3) with regard to units as in Eqs. (4)–(7), an inconsistency by a factor of 10^3 can be observed by which values of the updated model will be incorrect. Using standard SI units for all quantities, on the other hand, results in Eq. (3) being perfectly consistent. In order to account for these kinds of inconsistencies by the object-oriented implementation of ASKI, each quantity such as wavefields and strains (also for Green functions), seismic data, model parametrization and volume element provides a number by which the units of that quantity must be divided in order to transform to standard SI units. When computing kernels in ASKI, all these numbers are gathered and consistently multiplied to the pre-integrated kernel values before writing them to file (in the above example, scaling the kernel values by 10^3).

In the ASKI source repository (currently https://github.com/seismology-RUB/ASKI), the subdirectory `devel/` contains useful tools and documentation for code developers of ASKI. In particular, the ASKI developer's manual contains a whole section on how to add support for further wave propagation codes to ASKI.

5. Conclusions

We implemented the scattering-integral-based full waveform inversion (FWI) concept presented in Schumacher et al. [11] in form of the modularized software package ASKI—Analysis of Sensitivity and Kernel Inversion. In contrast to monolithic software realizations of FWI, which are still common today, ASKI offers a modularized approach to scattering-integral-based FWI in which the three fundamental steps of solving the seismic forward problem, computing waveform sensitivity kernels and deriving a model update are solved by independent software programs that interact via file output/input. ASKI is supplemented by tools for sensitivity analysis, data processing and visualization. This allows to compose customized interactive workflows of iterative FWI as well as sensitivity and resolution analysis based on frequency-domain waveform sensitivity kernels. Most operations in ASKI are done step by step and manually by the operating scientist, which offers full control on all aspects of the seismic inverse problem. Since the FWI method implemented by ASKI has Gauss–Newton convergence (which can be significantly faster than conjugate-gradient convergence), it is feasible to benefit from this flexibility, in contrast to fully automated decisions by established conjugate-gradient methods. The forward problem is solved by external wave propagation codes connected to ASKI by a general interface and ASKI can be easily extended by experienced programmers to support further forward codes and more functionality due to an object-oriented programming approach. The benefits of setting up the complete sensitivity matrix in an iteration of FWI or using it for sensitivity or resolution analysis must be paid by storing waveform sensitivity kernels to hard disk and significant input/output operations of large amounts of data for storing and using the kernels within the modular framework of ASKI.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research through grant 03G0752C, as well as by the German Research Foundation (DFG) through the Collaborative Research Centre 526 "Rheology of the Earth – from the upper crust to the subduction zone". We are grateful for the helpful advice given by two reviewers.

References

[1] Mora P. Nonlinear two-dimensional elastic inversion of multioffset seismic data. Geophysics 1987;52(9):1211–28.
[2] Tromp J, Tape C, Liu Q. Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. Geophys J Int 2005;160(1):195–216. http://dx.doi.org/10.1111/j.1365-246X.2004.02453.x.
[3] Fichtner A, Kennett BL, Igel H, Bunge H-P. Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods. Geophys J Int 2009;179(3):1703–25.
[4] Butzer S, Kurzmann A, Bohlen T. 3D elastic full-waveform inversion of small-scale heterogeneities in transmission geometry. Geophys Prospect 2013;61(6):1238–51.
[5] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program 1989;45(1–3):503–28.
[6] Pratt G, Shin C. Hicks, Gauss–Newton and full Newton methods in frequency–space seismic waveform inversion. Geophys J Int 1998;133(2):341–62.
[7] Fichtner A, Trampert J. Hessian kernels of seismic data functionals based upon adjoint techniques. Geophys J Int 2011;185(2):775–98.
[8] Akcelik V, Biros G, Ghattas O. Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation. In: Supercomputing, ACM/IEEE 2002 conference. IEEE; 2002. p. 41.
[9] Chen P, Jordan TH, Zhao L. Full three-dimensional tomography: A comparison between the scattering-integral and adjoint-wavefield methods. Geophys J Int 2007;170(1):175–81.
[10] Epanomeritakis I, Akelik V, Ghattas O, Bielak J. A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion. Inverse Problems 2008;24(3):034015.
[11] Schumacher F, Friederich W, Lamara S. A flexible, extendable, modular and computationally efficient approach to scattering-integral-based seismic full waveform inversion. Geophys J Int 2016;204(2):1100–19.
[12] Hewett RJ, Demanet L. PySIT documentation, http://pysit.org/, 2013.
[13] Krischer L, Fichtner A, Zukauskaite S, Igel H. Large-scale seismic inversion framework. Seismol Res Lett 2015;86(4):1198–207. http://dx.doi.org/10.1785/0220140248.
[14] Métivier L, Brossier R. The SEISCOPE optimization toolbox: A large-scale nonlinear optimization library based on reverse communication. Geophysics 2016;81(2):F1–15.
[15] Princeton University (Theoretical & Computational Seismology Group), SeisStar project website, http://seisstar.github.io/SeisStar/, 2016.
[16] Princeton University (Theoretical & Computational Seismology Group), SeisFlows documentation, http://seisflows.readthedocs.org/en/latest/.
[17] Schumacher F. Modularized iterative full seismic waveform inversion for 3d-heterogeneous media based on waveform sensitivity kernels [Doctoral dissertation]. Institute of Geology, Mineralogy and Geophysics, Ruhr-Universität Bochum, urn:nbn:de:hbz:294-40511; 2014.
[18] Colombi A, Nissen-Meyer T, Boschi L, Giardini D. Seismic waveform inversion for coremantle boundary topography. Geophys J Int 2014;198(1):55–71. http://dx.doi.org/10.1093/gji/ggu112.
[19] Ronchi C, Iacono R, Paolucci PS. The "cubed sphere": a new method for the solution of partial differential equations in spherical geometry. J Comput Phys 1996;124(1):93–114.
[20] csimsoft, TRELIS 15.1 User Documentation. http://csimsoft.com/help/trelishelp.htm, 2014.
[21] Friederich W, Dalkolmo J. Complete synthetic seismograms for a spherically symmetric earth by a numerical computation of Green's function in the frequency domain. Geophys J Int 1995;122:537–50.
[22] Tromp J, Komatitsch D, Liu Q. Spectral-element and adjoint methods in seismology. Commun Comput Phys 2008;3(1):1–32.
[23] Lambrecht L. Forward and inverse modeling of seismic waves for reconnaissance in mechanized tunneling [Doctoral dissertation]. Institute of Geology, Mineralogy and Geophysics, Ruhr-Universität Bochum, urn:nbn:de:hbz:294-44164; 2015.
[24] Stockwell JW. The CWP/SU: Seismic Un*x package. Comput Geosci 1999;25(4):415–9.
[25] Schroeder W, Martin K, Lorensen B. The visualization toolkit: An object-oriented approach to 3D graphics. Kitware; 2006.
[26] Squillacote AH, Ahrens J. The paraview guide, vol. 366. Kitware; 2007.
[27] Childs H, Brugger E, Whitlock B, Meredith J, Ahern S, Pugmire D, et al. VisIt: An end-user tool for visualizing and analyzing very large data. In: High performance visualization–enabling extreme-scale scientific insight. Chapman and Hall/CRC; 2012. p. 357–72.
[28] Ramachandran P, Varoquaux G. Mayavi: 3D visualization of scientific data. Comput Sci Eng 2011;13(2):40–51.
[29] Shepard D. A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM national conference, ACM'68. New York, NY, USA: ACM; 1968. p. 517–24. http://dx.doi.org/10.1145/800186.810616.
[30] Levin D. Stable integration rules with scattered integration points. J Comput Appl Math 1999;112:181–7.