Electronic Seismologist

Cartesian Meshing Spherical Earth (CMSE): A Code Package to Incorporate the Spherical Earth in SPECFEM3D Cartesian Simulations

Guoliang Li[*1], Kai Tao[2], Min Chen[1,3], Jiaqi Li[1], Ross Maguire[1,4], and Xiaodan Ma[5]

Abstract

The SPECFEM3D_Cartesian code package is widely used in simulating seismic wave propagation on local and regional scales due to its computational efficiency compared with the one-chunk version of the SPECFEM3D_Globe code. In SPECFEM3D_Cartesian, the built-in meshing tool maps a spherically curved cube to a rectangular cube using the Universal Transverse Mercator projection (UTM). Meanwhile, the geodetic east, north, and up directions are assigned as the local *x*–*y*–*z* directions. This causes coordinate orientation issues in simulating waveform propagation in regions larger than 6° × 6° or near the Earth's polar regions. In this study, we introduce a new code package, named Cartesian Meshing Spherical Earth (CMSE), that can accurately mesh the 3D geometry of the Earth's surface under the Cartesian coordinate frame, while retaining the geodetic directions. To benchmark our new package, we calculate the residual amplitude of the CMSE synthetics with respect to the reference synthetics calculated by SPECFEM3D_Globe. In the regional scale simulations with an area of 1300 km × 1300 km, we find a maximum of 5% amplitude residual for the SPECFEM3D_Cartesian synthetics using the mesh generated by the CMSE, much smaller than the maximum amplitude residual of 100% for the synthetics based on its built-in meshing tool. Therefore, our new meshing tool CMSE overcomes the limitations of the internal mesher used by SPECFEM3D_Cartesian and can be used for more accurate waveform simulations in larger regions beyond one UTM zone. Furthermore, CMSE can deal with regions at the south and north poles that cannot be handled by the UTM projection. Although other external code packages can be used to mesh the curvature of the Earth, the advantage of the CMSE code is that it is open-source, easy to use, and fully integrated with SPECFEM3D_Cartesian.

Cite this article as Li, G., K. Tao, M. Chen, J. Li, R. Maguire, and X. Ma (2022). Cartesian Meshing Spherical Earth (CMSE): A Code Package to Incorporate the Spherical Earth in SPECFEM3D Cartesian Simulations, *Seismol. Res. Lett.* **XX**, 1–11, doi: 10.1785/0220210131.

Supplemental Material

Introduction

The SPECFEM3D_Globe and SPECFEM3D_Cartesian code packages are powerful tools for simulating seismic wave propagation in realistic Earth models and performing full-wave inversion based on the adjoint method (e.g., Tromp *et al.*, 2005; Fichtner *et al.*, 2006; Liu and Tromp, 2006). The SPECFEM3D code packages employ the highly accurate spectral element method (SEM) to solve the seismic wave equation in the 3D heterogeneous Earth models (e.g., Faccioli *et al.*, 1997; Komatitsch and Tromp, 1999). The main advantage of SEM is that it combines the flexibility of the finite-element method with the accuracy of the pseudospectral method (e.g., Boore, 1972; Lysmer and Drake, 1972; Tessmer *et al.*, 1992; Garatani *et al.*, 2000; Komatitsch *et al.*, 2004). This makes the SEM more accurate than the widely used finite-difference method in honoring the surface topography and implementing the free-surface condition, which is important in simulating the surface wave propagation (Graves, 1996; Robertsson,

1. Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, Michigan, U.S.A., https://orcid.org/0000-0001-8601-8946 (GL); https://orcid.org/0000-0001-7525-5401 (JL); https://orcid.org/0000-0002-0822-8849 (RM); 2. College of Geophysics, China University of Petroleum at Beijing, Beijing, China, https://orcid.org/0000-0003-0269-3763 (KT); 3. Department of Earth and Environmental Sciences, Michigan State University, East Lansing, Michigan, U.S.A.; 4. Department of Geology, University of Maryland, College Park, Maryland, U.S.A.; 5. Research Institute of Petroleum Exploration and Development, PetroChina Tarim Oil Field Company, Korla, China

*Corresponding author: liguolia@msu.edu

TABLE 1
Comparison Parameters

Codes	Number of Spectral Element Method Elements	Elements on the Top Surface (s)	Minimum Period Resolved (s)	Used Cores	CPU Main Frequency (GHz)	Compiler and Version	Used Time Step (s)	Elapsed Time (s)
CUTM	350,288	×192	5.7	1212×	2.1	ifort (18.0.2)	0.045	380
CMSE	350,288	×192	5.7	1212×	2.1	ifort (18.0.2)	0.045	380
Global	1,590,336	×192	5.9	1212×	2.1	ifort (18.0.2)	0.045	1584

1996; Komatitsch and Tromp, 1999). Because of its accuracy and efficiency, the SPECFEM3D codes have become commonly used computational tools and have been widely applied in earthquake and structural seismology, for example, modeling hazardous ground shaking (Komatitsch *et al.*, 2004) and iteratively improving complex 3D velocity models (e.g., Fichtner *et al.*, 2009; Tape *et al.*, 2010; Chen *et al.*, 2015; Tao *et al.*, 2018). More specifically, the SPECFEM3D_Globe package, hereinafter referred to as the Global code, is designed to simulate wave propagation on global and regional scales, whereas the SPECFEM3D_Cartesian package is for simulations on regional and local scales. Here, we refer to the local scale as small regions with areas less than 6° × 6°, in which a flat surface is accurate enough to approximate the curved Earth surface.

Although the two code packages can calculate surface wave propagation on all scales, neither of them is ideal for simulating surface waves on a regional scale. For the Global package, the Earth's ellipsoidal geometry is considered; however, the meshing domain is by default extending from the free surface down to the inner core. When modeling surface waves at the regional scale (e.g., for ambient noise adjoint tomography), the maximum depth of the simulation domain only needs to be several tens of or a few hundred kilometers (e.g., Chen *et al.*, 2014; Liu *et al.*, 2017; Wang *et al.*, 2018). Therefore, the Global mesh extending down to the inner core unnecessarily increases the memory requirement and computational time, which leads to unnecessary but costly computations, particularly for regional adjoint tomography in which forward and adjoint simulations need to be solved many times. As our tests show, the computational time of the Global code is approximately four times that of the Cartesian code (shown in Table 1) in a model volume of 1300 km × 1300 km × 300 km. In addition, in the Global package, the maximum number of mesh layers in the crust is set by default as less than three, which is too coarsely meshed to capture the fine structures within the crust, such as thin sedimentary layers.

Compared with the Global code, the SPECFEM3D_Cartesian package is more flexible in meshing and relatively easier for the users to define the depth of the meshing volume and the numbers of mesh layers. The computational efficiency and the flexible meshing make the Cartesian code a more preferred choice in most of the ambient noise adjoint tomography studies (e.g., Chen *et al.*, 2014; Liu *et al.*, 2017; Wang *et al.*, 2018). However, the ellipsoidal Earth surface is not accounted for with the built-in meshing tool of the Cartesian package. The built-in mesher uses the Universal Transverse Mercator (UTM) method to project the spherically curved cube to a rectangular cube as shown in Figure 1a,b. This built-in meshing tool is referred to as the CUTM from hereon. The CUTM package defines the grid's east and north directions as the local x- and y-directions of the Cartesian system and the normal direction of the free surface as the local z-direction. The Cartesian code further specifies the local x–y–z directions as the geodetic directions in the entire meshing domain. Consequently, the geodetic directions at each receiver are misoriented. This limits the usage of the Cartesian code to only the local scale, in which a flat surface is sufficient to approximate the 3D ellipsoidal Earth's free surface. Furthermore, because the distortion of UTM becomes bigger at higher latitudes, the Cartesian package is incapable of modeling wave propagation near the south and north poles.

Apart from the SPECFEM3D packages, SES3D is also widely used for simulating elastic wave propagation in 3D Earth models (Gokhberg and Fichtner, 2016). Compared with SPECFEM3D_Globe, one advantage of the SES3D package is that the depth range of the modeling volume can be customized by users. However, because SES3D employs unit-cube elements, wave speed discontinuities are not honored in the mesh. In addition, packages such as RegSEM and AxiSEM3d, Salvus can also perform simulations in spherical earth models (Nissen-Meyer *et al.*, 2014; Jacobucci, 2017; Jacobucci *et al.*, 2019; Leng *et al.*, 2019; Van Herwaarden *et al.*, 2021). Compared with SPECFEM3D, they also have their merits and demerits; however, nowadays SPECFEM3D packages are widely used in various areas in seismology.

To overcome the limitations of SPECFEM3D packages, regarding meshing, here we introduce a new code package, named Cartesian Meshing Spherical Earth (CMSE), that can accurately mesh the spherically curved cube under the Cartesian coordinate system. This new meshing tool CMSE is compatible with the solver of the SPECFEM3D_Cartesian

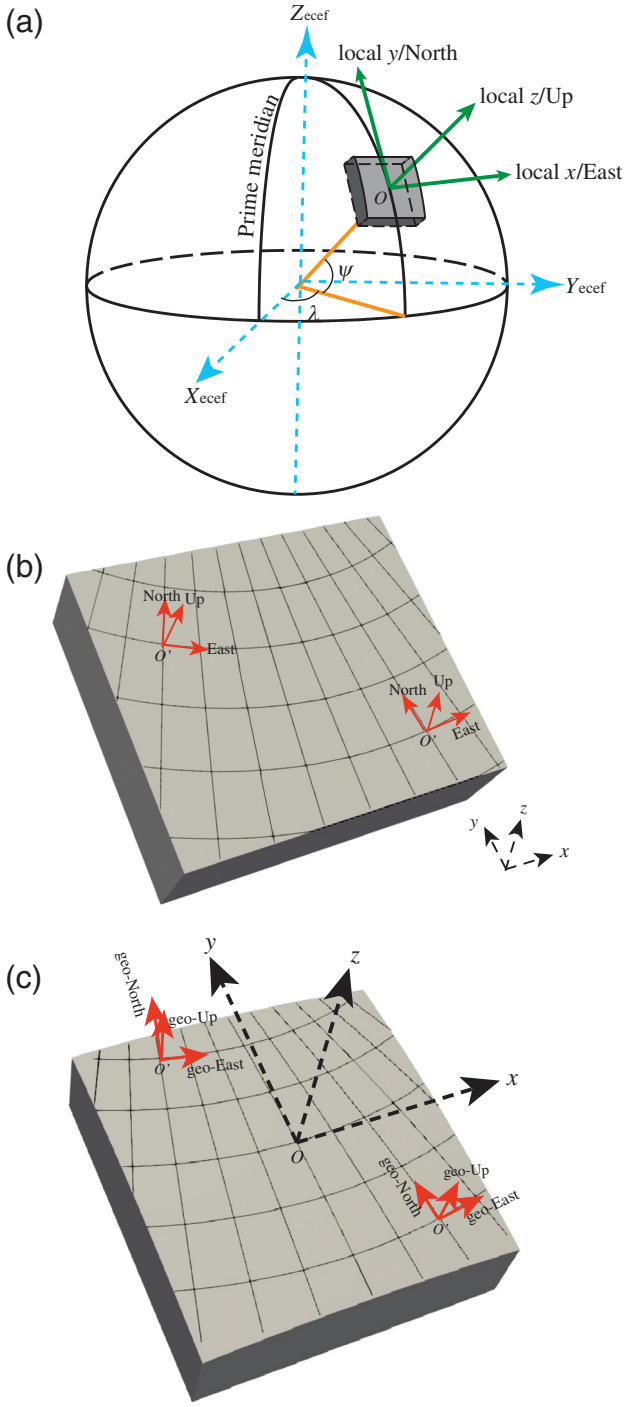

Figure 1. (a) Illustrations of the Earth-Centered-Earth-Fixed and the local *x*–*y*–*z* coordinates. The gray block represents the modeling region. The directions of local *x*–*y*–*z* are set as the geodetic directions of the reference points *O*. (b) The mesh volume of the gray region shown in panel (a) by the CUTM package. Because of the Universal Transverse Mercator (UTM) projecting, the top surface of the meshing volume is flat. (c) The mesh volume of the gray block by the Cartesian Meshing Spherical Earth (CMSE) package, in which the geometry of the ellipsoidal Earth surface is preserved. The local *x*–*y*–*z* coordinates and geodetic directions at different sites are inconsistent. In panels (b) and (c), the black lines represent the longitudinal and latitudinal lines projected to the surface the mesh volume.

code package. As proved in the following sections, it combines the flexibility of the Cartesian mesh, which makes the solver more computationally efficient, and the accuracy of Global mesh, which is more accurate in wave simulations on the regional scale, including in polar regions. In this article, we first summarize the meshing method implemented in this new package and illustrate its workflow. Then, we benchmark the simulated waveforms using this improved meshing code package and discuss the potential issues and further improvements.

Method

Mapping the free surface geometry in the Earth-Centered-Earth-Fixed (ECEF) coordinate system

The Earth-Centered-Earth-Fixed (ECEF) coordinate system is a Cartesian system with three orthogonal axes precisely describing the locations with the origin at the center of the Earth, as illustrated in Figure 1a. In this system, the geometry of the free surface of the modeling region can be extracted. It is a right-handed orthogonal coordinate system defined as follows:

1. The positive x axis goes through the location at the Earth's surface where the equator (Latitude 0°) intersects with the prime meridian in Greenwich (Longitude 0°).
2. The positive y axis passes through the location where Longitude 90° intersects with Latitude 0° to ensure the coordinate system is right handed.
3. The positive z axis extends through the true north, which does not coincide with the instantaneous Earth rotational axis.

With these defined Cartesian coordinates, the geometry of the ellipsoidal Earth can be well described for regional surface wave simulations. The mapping from the geodetic latitude ϕ and longitude λ and altitude h position to the ECEF coordinates (X, Y, Z) is well defined in geodesy (Hoffmann-Wellenhof *et al.*, 1997) with the following equations:

$$\begin{cases} X = (N(\phi) + h)\cos\phi\cos\lambda \\ Z = (\frac{b^2}{a^2}N(\phi) + h)\sin\phi \\ Y = (N(\phi) + h)\cos\phi\sin\lambda \end{cases}, \tag{1}$$

in which

$$(\phi) = \frac{a^2}{\sqrt{a^2\cos\phi^2 + b^2\sin\phi^2}} - \frac{a}{\sqrt{1 - e^2\sin^2\phi}}. \tag{2}$$

Here a and b are the equatorial radius (semimajor axis) and the polar radius (semiminor axis), respectively. $N(\phi)$, named as the prime vertical radius of curvature, is the distance from the surface to the x-axis along the ellipsoid normal. $e^2 = 1 - \frac{b^2}{a^2}$ is the square of the first numerical eccentricity of the ellipsoid. According to the WGS86 reference Earth model, the values

of the equatorial radius and the polar radius are 6378.1370 and 6356.7523 km, respectively, and therefore the value of e is approximately 6.7×10^{-3}. Using the earlier equations, the location and geometry of the free surface of the modeling region can be accurately mapped. However, because the origin of the coordinate is located at the center of the earth, far away from simulated regions, the relative positions of the grid points within the simulated region are not clear with the calculated ECEF coordinate values. To make the coordinate values more intuitive in indicating the relative location of the grid points, we further convert the ECEF coordinates to the local x–y–z coordinates.

The definition of the local *x–y–z* coordinates

The local x–y–z coordinates are defined with respect to a plane tangent to the Earth's surface at a specific location as the origin (or reference location), and hence it is sometimes known as a "Local Tangent" or "local geodetic" plane. As Figure 1c shows, the local east, north, and up directions are set as the geodetic longitudinal, latitudinal, and up direction at the origin. By convention, the local east direction is set as the positive direction of axis x, the local north direction as axis y, and the local up direction as axis z. To transform from ECEF coordinates to the local x–y–z coordinates, we need a local reference point as the origin, typically this is the location of a point on the Earth's surface. If the reference point is located at (X_r, Y_r, Z_r) and another point is at (X_p, Y_p, Z_p) with the ECEF coordinates, then the vector pointing from the reference point to the other point in the local x–y–z coordinates is defined as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & \cos\lambda_r & 0 \\ -\sin\phi_r\cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r \\ \cos\phi_r\cos\lambda_r & \cos\phi_r\sin\lambda_r & \sin\phi_r \end{bmatrix} \begin{bmatrix} X_p - X_r \\ Y_p - Y_r \\ Z_p - Z_r \end{bmatrix}. \tag{3}$$

Here, ϕ_r and λ_r are the geodetic latitude and longitude of the reference point, which is also the origin of the local x–y–z coordinates. With equations (1)–(3), the free surface geometry of the modeling volume is extracted and represented by local x–y–z coordinates.

Rotate the synthetic waveforms from local *x–y–z* directions to the geodetic directions

For the Cartesian package, the orientations of all three components of the synthetic waveforms are defined in the local x–y–z coordinates. However, as shown in Figure 1c, the geodetic directions at each site are different from the local x–y–z directions. Therefore, we need to further rotate the synthetic waveforms from local x–y–z directions to the geodetic directions at each recording site. To perform such a rotation with the change of variables, we first show how to define the local x–y–z coordinates with the geodetic coordinates and then build the rotation matrix by calculating derivatives of the local coordinates with respect to the geodetic coordinates. Based on equations (1)–(3), we have the following relationships by perturbing the ECEF and local coordinates at grid point (X_p, Y_p, Z_p):

$$\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & \cos\lambda_r & 0 \\ -\sin\phi_r\cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r \\ \cos\phi_r\cos\lambda_r & \cos\phi_r\sin\lambda_r & \sin\phi_r \end{bmatrix} \begin{bmatrix} dX_p \\ dY_p \\ dZ_p \end{bmatrix}, \tag{4}$$

and

$$\begin{cases} dX_p = d\{[N(\phi) + h]\cos\phi\cos\lambda\} \\ dY_p = d\{(N(\phi) + h)\cos\phi\sin\lambda\} \\ dZ_p = d\left\{\left(\frac{b^2}{a^2}N(\phi) + h\right)\sin\phi\right\} \end{cases}. \tag{5}$$

Because the geodetic directions are measured along the horizontal plane at sea level, in the earlier equation, the elevation (h) is equal to zero. Hence, in the local x–y–z Cartesian system, the coordinates of the geodetic directions at the location of $(\phi, \lambda, 0)$ are calculated by the flowing Jacobians matrix:

$$\begin{bmatrix} \frac{\partial x}{\partial\phi} & \frac{\partial x}{\partial\lambda} & \frac{\partial x}{\partial h} \\ \frac{\partial y}{\partial\phi} & \frac{\partial y}{\partial\lambda} & \frac{\partial y}{\partial h} \\ \frac{\partial z}{\partial\phi} & \frac{\partial z}{\partial\lambda} & \frac{\partial z}{\partial h} \end{bmatrix} \approx \begin{bmatrix} -\sin\lambda_r & \cos\lambda_r & 0 \\ -\sin\phi_r\cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r \\ \cos\phi_r\cos\lambda_r & \cos\phi_r\sin\lambda_r & \sin\phi_r \end{bmatrix} \times \begin{bmatrix} -R\sin\phi\cos\lambda & -R\cos\phi\sin\lambda & \cos\phi\cos\lambda \\ -R\sin\phi\sin\lambda & R\cos\phi\cos\lambda & \cos\phi\sin\lambda \\ R\cos\phi & 0 & \sin\phi \end{bmatrix}. \tag{6}$$

In the earlier equation, R is equal to $N(\phi)$. The perturbation of $N(\phi)$ with respect to latitude ϕ is neglected in equation (6) because the value of $\frac{dN(\phi)/d\phi}{N(\phi)}$ is on the order of 10^{-5}. Combining equations (4) and (6), we construct the rotation matrix to rotate the simulated seismic time series from local x–y–z directions to the geodetic directions:

$$\begin{bmatrix} E(t) \\ N(t) \\ U(t) \end{bmatrix} = \begin{bmatrix} \alpha\frac{\partial x}{\partial\phi} & \alpha\frac{\partial y}{\partial\phi} & \alpha\frac{\partial z}{\partial\phi} \\ \beta\frac{\partial x}{\partial\lambda} & \beta\frac{\partial y}{\partial\lambda} & \beta\frac{\partial z}{\partial\lambda} \\ \gamma\frac{\partial x}{h} & \gamma\frac{\partial y}{\partial h} & \gamma\frac{\partial x}{\partial h} \end{bmatrix} \begin{bmatrix} Y(t) \\ X(t) \\ Z(t) \end{bmatrix}, \tag{7}$$

in which, the $X(t)$, $Y(t)$, and $Z(t)$ mean the synthetic waveform in the local x, y, and z directions and $E(t)$, $N(t)$, and $U(t)$ represent the time series in geodetic directions. Here, the Greek alphabet α, β, and γ are the normalization coefficients to make the vectors in each row be a unit vector, for example, $\alpha = \sqrt[1/2]{(\frac{\partial x}{\partial\phi})^2 + (\frac{\partial y}{\partial\phi})^2 + (\frac{\partial z}{\partial\phi})^2}$.

Implementation

Prerequisite

Operation system: Linux, MacOS

Essential software: SPECFEM3D_Cartesian, Python 3

Python modules: getopt, importlib, NumPy, math, matplotlib, multiprocessing, obspy, pyproj, scipy, sys

Our CMSE package consists of the SPECFEM3D_Cartesian code and four Python modules. All of the Python modules are openly available via GitHub (see Data and Resources). CMSE directly meshes the 3D geometry of the earth and retains accurate geodetic information. In addition, we include a tutorial in the package that demonstrates how to use this code package (see Data and Resources). Because the SPECFEM3D_Cartesian code is already well documented online (see Data and Resources), in the following subsections, we only briefly introduce the workflow of our package and how to use the python modules.

Flow of the SPECFEM3D_Cartesian package

Here, we adopt the three-step strategy of the CUTM-based meshing tool for simulating the seismic wave propagation as shown in Figure 2. The first step involves constructing a high-quality mesh for the simulation region. In the CMSE package, this step is performed using the internal mesher xmeshfem3D. This mesher code internally uses the UTM system to project the 3D Earth surface to the 2D flat plane. Therefore, to mesh the ellipsoidal Earth geometry, we have to suppress the UTM projection by setting the flag SUPPRESS_UTM_PROJECTION as TRUE in Mesh_Par_file; then all "longitude" parameters will be defined related to the x-axis and the "latitude" parameters to the y-axis. To simulate a more realistic Earth model, interfaces such as free surface, the basement interface of the sedimentary basin, and the Moho discontinuity should be honored by the mesh. Here, we implement our Python code named GEN_interface.py (Fig. 2) to prepare these interface files before running xmeshfem3D. GEN_interface.py reads the gridded topography files of interfaces (e.g., free surface, sedimentary basement, Moho discontinuity) in the format of longitude, latitude, and altitude (or depth) (Fig. 3a). It first transforms those geodetic coordinates to the ECEF coordinates, and then further converts them to the local x–y–z coordinate as mentioned in the previous sections. Here, the free surface is also regarded as an important interface; therefore, the true 3D geometry of the ellipsoidal earth surface is preserved in this step without using any projection algorithms, which adds potential distortion. After the first step, xmeshfem3D divides the modeling volume into nonoverlapping elements. Figure 4 shows the shape of the modeling region, which is a volume resembling the cube except that its top surface is curved. Because the other five faces of the meshing volume on the four sides and the bottom are flat, Stacey and C-PML absorbing condition can be used without modifying the SPECFEM3D_Cartesian code (e.g., Clayton and Engquist, 1977; Xia *et al.*, 2004; Komatitsch and Martin, 2007; Martin and Komatitsch, 2009; Xie *et al.*, 2016).

The second step is performed using the Fortran code xgenerate_databases to create all of the necessary information needed by the SEM solver (Fig. 2), which includes the simulation type, absorbing boundary type, and velocity model.

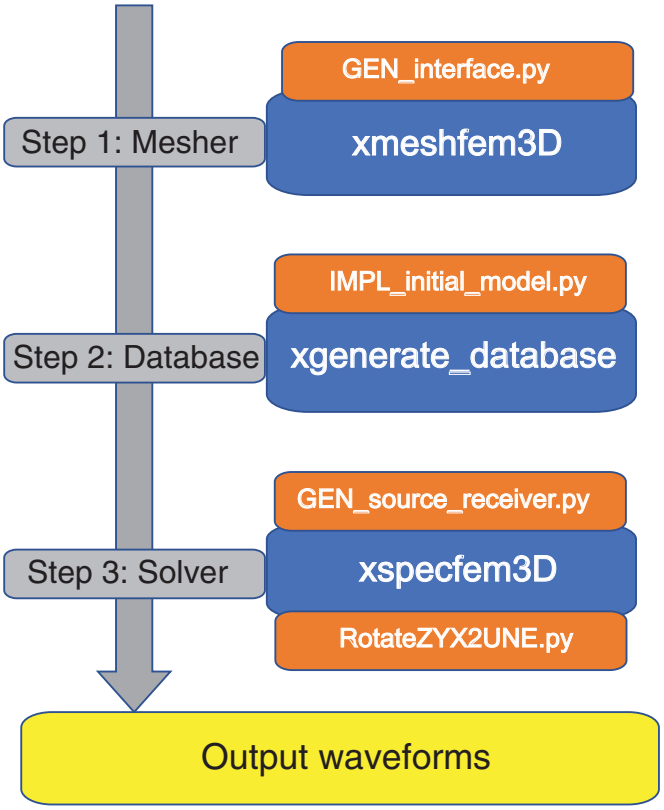

Figure 2. Flowchart of the CMSE-mesh-based SPECFEM3D_Cartesian code package. The package consists of python-based and Fortran-based codes. The python-based codes are marked by orange, whereas the Fortran-based code from the SPECFEM3D_Cartesian code is marked as dark blue.

Before running the Fortran code, the Python code IMPL_initial_model.py is designed to interpolate the velocity, density, and Q-factors of the input model to each Gauss-Lobato-Legendre (GLL) point and write out those values at each GLL point to the binary files. Those output binary files can be directly read by the xgenerate_databases by simply setting the "MODEL" parameter as "gll" in the Par_file. With the completion of this step, the values of the input model will be correctly implemented at the mesh GLL points.

The last step is to run the solver, that is, to run the simulation with specified source and receiver configurations. The locations of source and receivers are usually provided with the geodetic coordinates, which need to be converted to the local x–y–$z$coordinates based on equation (1)–(3). In addition, the directions of the forces at the source location are also given by the geodetic coordinates. Therefore, the forces also need to be converted to the values in the local x–y–z coordinate system based on equations (4)–(7) before running the xspecfem3D code. In the CMSE package, the source and receiver information is prepared by the python code GEN_source_receiver.py. After running the seismic wave simulation with Fortran code xspecfem3D, the synthetic waveforms calculated in the local x–y–z coordinates

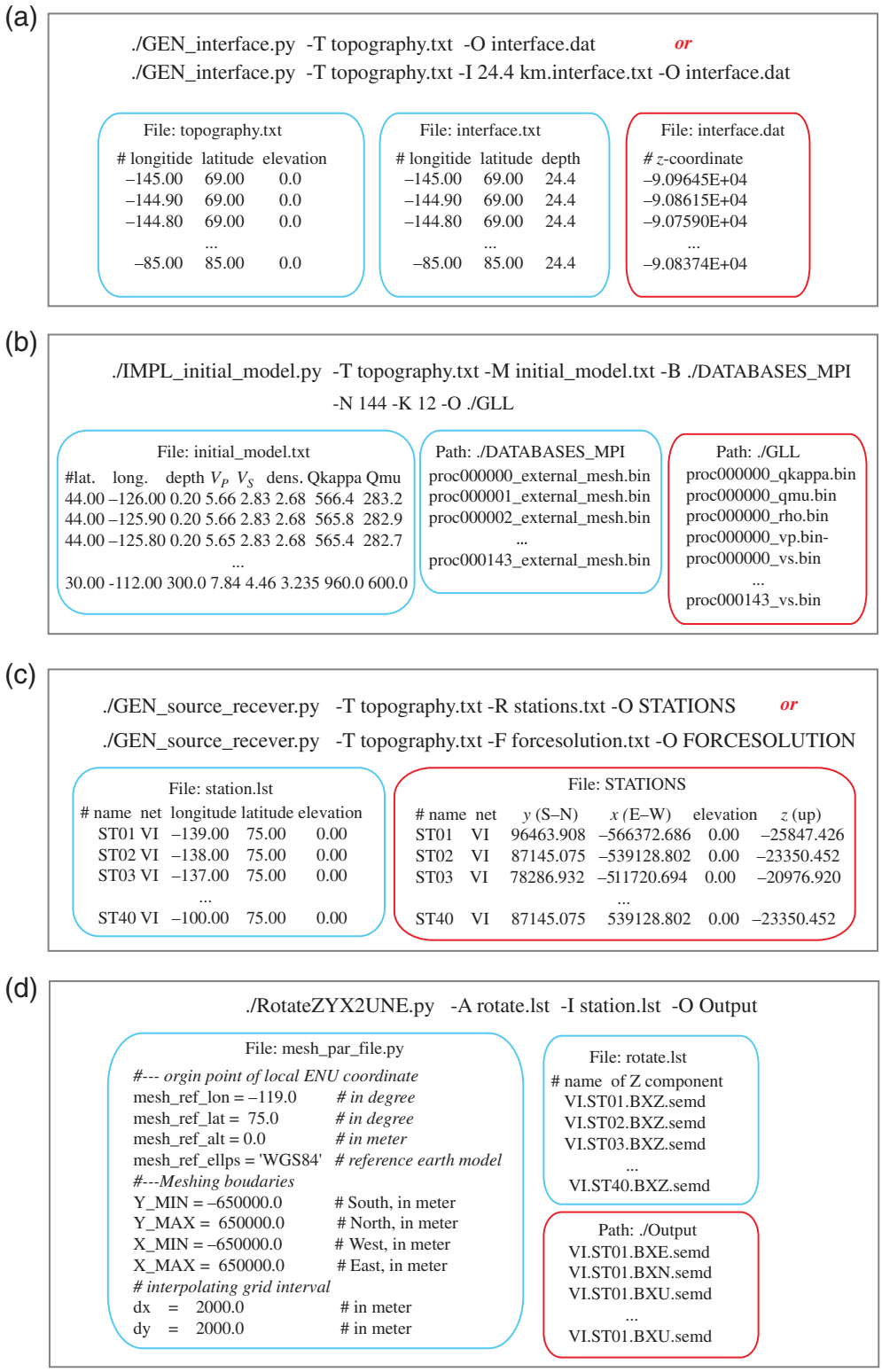

Figure 3. (a–d) Usages and file formats of the four python-based codes. The content in the blue frame shows the file format of the input files, while the content in the red frame shows the file format of output files.

will be rotated to the geodetic directions with RotateXYZ2UNE.py (Figs. 2 and 3d) with the matrix in equation (7).

Manual of the Python-based codes

Because the CUTM-meshing-based SPECFEM3D_Cartesian package has a very detailed manual, here we will only briefly describe the usage and relevant files of the Python-based codes of the CMSE package. For a more detailed description of the code, please refer to the online version. The Python codes have a common implicit input file, that is, the mesh_par_file.py (Fig. 3d), which defines the origin of the local x–y–z coordinates and boundaries of the simulated domain. For example, in Figure 3d, we chose the origin of the local x–y–z coordinates with longitude, latitude, and altitude of (−119.0°,75°,0) and defined a region with the lateral dimension of 1300 km × 1300 km to mesh. The last two rows in the mesh_par_file.py define the interpolation interval of the interfaces to be 2 km in this example.

The Python script GEN_interfaces.py prepares the interfaces for the simulated region (Fig. 3a). To construct the 3D ellipsoidal surface, apart from the implicit mesh_par_file.py, the user should provide the elevation data of the modeling region with appending parameter *-T* (topography.txt in Fig. 3a). The *-O* parameter is used to specify the name of the output file (interface.dat). To generate the internal interfaces, such as the Moho discontinuity, parameter *-I* is used to specify the interface

data file (e.g., 24.4 km interface.txt). The data format of the topography and internal discontinuity files are shown in the blue frame, and the format of the output file is shown in the red frame in Figure 3a. The Python script IMPL_initial_model.py uses multiple processors to interpolate values given at the grids of the input model to the values at the GLL points of the mesh. The input model is provided to the code with parameter *-M* with the file format of the input model shown in the blue frame of Figure 3b. The user can also specify the directory of the mesh files (-B), the total number of slices (-N), the number of processors used to do the interpolation (-K), and the directory to store the output GLL binary files (-O). GEN_source_receiver.py converts the coordinates of stations and sources from the geodetic coordinate system to the local *x*–*y*–*z* Cartesian coordinate system. The data format of the point force file forcesolution.txt is given in the SPECFEM3D_Cartesian manual. Finally, RotateZYX2UNE.py rotates the synthetic seismograms in the local *x*–*y*–*z* directions to the geodetic directions (Fig. 3d).

Numerical Validation

In this section, we design numerical forward simulations to show the accuracy of the CMSE mesh and compare its simulated waveforms with waveforms generated using both the Global and CUTM-based Cartesian packages. The Global package meshes the ellipsoidal Earth. Hence, the synthetic seismograms from the Global package can be used as references to benchmark the seismograms from the Cartesian package based on either the CMSE or the CUTM meshing schemes. For our benchmark simulations, the mesh domain is designed with a center situated at (−119° E, 75° N) and the origin of the local *x*–*y*–*z* coordinates at (−119° E,75° N,0 km) as what have been set up in Mesh_par_file.py (Fig. 3d). The mesh domain has a lateral dimension of 1300 km × 1300 km with its bottom located at a depth of 300 km in the local *x*–*y*–*z* coordinate system. We ignore the surface topography in our benchmark simulations to avoid the waveform differences introduced by different smoothing or interpolating of topographic data among the three packages. For example, the topography data used in the Global package is internally smoothed, whereas the Cartesian packages based on either CUTM or CMSE meshing incorporate the topography data specified by the user. For purposes other than code benchmarking, surface topography can certainly be implemented by both Cartesian packages as shown in Figure S1. We implement a simple three-layer 1D isotropic model in the CUTM and CMSE packages, and the details of this 1D speed model are listed in Table S1. For the Global package, we also implement this three-layer 1D mode at depth ranges shallower than 600 km, and when the depth ranges are greater than 600 km, the 1D isotropic PREM model (Dziewonski and Anderson, 1981) is implemented.

The 3D mesh domains for CMSE, CUTM, and Global code packages are illustrated, respectively, in Figure 4, and Figures S2 and S3. We use four layers to mesh the crust, another four layers to mesh the mantle with a depth of less than 80 km, and 12 layers for the deep mantle with a depth range between 80 and 300 km. The seismic source is set as isotropic and located at (−139° E, 75° N, 0 km), and three receivers are positioned at (−138° E, 75° N, 0 km), (−130° E, 75° N, 0 km), and (−104° E, 75° N, 0 km), respectively. We intentionally place the source and the furthest receiver close to the mesh domain boundaries to investigate the maximum waveform differences from different packages. The corresponding great-circle distances between the source and the receivers are 29, 260, and 999 km, respectively. To calculate the reference synthetics for the benchmark, we use the Global package to mesh the same simulation region, except that the Global mesh extends from the free surface to the inner core (Fig. S3), instead of a depth of 300 km for the Cartesian mesh domains (Fig. 4; Fig. S2). The spatial configuration of the source–receiver is the same as that of the CMSE simulation.

The synthetic waveform comparison shows that there are almost no differences between the CMSE and Global synthetics (Figs. 5; Figs. S4, S5) in the period band of 6–100 s as they totally overlap with each other (red and black lines), whereas the CUTM synthetics are very different from the Global synthetics. The time windows shown in Figure 5 mainly focus on the surface wave part, whereas those windows shown in Figures S4 and S5 concentrate on the shallow-turning *P* wave and *S* wave, respectively. The biggest amplitude residuals with respect to the Global synthetics are 5%, 2%, and 3%, respectively, for the surface wave, *P* wave, and *S* wave, which is considered an acceptable fit because it is comparable to or smaller than data measurement errors. In comparison, the CUTM synthetics largely deviate from the Global synthetics by about 20%–30% for receivers at smaller epicentral distances and up to 100% for the receiver located near the boundary (Fig. 5c). Benchmark tests prove that the synthetic waveforms calculated with the CMSE mesh are reliable even when the modeling region is larger than one 6°. Furthermore, compared with the Global package, the CMSE-meshing-based Cartesian package saves storage and is more computationally efficient because it only uses less than one quarter of the storage and the computational time of the Global package (Table 1).

Discussion

The advantages of CMSE

As mentioned earlier, the advantage of CMSE is that it incorporates the flexibility and efficiency of the Cartesian mesh and the accuracy of the Global mesh. Similar to the CUTM mesher, the CMSE mesher is flexible because the meshing depth, the number of meshing layers, and the position of doubling-layers can be easily customized by the users. Moreover, the CMSE package also preserves the efficiency of the CUTM package. As Table 1 shows, compared with the Global package, CMSE only uses one quarter of the SEM elements and the computational time. Meanwhile, the accuracy of the CMSE package

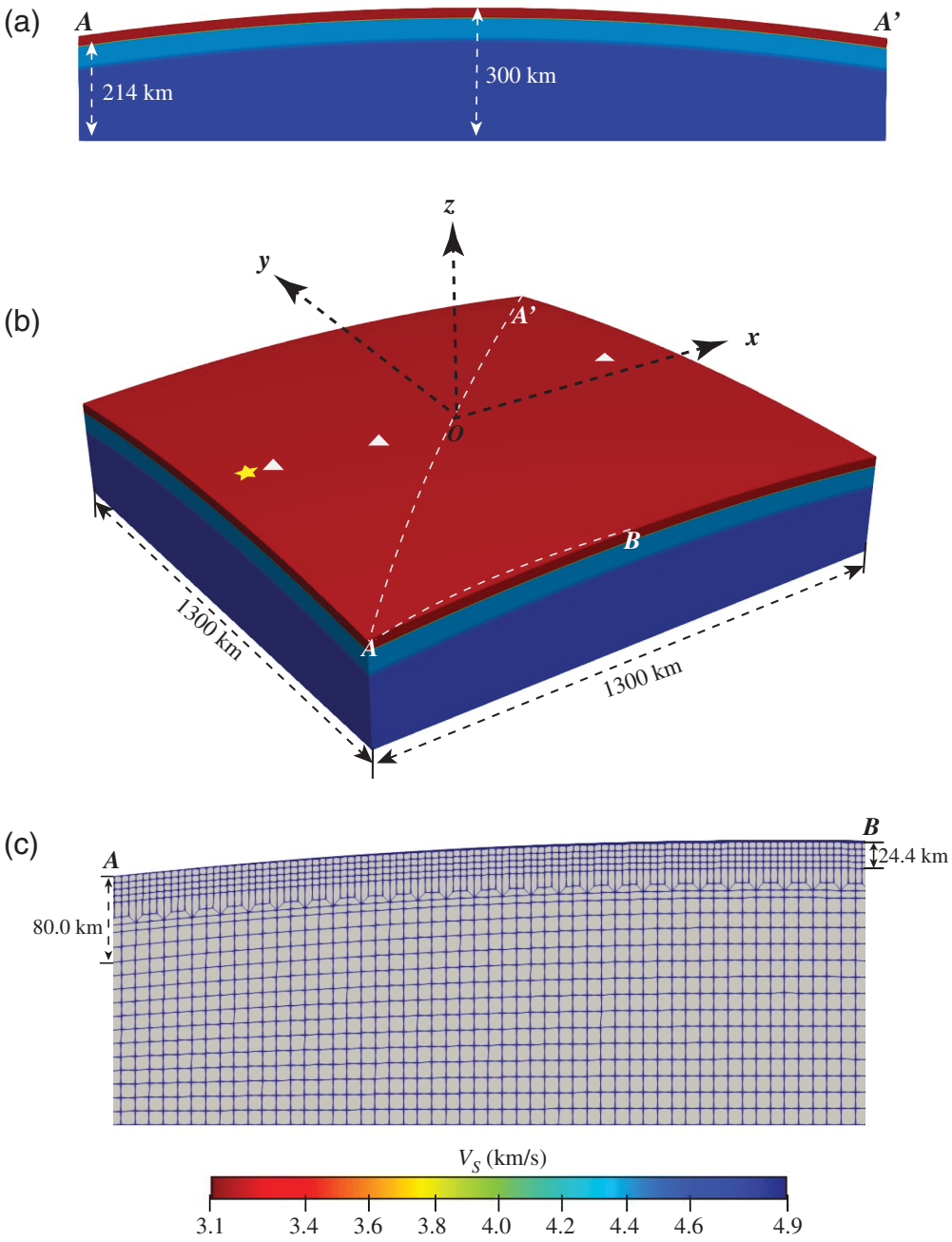

Figure 4. Spectral element mesh for the modeling region colored by shear-wave speed V_S. (a) Cross section of the profile along the diagonal line in panel (b). (b) The meshing volume of the simulated region. The source is marked as the yellow star and receivers as white triangles. Two profiles *AA'* and *AB* are indicated by the dotted-white lines. (c) A cross section along *AB* shows the detailed mesh.

is superior to that of the CUTM package on a regional scale and comparable with that of the Global package in simulating regional surface waves and body waves turning in the crust and uppermost mantle.

Furthermore, the CMSE mesher is compatible with two types of methods of dealing with absorbing boundaries, that is, the Stacey and the convolutional perfectly match layer (C-PMP) (Stacey, 1988; Clayton and Engquist, 1977; Xia *et al.*, 2004; Komatitsch and Martin, 2007; Martin and Komatitsch, 2009; Xie *et al.*, 2016). In theory, the C-PML absorbing condition is better than the Stacy absorbing boundary as the former can effectively absorb the outgoing wave energies regardless of the incident angels (Xie *et al.*, 2016), whereas the latter only works well with reflections having small incident angels (Xia *et al.*, 2004). For the CMSE mesh domain (Fig. 4), the top surface is a curve and the other five boundaries are flat planes, which allows the two built-in artificial absorbing boundaries of the Cartesian package to be used directly, especially for the C-PML absorbing condition. In the case without the absorbing boundaries, the artifacts created by the outgoing waves reflecting from the boundaries are most severe (Fig. S6a). With the Stacey absorbing boundaries, the artifacts due to the reflected waves are largely suppressed on the synthetic seismograms (Fig. S6b). The C-PML absorbing boundaries perform better than the Stacy absorbing boundaries as it is more efficient in absorbing the reflected waves.

Future improvements of CMSE

The current CMSE mesher generates a mesh volume with laterally uneven thicknesses due to the curved top surface and the flat bottom interface (Fig. 4). For example, the thickness of the mesh volume is smaller in the regions near corners or the boundaries, whereas the thickness is largest at the center of the modeling region. This can cause mesh distortion, and the distortion becomes prominent when the meshing areas become big, such as larger than 20°. One way to reduce the degree of distortions is to increase the maximum meshing

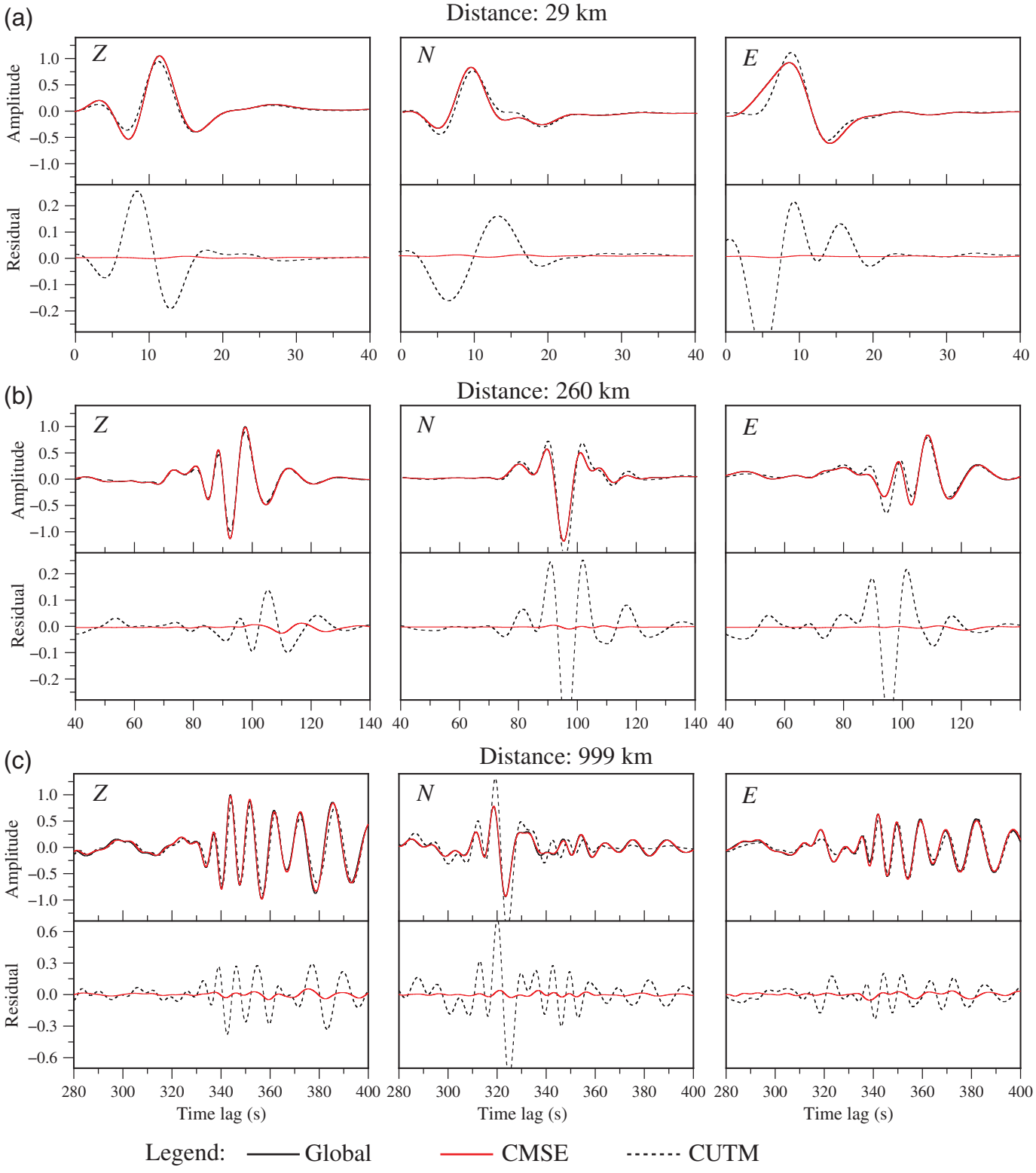

Figure 5. Waveform comparisons at three epicentral distances: (a) 29 km, (b) 260 km, and (c) 999 km. In each subfigure, the upper panel shows the synthetic waveform comparison, and the lower-panel shows the vertically exaggerated waveform residuals relative to the Global synthetics. More specifically, the red lines represent the waveforms or amplitude residuals of the CMSE synthetics, and the black dashed lines represent the amplitude residual or waveforms for CUTM synthetics. The black lines in the upper panel figure, which is almost covered by the red lines, represent the reference waveforms simulated from the global package. In each trace, the waveform amplitudes are normalized by the maximum amplitude of the *Z* components from global package. All the waveforms are filtered in a period band of 6–100 s.

depth. As a consequence, this can introduce extra unnecessary elements and computational time. However, CMSE is still a good option as long as it is still more computationally efficient than the SPECFEM3D_Globe package. One of the future improvements of the CMSE mesher is to allow the bottom surface to curve along with the top surface, similar to the mesh setup in the Global package. However, this poses a new challenge to implementing the C-PML absorbing boundaries on the curved boundaries, which is currently beyond the scope of this article.

Conclusions

In this study, to overcome the limitations arising from UTM projection used in the internal mesher of the SPECFEM3D_Cartesian package, we developed a new meshing tool, the CMSE, that can directly mesh the Earth's ellipsoidal geometry under the Cartesian coordinate system and retain the geodetic directions. Synthetic tests verify that the new package can achieve the same precision as the Global code in regional-scale simulations of surface waves and shallow turning body waves. Moreover, the CMSE mesher is more efficient compared with SPECFEM3D_Globe, using only one quarter of the number of SEM elements, which leads to the computational speedup by fourfolds in a modeling region with a volume of 1300 km × 1300 km × 300 km. Therefore, the proposed CMSE package expands the applications of SPECFEM3D_Cartesian to regional-scale forward waveform modeling and inversion, especially in the two polar regions.

Data and Resources

The supplemental material for this article includes the implemented model used in the test, the detailed mesh configurations of the SPECFEM3D_Globe and SPECFEM3D_Cartesain packages and the body waveform validations. The Cartesian Meshing Spherical Earth (CMSE) package, user manual, and example files are available on GitHub at https://github.com/LiGuoliangMSU/CMSE. The authors are glad to share the digital data, data processing scripts, and plotting scripts for the readers to reproduce the figures in this article. In addition, the manual of SPECFEM3D_Cartesian code mentioned in the Prerequisite section is available at https://geodynamics.org/cig/software/specfem3d/. Further, you are welcome to contact the author (liguolia@msu.edu) for detailed assistance on using CMSE or its application to your own work.

Declaration of Competing Interests

The authors acknowledge that there are no conflicts of interest recorded.

Acknowledgments

The authors thank the Institute for Cyber-Enabled Research (ICER) at Michigan State University, the Extreme Science and Engineering Discovery Environment (XSEDE supported by NSF Grant ACI-1053575) for providing the high-performance computing resources. The authors also sincerely thank Editor Allison Bent and the reviewer Tianchi Liu for their constructive comments and suggestions, which significantly improved the quality of this article. This research is supported by NSF Grant 1942431 and startup funds of M. C. at Michigan State University. In addition, coauthor R. M. was supported by NSF-EAR postdoctoral followship 1806412.

References

Boore, D. M. (1972). Finite difference methods for seismic wave propagation in heterogeneous materials, *Meth. Comput. Phys.* **11,** 1–37.

Chen, M., H. Huang, H. Yao, V. Rob, and F. Niu (2014). Low wave speed zones in the crust beneath SE Tibet revealed by ambient noise adjoint tomography, *Geophys. Res. Lett.* **41,** no. 2, 334–340.

Chen, M., F. Niu, Q. Liu, J. Tromp, and X. Zheng (2015). Multiparameter adjoint tomography of the crust and upper mantle beneath East Asia: 1. Model construction and comparisons, *J. Geophys. Res.* **120,** 1762–1786, doi: 10.1002/2014JB011638.

Clayton, R., and B. Engquist (1977). Absorbing boundary conditions for acoustic and elastic wave equations, *Bull. Seismol. Soc. Am.* **67,** 1529–1540.

Dziewonski, A. M., and D. L. Anderson (1981). Preliminary reference Earth model, *Phys. Earth Planet. In.* **25,** no. 4, 297–356, doi: 10.1016/0031-9201(81)90046-7.

Faccioli, E., F. Maggio, R. Paolucci, and A. Quarteroni (1997). 2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method, *J. Seismol.* **1,** 237–251.

Fichtner, A., H. P. Bunge, and H. Igel (2006). The adjoint method in seismology: II. Applications: Traveltimes and sensitivity functionals, *Phys. Earth planet. In.* **157,** 105–123.

Fichtner, A., B. L. Kennett, H. Igel, and H.-P. Bunge (2009). Full seismic waveform tomography for upper-mantle structure in the Australasian region using adjoint methods, *Geophys. J. Int.* **179,** no. 3, 1703–1725.

Garatani, K., H. Nakamura, H. Okuda, and G. Yagawa (2000). Large-scale parallel wave propagation analysis by GeoFEM, *Lect. Notes Comput. Sci.* **1823,** 445–453.

Graves, R. W. (1996). Simulating seismic wave propagation in 3D elastic media using staggered-grid finite diferences, *Bull. seismol. Soc. Am.* **86,** 1091–1106.

Gokhberg, A., and A. Fichtner (2016). Full-waveform inversion on hetrogeneous HPC systems, *Comput. Geosci.* **89,** 260–268.

Hoffmann-Wellenhof, B., H. Lichtenegger, and J. Collins (1997). *Global Positioning System: Theory and Practice*, Fourth Ed., Section 10.2.1, Springer-Verlag, 282 pp.

Jacobucci, R. (2017). regsem: Regularized structural equation modeling, *arXiv*, available at https://arxiv.org/abs/1703.08489 (last accessed December 2021).

Jacobucci, R., A. M. Brandmaier, and R. A. Kievit (2019). A practical guide to variable selection in structural equation modeling by using regularized multiple-indicators, multiple-causes models, *Adv. Meth. Pract. Psychol. Sci.* **2,** no. 1, 55–76, doi: 10.1177/2515245919826527.

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

Komatitsch, D., and J. Tromp (1999). Introduction to the spectral element method for three-dimensional seismic wave propagation, *Geophys. J. Int.* **139,** 806–822.

Komatitsch, D., Q. Liu, J. Tromp, P. Suss, C. Stidham, and J. H. Shaw (2004). Simulations of ground motion in the Los Angeles basin based upon the spectral-element method, *Bull. Seismol. Soc. Am.* **94,** no. 1, 187–206.

Liu, Q., and J. Tromp (2006). Finite-frequency kernels based on adjoint methods, *Bull. Seismol. Soc. Am.* **96,** no. 6, 2383–2397.

Liu, Y., F. Niu, M. Chen, and W. Yang (2017). 3-D crustal and uppermost mantle structure beneath NE China revealed by ambient noise adjoint tomography, *Earth Planet. Sci. Lett.* **461,** 20–29.

Leng, K., T. Nissen-Meyer, M. van Driel, K. Hosseini, and D. Al-Attar (2019). AxiSEM3D: Broad-band seismic wavefields in 3-D global earth models with undulating discontinuities, *Geophys. J. Int.* **217,** no. 3, 2125–2146, doi: 10.1093/gji/ggz092.

Lysmer, J., and L. A. Drake (1972). A finite element method for seismology, *Meth. Comput. Phys.* **11,** 181–216.

Martin, R., and D. Komatitsch (2009). An unsplit convolutional perfectly matched layer technique improved at grazing incidence for the viscoelastic wave equation, *Geophys. J. Int.* **179,** no. 1, 333–344, doi: 10.1111/j.1365-246X.2009.04278.x.

Nissen-Meyer, T., M. van Driel, S. C. Stähler, K. Hosseini, S. Hempel, L. Auer, A. Colombi, and A. Fournier (2014). AxiSEM: Broadband 3-D seismic wavefields in axisymmetric media, *Solid Earth* **5,** 425–445, doi: 10.5194/se-5-425-2014.

Robertsson, J. O. A. (1996). A numerical free-surface condition for elastic/viscoelastic ¢nite-di¡erence modeling in the presence of topography, *Geophysics* **61,** 1921–1934.

Stacey, R. (1988). Improved transparent boundary formulations for the elastic-wave equation, *Bull. Seismol. Soc. Am.* **78,** no. 6, 2089–2097.

Tao, K., S. P. Grand, and F. Niu (2018). Seismic structure of the upper mantle beneath eastern Asia from full waveform seismic tomography, *Geochem. Geophys. Geosyst.* **19,** no. 8, 2732–2763, doi: 10.1029/2018GC007460.

Tape, C., Q. Liu, A. Maggi, and J. Tromp (2010). Seismic tomography of the southern california crust based on spectral-element and adjoint methods, *Geophys. J. Int.* **180,** no. 1, 433–462.

Tessmer, E., D. Kessler, D. Kosloff, and A. Behle (1992). Multi-domain Chebyshev-Fourier method for the solution of the equations of motion of dynamic elasticity, *J. Comput. Phys.* **100,** 355–363.

Tromp, J., C. Tape, and Q. Liu (2005). Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels, *Geophys. J. Int.* **160,** 195–216.

van Herwaarden, D. P., M. Afanasiev, S. Thrastarson, and A. Fichtner (2021). Evolutionary full-waveform inversion, *Geophys. J. Int.* **224,** no. 1, 306–311.

Wang, K., Y. Yang, P. Basini, P. Tong, C. Tape, and Q. Liu (2018). Refined crustal and uppermost mantle structure of southern California by ambient noise adjoint tomography, *Geophys. J. Int.* **215,** no. 2, 844–863, doi: 10.1093/gji/ggy312.

Xia, F., L. G. Dong, and Z. T. Ma (2004). Absorbing boundary conditions for 3-D elastic wave numerical modeling, *Chin. J. Geophys.* **47,** no. 1, 149–154.

Xie, Z., R. Matzen, P. Cristini, D. Komatitsch, and R. Martin (2016). A perfectly matched layer for fluid-solid problems: Application to ocean-acoustics simulations with solid ocean bottoms, *J. Acoust. Soc. Am.* **140,** no. 1, 165–175, doi: 10.1121/1.4954736.

Manuscript received 21 May 2021
Published online 12 January 2022