TOMO3D: 3-D joint refraction and reflection traveltime tomography parallel code for active-source seismic data-synthetic test

A. Meléndez, J. Korenaga, V. Sallarès, A. Miniussi, C.R. Ranero

Geophysical Journal International

Geophys. J. Int. (2015) **203,** 158–174 doi: 10.1093/gji/ggv292

GJI Marine geosciences and applied geophysics

TOMO3D: 3-D joint refraction and reflection traveltime tomography parallel code for active-source seismic data—synthetic test

A. Meléndez,[1] J. Korenaga,[2] V. Sallarès,[1] A. Miniussi[3] and C.R. Ranero[4]

[1]*Barcelona Center for Subsurface Imaging, Institut de Ciències del Mar (CSIC), Barcelona, Spain. E-mail: melendez@icm.csic.es*
[2]*Department of Geology and Geophysics, Yale University, New Haven, CT, USA*
[3]*Observatoire de la Côte d'Azur, CNRS, UMS Galilée, Boulevard de l'Observatoire, CS* 34229, *F*-06304 *Nice Cedex* 4, *France*
[4]*Barcelona Center for Subsurface Imaging, ICREA at Institut de Ciències del Mar (CSIC), Barcelona, Spain*

Accepted 2015 July 9. Received 2015 June 22; in original form 2015 February 18

SUMMARY

We present a new 3-D traveltime tomography code (TOMO3D) for the modelling of active-source seismic data that uses the arrival times of both refracted and reflected seismic phases to derive the velocity distribution and the geometry of reflecting boundaries in the subsurface. This code is based on its popular 2-D version TOMO2D from which it inherited the methods to solve the forward and inverse problems. The traveltime calculations are done using a hybrid ray-tracing technique combining the graph and bending methods. The LSQR algorithm is used to perform the iterative regularized inversion to improve the initial velocity and depth models. In order to cope with an increased computational demand due to the incorporation of the third dimension, the forward problem solver, which takes most of the run time (∼90 per cent in the test presented here), has been parallelized with a combination of multi-processing and message passing interface standards. This parallelization distributes the ray-tracing and traveltime calculations among available computational resources. The code's performance is illustrated with a realistic synthetic example, including a checkerboard anomaly and two reflectors, which simulates the geometry of a subduction zone. The code is designed to invert for a single reflector at a time. A data-driven layer-stripping strategy is proposed for cases involving multiple reflectors, and it is tested for the successive inversion of the two reflectors. Layers are bound by consecutive reflectors, and an initial velocity model for each inversion step incorporates the results from previous steps. This strategy poses simpler inversion problems at each step, allowing the recovery of strong velocity discontinuities that would otherwise be smoothened.

Key words: Inverse theory; Controlled source seismology; Seismic tomography.

1 INTRODUCTION

Traveltime tomography is a well-established inverse modelling technique (e.g. Iyer & Hirahara 1993; Nolet 2008), which has multiple applications in the investigation of the Earth's interior. It has been applied to studies at a broad range of scales as well as for different data acquisition configurations and exploration methods, such as refraction and wide-angle reflection seismic surveys, multichannel seismic reflection records, cross-hole data or passive seismic experiments. The basic principle behind this technique is the ray theory, which is the high-frequency approximation of seismic wave propagation (Čevern ý 2001). Traveltime tomography was first devised for applications to cross-well (e.g. Bois *et al.* 1972) and seismological data (e.g. Aki & Lee 1976). Since then, traveltime tomography codes have been developed to invert for velocity using traveltime picks from either first-arrival refracted phases (e.g. Zhang & Toksöz 1998) or reflected phases (e.g. Pullammanappallil & Louie 1993). Traveltime tomography has also been adapted to use near-vertical reflection data to produce velocity and reflector depth models (e.g. Bishop *et al.* 1985; Farra & Madariaga 1988). Early implementations of the joint (i.e. simultaneous) inversion of refraction and reflection arrivals used a small number of parameters by defining velocity and depth functionals (Zelt & Smith 1992). The nonlinear tomography that simultaneously inverts both reflection and refraction traveltime data was first developed by Zhang *et al.* (1998) and is nowadays commonly used. The modelling of reflected phases caused by impedance contrasts related to geological discontinuities permits the recovery of their geometry. A fundamental advantage of the combination of both types of phases, besides the greater number of available data and the consequent enhancement of ray coverage, is the reduction of the negative effects on the inversion result produced by the ambiguity, inherent to reflection data,

between a reflector's depth and the velocity distribution above it (e.g. McCaughey & Singh 1997).

The main differences between traveltime tomography codes are found in their ray-tracing algorithms. The code of Korenaga *et al.* (2000) (TOMO2D), for example, combines the graph (or shortest path) method (Moser 1991; Cheng & House 1996) with the bending refinement method in the calculation of ray paths and synthetic traveltimes (Um & Thurber 1987). The combination of these two methods is optimal; the bending refinement requires a good initial guess to avoid local minima in traveltimes and converges to the global minimum, and the graph solution can serve as such an initial guess. Modern ray tracing is dominated by wave-front marching methods. Two main families exist: graph solvers (Moser 1991; Korenaga *et al.* 2000; Van Avendonk *et al.* 2001a; Bai *et al.* 2007) and finite-differences eikonal equation solvers (Vidale 1988; Kim 2002; Rawlinson & Sambridge 2004). The former finds minimum traveltime ray paths as a series of node connections following Fermat's principle, whereas the latter propagates wave fronts from sources following the eikonal equation and then uses them to trace back rays. The graph scheme is more flexible in terms of model parametrization. Irregular grids are straightforward to implement and geological relief can accurately be represented. For a specific parametrization, accuracy and efficiency are directly controlled by a simple definition of the node connections that are permitted. Its combination with the bending method improves accuracy with a minor loss in efficiency. In an analogy with eikonal equation solvers, ray bending would correspond to the back tracing of rays perpendicularly to the wave fronts.

There are also several approaches to the iterative solution of the nonlinear inversion problem in traveltime tomography. Two popular techniques in recent literature are regularized inversion and back-projection (for a detailed review see Rawlinson & Sambridge 2003). The first formulates the inversion as an optimization problem that seeks to minimize data misfit together with some constraints in the form of penalty functions on the roughness and size of the model perturbations, using a variety of classic gradient methods (e.g. Lutter & Nowack 1990; Sambridge 1990). The second distributes the traveltime residual among the parameters affected by the ray path proportionally to the length of their corresponding ray-path segment (e.g. Hole 1992; Zelt & Barton 1998). Between the two techniques, the former is more popular, despite being computationally more expensive, because its convergence behaviour is faster and more stable.

Technical improvements in acquisition experiments, the proliferation of research groups with a growing number of recording units and the increasing number of international collaborative projects are all favouring the collection of 3-D data in various geological settings, stimulating the extension of traveltime tomography to 3-D (e.g. Kissling 1988; Toomey & Foulger 1989; Zelt & Barton 1998; Koulakov 2009). Among the existing 3-D codes, only a few of them perform joint refraction and reflection traveltime tomography (Van Avendonk *et al.* 2001b; Hobro *et al.* 2003; Dunn *et al.* 2005; Rawlinson & Urvoy 2006). The code of Van Avendonk *et al.* (2001b) is most similar to TOMO3D, but it does not allow for the systematic investigation of the velocity–depth ambiguity intrinsic to reflection data, and there are differences in their ray-tracing algorithm. The code of Hobro *et al.* (2003) uses a ray-tracing technique (shooting) whose performance deteriorates with the increasing complexity of the model, especially in 3-D. The ray-tracing method of Dunn *et al.* (2005) (graph or shortest path method) is typically less accurate for the same computational time or slower for the same target accuracy. Rawlinson & Urvoy's (2006) FMTOMO code uses an inversion solver that is typically used in problems with a small- to moderate-sized models, although their code has been successfully applied to the combined modelling of all sorts of active and passive seismic data. The increasing size of data sets, along with the modelling of the additional third dimension, results in computationally demanding inversions, creating a need for more efficient algorithms as well as for parallel computing (Taillandier *et al.* 2009).

This paper describes our new implementation of 3-D traveltime tomography with a synthetic example. First, a methodology section introduces the new parallel 3-D tomography tool TOMO3D, which has been developed using the same principles of TOMO2D. The model parametrization and the treatment of the forward and inverse problems are described, including some detail of the parallelization of the forward problem solver. Next, the code is tested with a complex synthetic case to evaluate its performance. The layer-stripping strategy for the inversion of several reflecting interfaces is explained and applied to the recovery of two reflectors in the synthetic example. Finally, we discuss the advantages of this strategy, which provides an accurate recovery of the sharp velocity changes associated with reflecting boundaries for realistic and easy-to-interpret geological models.

2 3-D JOINT REFRACTION AND REFLECTION TRAVELTIME TOMOGRAPHY

This section provides a methodological and technical description of the newly implemented code TOMO3D. For further details, see Meléndez (2014).

2.1 Model parametrization

TOMO3D produces models of the subsurface *P*-wave velocity structure as well as of the geometry of energy reflecting boundaries. Velocity and depth models are represented by independent 3-D and 2-D node meshes, respectively. These meshes are defined by the spatial coordinates of their nodes, each of which is given a specific parameter value.

2.1.1 P-wave velocity model

The 3-D velocity mesh (Fig. 1) is defined by the spatial coordinates of its nodes, and it hangs from the seafloor and the land surface. The spacing between nodes is variable and the topology of the grid is cuboidal, and it should be finer than the smallest spatial size of velocity changes that the model is expected to account for. For the sake of accuracy, the mesh may include nodes coinciding with the locations of sources and/or receivers. For instance, in marine experiments, the grid can be designed to match the locations of all ocean bottom hydrophones/seismometers.

Each node in the mesh is assigned a value that corresponds to the *P*-wave velocity at that particular location in the subsurface. Each set of eight nodes defining a minimum-volume rectangular cuboid is called a cell.

The velocity within each cell is found by trilinear interpolation of the eight nodes (Thurber 1983). Using the local Cartesian cell coordinates (r, s, t), the interpolated velocity may be expressed as

$$\begin{aligned} u(r,s,t) = {} & (1-r)\cdot(1-s)\cdot(1-t)\cdot u_1 + r\cdot(1-s)\cdot(1-t)\cdot u_2 \\ & + (1-r)\cdot s\cdot(1-t)\cdot u_3 + (1-r)\cdot(1-s)\cdot t\cdot u_4 \\ & + (1-r)\cdot s\cdot t\cdot u_5 + r\cdot(1-s)\cdot t\cdot u_6 + r\cdot s\cdot(1-t)\cdot u_7 \\ & + r\cdot s\cdot t\cdot u_8, \end{aligned} \quad (1)$$

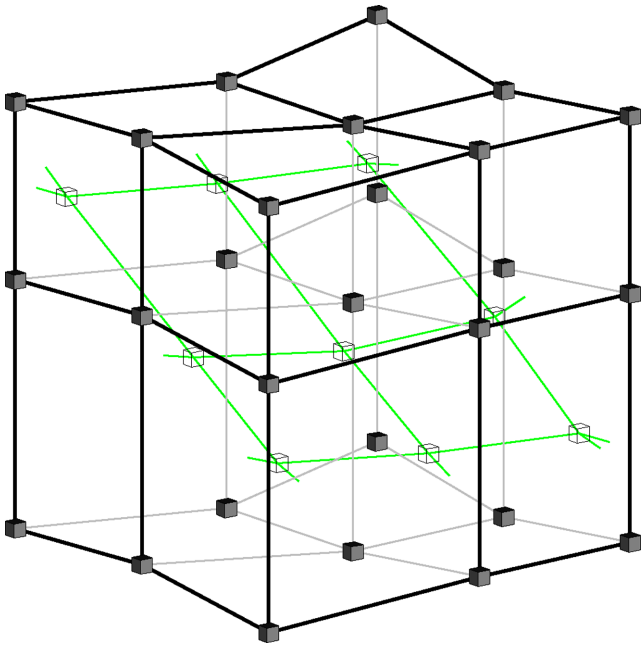

Figure 1. Portion of a 3-D velocity sheared mesh with variable spacing and a vertical shift corresponding to topography and/or bathymetry (black lines and filled nodes) crossed by a portion of a 2-D depth mesh also with variable spacing (green lines and empty nodes). Velocity and depth nodes are independent from each other.

where u_n is the value of velocity at the node with local index n. In order to account for geological relief, the mesh definition includes a vertical shift, $b(i,j)$ in eq. (2), for each $(x(i),y(j))$ nodal location. This sort of mesh is commonly known as a sheared mesh (Toomey *et al.* 1994). The geological relief (b') at any given position is defined by the bilinear interpolation

$$b' = (1-r)\cdot(1-s)\cdot b(i,j) + r\cdot(1-s)\cdot b(i+1,j) + (1-r)\cdot s\cdot b(i,j+1) + r\cdot s\cdot b(i+1,j+1). \quad (2)$$

Note that, because the mesh hangs from the bathymetry ($b' > 0$) or topography ($b' < 0$), the velocity model to be estimated does not include neither water nor air layer, and constant velocities are assumed for both layers.

2.1.2 Reflector depth model

Similarly, each node in the 2-D depth mesh is attributed the value of interface depth corresponding to its location (Fig. 1). This depth mesh is formulated as a floating reflector; its nodes can vary without modifying the velocity mesh. The velocity discontinuities typically associated with reflecting interfaces are included following the data-driven layer-stripping strategy described in Section 3.3.

As with 3-D velocity cells, 2-D depth cells are defined to bilinearly interpolate the depth at any position as

$$d(r,s) = (1-r)\cdot(1-s)\cdot d_1 + r\cdot(1-s)\cdot d_2 + (1-r)\cdot s\cdot d_3 + r\cdot s\cdot d_4, \quad (3)$$

where d_n is the depth at the node with local index n, and r and s denote the local coordinates.

2.2 Forward problem

In TOMO3D, the forward problem is solved using a combination of two ray-tracing methods: the graph method (Moser 1991) and the bending method (Moser *et al.* 1992a; Fig. 2). This hybrid approach to ray tracing, inherited from Korenaga *et al.* (2000), is based exclusively on Fermat's principle and is similar to those presented by Papazachos & Nolet (1997) and Van Avendonk *et al.* (1998). The graph method originates in network theory (e.g. Dijkstra 1959; Gallo & Pallottino 1986). In a medium represented as a mesh, this method finds the shortest route from an origin node to all other nodes as a succession of node connections, sometimes referred to as polygonal path. The possible connections of each node are limited to the nodes within the selected definition of neighbourhood, known as a forward star (FS). Each connection is assigned a measure of length in a pre-defined unit that works as a weight. If seismic traveltime is chosen as nodal distance (Nakanishi & Yamaguchi 1986), by Fermat's principle the graph method can be adapted to ray tracing and used to generate approximate ray paths corresponding to first-arrival traveltimes. Ray paths of later arrivals corresponding to reflected waves can also be obtained by imposing a constraint defined as the reflecting surface and applying the graph method twice (Moser 1991; Zhang *et al.* 1998; Bai *et al.* 2010); our code follows the same procedure as its 2-D version (Korenaga *et al.* 2000). The nature of the graph method is such that calculating one ray path is just as costly as calculating all of them. Considering this and in order to improve efficiency, in a sequential ray-tracing algorithm, one can apply the reciprocity principle so that the code solves the graph problem using either sources or receivers as origin nodes depending on whichever is smaller in number in each particular experiment. In addition, as the water layer is not part of the velocity model, at each iteration the path through it is found by selecting the seafloor node that minimizes the traveltime to the shot location.

The accuracy and efficiency of the graph method are controlled by the chosen FS that determines the search directions for candidate ray paths, and by the given mesh parametrization that affects the local coarseness of polygonal paths. The FS may limit the allowed connections to nodes in the immediate vicinity of the origin node, but for a better coverage of ray-path directions it must consider connections to nodes other than the nearest neighbours (Moser *et al.* 1992b). To improve the accuracy of the graph ray paths, one must refine the mesh and/or increase the FS. The alternative that we opt for is to use the polygonal path as initial guess for the bending method (Moser *et al.* 1992a; Van Avendonk *et al.* 1998; Korenaga *et al.* 2000). Compared to the pure graph algorithm, for the same target accuracy, the hybrid approach is generally faster than using a finer mesh and a higher-order FS, whereas for the same computational time it is more accurate. It is worth noting that a trade-off in computational time exists between the graph and bending methods: a higher-order FS increases the time spent solving the graph problem but provides a better initial guess that reduces the time spent refining it in bending. The ideal FS depends on the parametrization and complexity of the velocity model required in each specific study case. The hybrid approach is favoured because instead of refining the entire mesh, bending refinement only involves a small number of nodes in the mesh, that is, those in the polygonal paths, which correspond to global traveltime minima within the chosen model discretization.

Cubic B-spline interpolation is also preferred over the finer resampling of polygonal paths (e.g. Pereyra *et al.* 1980; Van Avendonk *et al.* 2001a) for several reasons described in Moser *et al.* (1992a). For instance, interpolation saves computational time

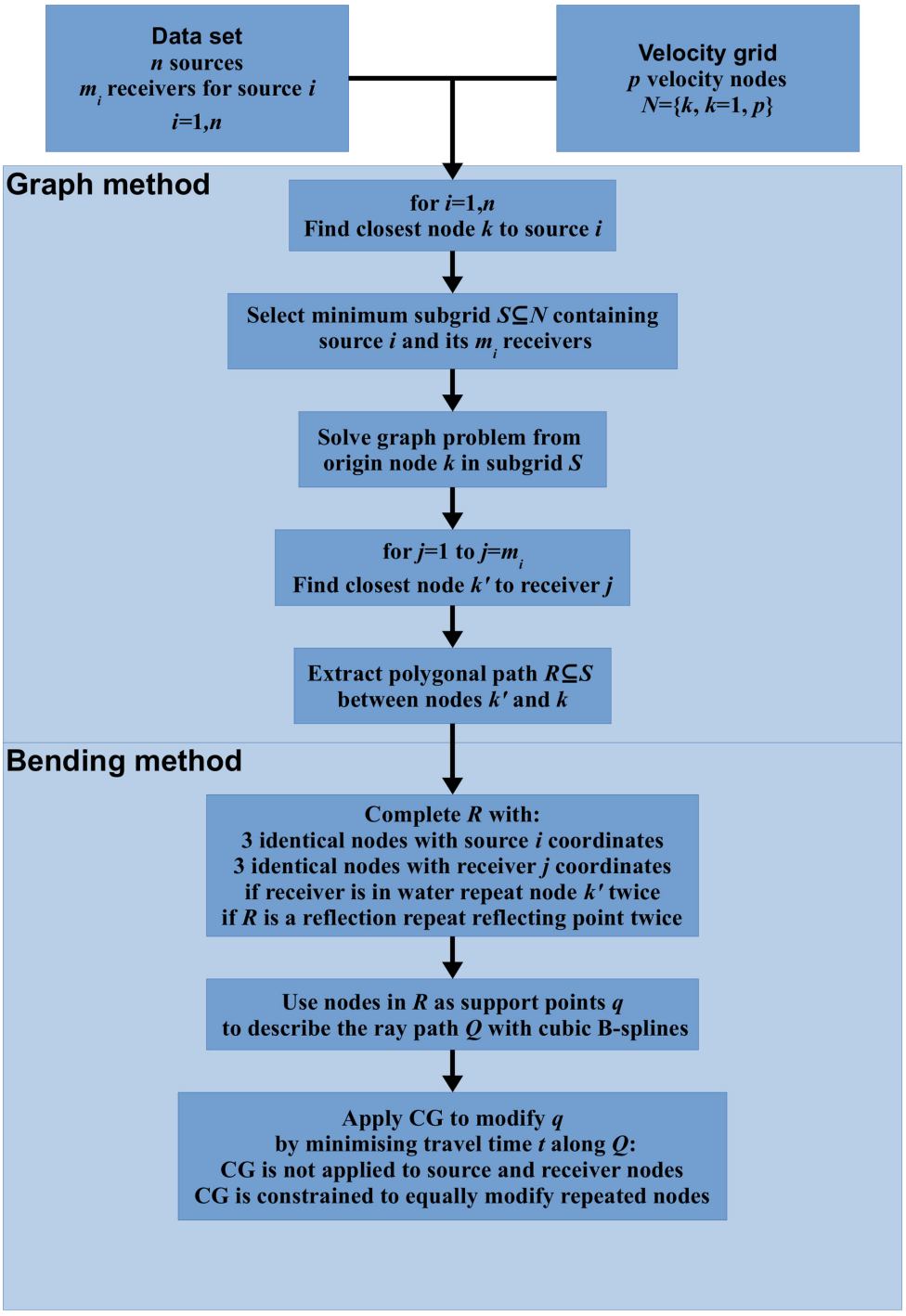

Figure 2. Scheme depicting the ray-tracing process in a velocity grid with p velocity nodes and a data set with n sources and m_i receivers per source. In the first part of the process, for every source i, the graph problem is applied and its solution, in the form of polygonal paths to every j receiver, is used to create initial guesses to be refined by the bending method. In this second part, the nodes in these polygonal paths are taken as support points to represent rays using cubic B-splines. These interpolated rays are refined by adjusting their support points using the conjugate gradient (CG) method to minimize traveltime along the ray.

because it avoids having to increase the sampling of ray paths to make them behave correctly in the presence of strong gradients or especially in low velocity zones. Also, close ray-path points are not completely independent from each other because ray paths are continuous curves. In a path resampling scheme, additional constraints on ray-path points are needed to impose such continuous behaviour. In contrast, B-spline interpolation requires only a few parameters to describe complex continuous curves that can later be sampled as necessary. In other words, the number of points to be perturbed in the minimization process to adequately modify the ray path is smaller than the number of points for a sufficiently accurate integration of traveltime along it. This formulation intrinsically supports the dependency between nearby ray points, and it facilitates the convergence of the conjugate-gradient minimization of traveltime along

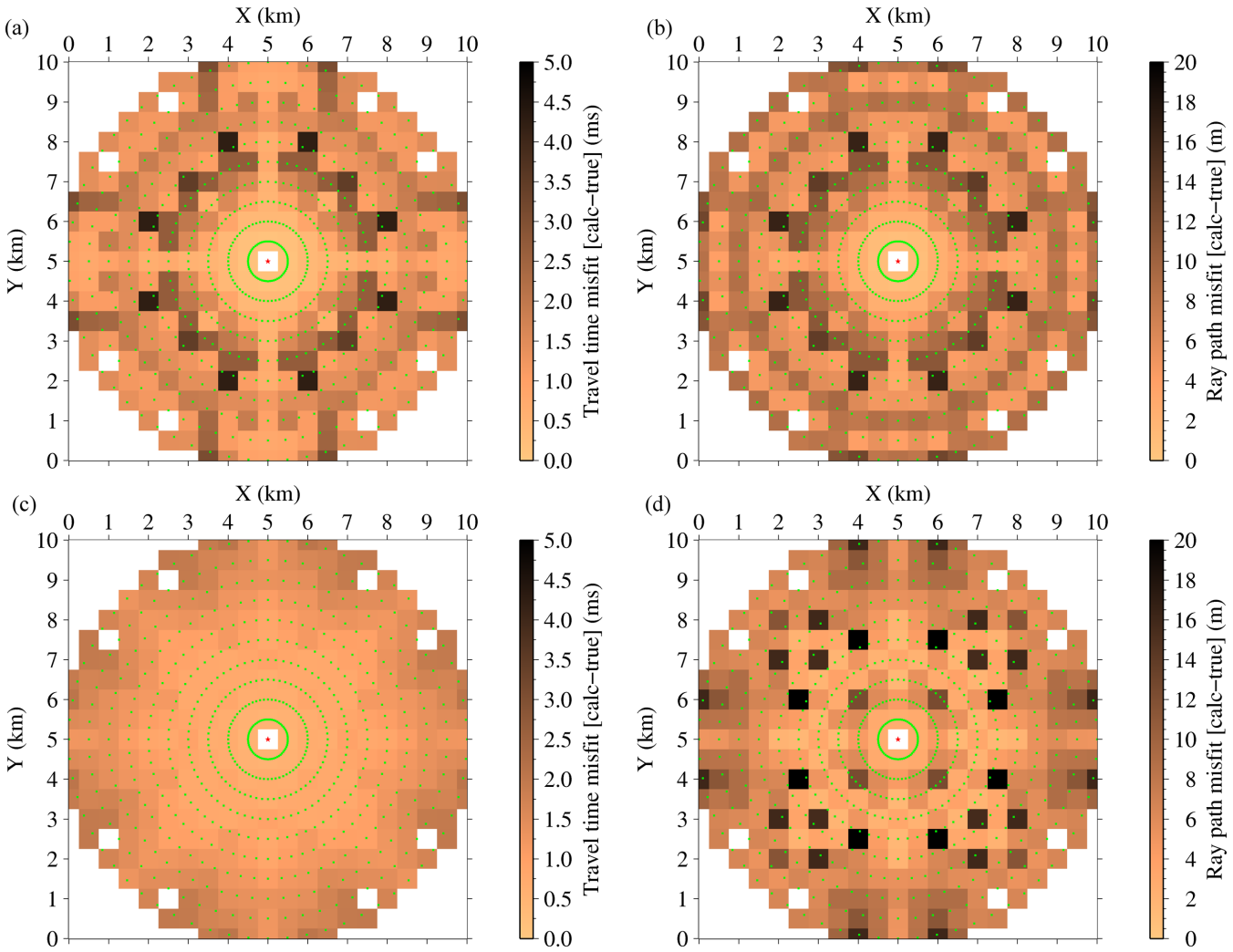

Figure 3. Traveltime and ray-path misfits for (a, b) refracted and (c, d) reflected data. A discrete plot with 2-D bins is preferred over a smoothened one because significantly different misfit values are obtained for close locations. The misfit value in each bin is given by the average of misfits for all receivers in that bin. Bin size is given by the horizontal grid spacing, and the circular distribution of receivers homogeneously scans azimuthal misfit variations. This combination of squared bins and circular acquisition highlights the azimuthal dependence of our ray-tracing accuracy. This depends on the selected parametrization and FS definition which inevitably privileges certain directions over others and is ultimately intrinsic to the use of a Cartesian coordinate system. The geometry of the acquisition configuration includes 640 receivers (black dots) equally distributed in 10 concentric circles and 1 source (red star), all located at the upper face of the cube. Given this acquisition configuration and the velocity and depth models used in this test, the observed symmetry can be anticipated, and it means that the code does not present undesirable directional biases.

the ray path. Thus, the bending refinement method as described in Moser *et al.* (1992a) is applied to the polygonal paths resulting from the graph method to produce smooth ray paths. The mesh nodes forming the polygonal paths are taken as support points to represent the rays using cubic B-splines, the smoothest B-spline curves possible as they respect the continuity of the first and second derivatives. Excessively long segments produced by high-order FS are split to include additional support points. Travel time along interpolated rays is computed and minimized using the conjugate gradients method. When refining a ray path that interacts with the seafloor and/or the reflector, the conjugate gradient search must be constrained. A comparison between polygonal paths and final ray paths after bending is available as Supporting Information (Fig. S1).

A simple test was performed to check that our implementation of the hybrid ray-tracing method works correctly and free of any major coding issue. The velocity model was a 1-D linear velocity function of depth on a 10-km sided cube with no topography, and the reflector model was a flat surface at 5 km depth. A node spacing of 0.5 km was used for both velocity and depth models. For such model, analytical solutions exist for the ray trajectories and the associated traveltimes (Sheriff & Geldart 1995) so that misfits can be computed (Fig. 3). Despite having no 3-D structure this is an adequate test as vertical velocity changes are dominant in the Earth's subsurface. The ray paths calculated with our code for this accuracy test, along with their corresponding polygonal paths, are shown as Supporting Information (Fig. S1). Traveltime misfit is obtained as the difference between analytical and calculated values. Ray-path misfit is a weighted average expressed as

$$\delta p = \sqrt{\sum_{i=1}^{N} p_i \cdot \left[(\delta r_i)^2 + (\delta z_i)^2\right] / \sum_{i=1}^{N} p_i}, \quad (4)$$

where i indicates the ith segment of the calculated ray, δr_i and δz_i are the horizontal and vertical misfits respectively, and p_i is the

length of that segment. In turn, δr_i (and equivalently δz_i) is defined as the difference in length between the horizontal projections of the analytical and calculated trajectories at the *i*th segment. For both refractions and reflections, the average ray-path and traveltime misfits are respectively ∼6 m and ∼1 ms. These misfits are satisfactory considering that the typical node spacing in velocity grids is of several hundreds of meters and the typical data sampling interval is ∼10 ms.

2.3 Inverse problem

Refraction (δt_i^0) and reflection (δt_j^1) traveltime residuals for a given slowness ($u = 1/v$, where v is velocity) and depth model are turned into parameter perturbations (δu and δz, respectively) following the integral equations for refracted (Γ_i^0) and reflected (Γ_j^1) ray paths

$$\delta t_i^0 = \int_{\Gamma_i^0} \delta u \; \mathrm{d}\Gamma \tag{5}$$

$$\delta t_j^1 = \int_{\Gamma_j^1} \delta u \; \mathrm{d}\Gamma + \left[\frac{\partial t}{\partial z}\right]_{(x,y)_j} \delta z \tag{6}$$

$$\left[\frac{\partial t}{\partial z}\right]_{(x,y)_j} = \frac{2 \cdot \cos\theta \cdot \cos\beta}{v\left((x,y)_j\right)}, \tag{7}$$

where $(x,y)_j$ is the reflection point on the interface, θ is the incidence angle with respect to the interface normal vector, β is the local slope of the reflector, and v is the velocity at this point (Bishop *et al.* 1985). Considering our discretization of the problem, eqs (5) and (6) can be collectively expressed in the following linear system:

$$\begin{pmatrix} \boldsymbol{\delta t^0} \\ \boldsymbol{\delta t^1} \end{pmatrix} = \begin{pmatrix} \mathbf{G^{u0}} & \mathbf{0} \\ \mathbf{G^{u1}} & w\mathbf{G^z} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta u} \\ \frac{1}{w}\boldsymbol{\delta z} \end{pmatrix} \tag{8}$$

where $\boldsymbol{\delta t^0}$ and $\boldsymbol{\delta t^1}$ are, respectively, the vectors of refraction and reflection traveltime residuals, $\mathbf{G^{u0}}$, $\mathbf{G^{u1}}$ and $\mathbf{G^z}$ are the Fréchet derivative matrices (or kernels) for velocity and depth, and $\boldsymbol{\delta u}$ and $\boldsymbol{\delta z}$ are the vectors of parameter perturbations. Kernels are built considering eqs (5) and (6); both refractions and reflections contribute to $\mathbf{G^u}$, whereas only reflections contribute to $\mathbf{G^z}$. Each element in the velocity kernel is the length of the portion of a specific ray path partitioned to a relevant velocity parameter consistently with the trilinear interpolation. Similarly, each element in the depth kernel is the portion of eq. (7) for a specific reflected ray path partitioned to a relevant depth parameter consistently with the bilinear interpolation. The depth-kernel weighting parameter w is described in Korenaga *et al.* (2000). Note that w is not an inversion parameter; its purpose is to explore the inherent velocity–depth trade-off in reflection data (Korenaga 2011).

Typically, the number of available data is smaller than the number of model parameters, so the inversion of eq. (8) is conducted with additional regularization constraints. The program works with three smoothing matrices for velocity perturbations ($\mathbf{L^u}$), one for each direction as shown in eq. (9). Similarly, two smoothing matrices for depth perturbations ($\mathbf{L^z}$) are used. These matrices are built with correlation lengths that can vary among nodes. The coefficients λ_u and λ_z determine the relative importance of smoothing with respect to data fit and are selected empirically

$$\begin{pmatrix} \boldsymbol{\delta t^0} \\ \boldsymbol{\delta t^1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{G^{u0}} & \mathbf{0} \\ \mathbf{G^{u1}} & w\mathbf{G^z} \\ \lambda_u \mathbf{L^{uX}} & \mathbf{0} \\ \lambda_u \mathbf{L^{uY}} & \mathbf{0} \\ \lambda_u \mathbf{L^{uZ}} & \mathbf{0} \\ \mathbf{0} & w\lambda_z \mathbf{L^{zX}} \\ \mathbf{0} & w\lambda_z \mathbf{L^{zY}} \\ \alpha_u \mathbf{D^u} & \mathbf{0} \\ \mathbf{0} & w\alpha_z \mathbf{D^z} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta u} \\ \frac{1}{w}\boldsymbol{\delta z} \end{pmatrix}. \tag{9}$$

Because the initial model is generally far from the true one, this system of linear equations must be solved iteratively. However, this could lead to excessively large model perturbations and inversion instabilities. In such cases, damping constraints on velocity and/or depth perturbations can be incorporated to stabilize the inversion (Van Avendonk *et al.* 1998). Whenever the average perturbation for velocity and/or depth is found to be greater than the user-defined limits, the corresponding damping matrix ($\mathbf{D^u}$ and $\mathbf{D^z}$) is added to eq. (9). The code automatically finds the appropriate weights, α_u and α_z, to keep the average perturbation below this limit.

As in the former 2-D version of the code, all entries in eq. (9) are normalized to avoid any biases in the final model deriving from the particular magnitude of either traveltimes or model parameters (Korenaga *et al.* 2000). This equation is inverted using the LSQR algorithm of Paige & Saunders (1982). If a reliable initial model is accessible by other geophysical means, a jumping strategy can be selected instead of the default creeping strategy (Backus & Gilbert 1967; Shaw & Orcutt 1985). The post-inversion smoothing at each iteration remains an available option but is not recommended. Instead we favour the approach of Korenaga & Sager (2012) to handle model roughness.

2.4 Parallelization

In our code, most of the computational time is spent on solving the forward problem. For instance, in the test case presented in the next section this part takes an average of ∼90 per cent of the run time per iteration in the sequential mode. Therefore we prioritized the parallelization of the forward problem over the inverse problem.

The code is parallelized with a combination of multi-processing (MP) and message passing interface (MPI) extensions. Taillandier *et al.* (2009) presented a parallel 3-D refraction traveltime tomography program that used an eikonal-based forward problem solver and the adjoint-state method in combination with the steepest-decent minimization for inversion. Our parallelization strategy is rather intuitive and is conceptually identical to theirs, as our ray-tracing method also deals with each source independently. The graph problem is solved simultaneously for as many sources as the available computational resources allow: each CPU takes care of one source or more, depending on the number of available CPUs. According to the seismic reciprocity principle, sources and receivers can be exchanged in the input data file to minimize computational time depending on the availability and characteristics of computational resources and on the ratio between the average computational times spent in solving one graph and one bending problem for each particular data set; test iterations can be performed to choose between

the two configurations. These source assignments are done in the form of MPI processes. Then bending refinement is performed simultaneously for as many rays as possible: each core traces its share of rays associated with the source(s) assigned to their CPU. These ray assignments are controlled by MP threads. This avoids the need for communication between nodes, which would result in higher computational time. The only significant communication between CPUs happens when all rays have been traced and partial kernels for all sources can be added to obtain the total velocity and depth kernels. A scheme describing the parallelized forward problem is included as Supporting Information (Fig. S2).

3 SYNTHETIC TEST

We conducted a series of synthetic tests to illustrate the resolving potential of TOMO3D under favourable conditions in terms of data acquisition configuration. We proceeded by progressively increasing the complexity of the models to be resolved, and only the final most complicated example is presented here. All of our tests included refraction and reflection data and thus inverted for both velocity and depth parameters. The number of iterations needed to converge to a final result was typically ∼15.

This test was conducted on 36 CPUs featuring an Intel® Xeon® Processor E5-2670 v2 (25M Cache, 2.50 GHz). Each inversion step took ∼4 hr when performed sequentially. Since the experiment includes 36 receivers (treated as sources by the code), the number of CPUs used corresponds to the optimal parallelization. The number of cores per CPU is 10, among which ray bending is distributed. The use of these resources reduced computational time to ∼25 min, which represents a speed-up factor of ∼9.

3.1 Velocity and depth parametrization and true model

The 3-D velocity grid is a prism with squared upper and lower faces of 5 km × 5 km and a depth of 3 km. Velocity nodes are spaced at an interval of 0.125 km in all three spatial dimensions, amounting to a total of 42 025 nodes. Geological relief was added to the velocity models with a geometry created by applying a sinusoidal perturbation to an inclined plane (Figs 4a and b). Concerning the two reflectors, nodal spacing is 0.5 km resulting in 121 depth nodes. This test is focused on the recovery of the shallowest two layers, that is, their velocity distributions and the geometry of their basal reflectors. These synthetic experiments are straightforward to rescale, and the results are independent of the specific dimensions of the model. If the distance unit of the parametrization is multiplied by any factor and velocity values are kept the same, then new traveltimes are simply the old ones scaled by that factor.

Here we use z for vertical position in the velocity mesh, and z' for depth with respect to the sea surface, which are related by b' from eq. (2) as $z = z' - b'$. The true (or target) velocity model is built starting from the linear velocity function of depth $v(z) = 1.6 + 0.5 \cdot z$. The geometry of the shallowest true reflector is given by

$$z'_{T1}(x, y) = z'(x) - A \cdot \sin \times \left(2\pi \cdot \sqrt{\left((x - x_0)^2 + \left(z' - z'_0\right)^2\right)} / \left(c_1 \cdot \sqrt{\left((\Delta x)^2 + (\Delta z')^2\right)}\right)\right) \cdot \sin\left(2\pi \cdot (y - y_0) / (c_2 \cdot \Delta y)\right), \quad (10)$$

where z'_{T1} is depth, $z'(x) = z'_0 + s \cdot x$ with $s = 0.25$ (slope) and $z'_0 = 1$ km (depth at origin), $A = 0.25$ km (amplitude of the perturbation), $x_0 = y_0 = 0$ km (origin coordinates), $c_1 = 1$, $c_2 = 0.5$ (number of sinus cycles), and $\Delta x = 5$ km, $\Delta y = 5$ km, $\Delta z' = 3$ km (model dimensions). The second true reflector (z'_{T2}) follows the same expression but for $z'_0 = 1.75$ km (Figs 4a and b). With these two reflectors, the previous velocity model is modified by applying a velocity shift of −0.25 km/s just beneath each reflector, which can be expressed as

$$v'(x, y, z) = \begin{cases} -v(z), & z' < z'_{T1} \\ -v(z) - 0.25, & z'_{T1} < z' < z'_{T2} \\ -v(z) - 0.5, & z' < z'_{T2} \end{cases}. \quad (11)$$

The construction of the true velocity model is completed after $v'(x, y, z)$ is perturbed with a checkerboard pattern described by

$$v_T(x, y, z) = v'(x, y, z) \cdot \left(1 + 0.01 \cdot A \cdot \sin\left(\frac{2\pi \cdot x}{l_x}\right) \cdot \sin\left(\frac{2\pi \cdot y}{l_y}\right) \cdot \sin\left(\frac{2\pi \cdot z}{l_z}\right)\right), \quad (12)$$

where $l_x = l_y = l_z = 2$ km, and $A = 10$ per cent (Fig. 4).

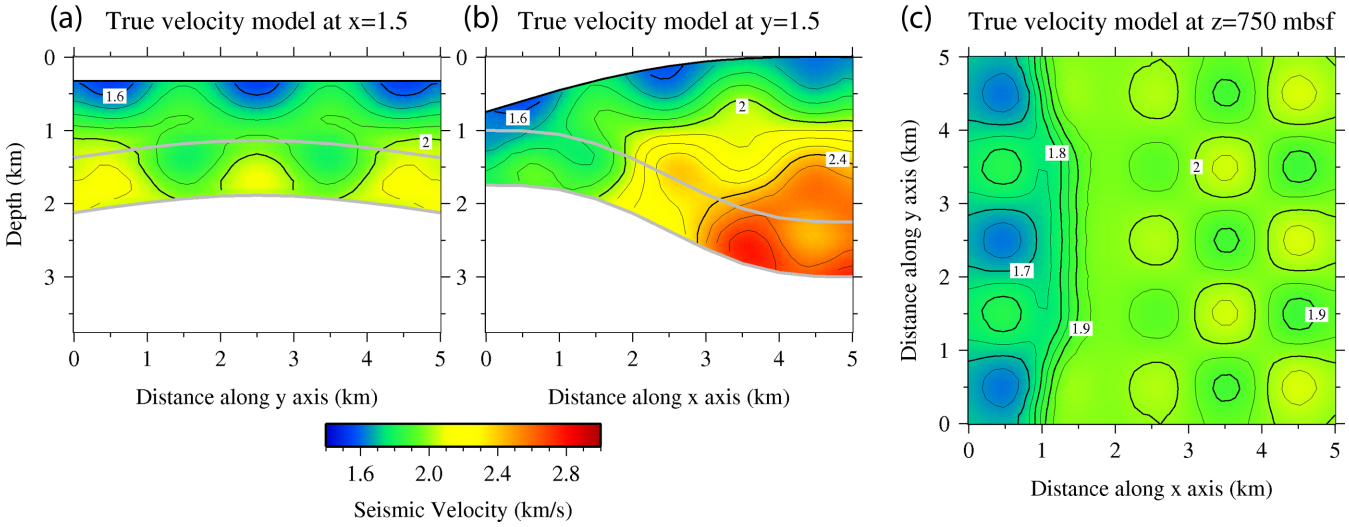

Figure 4. 2-D slices of the true velocity model at (a) $x = 1.5$ km, (b) $y = 1.5$ km and (c) $z = 750$ mbsf with the corresponding 1-D cuts of both true reflectors (grey lines).

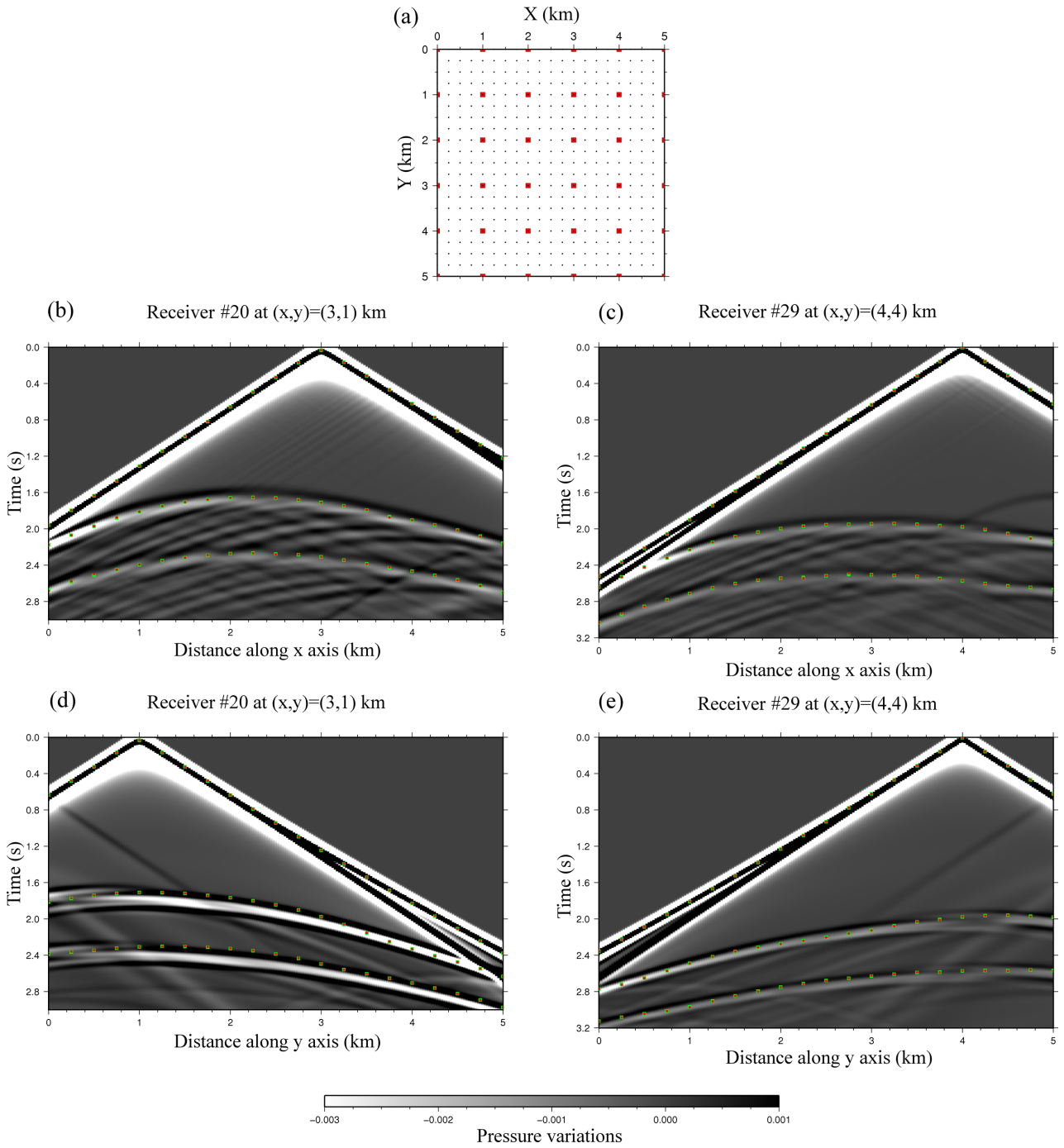

Figure 5. (a) Acquisition geometry with 36 receivers and 441 sources. Black dots mark shot locations and red squares indicate receivers. (b–e) Synthetic record sections for four sample profiles. These records were generated on slices of our true 3-D velocity model. Applying the reciprocity principle, the 2-D wavefield was computed for a source at the location of the chosen receiver. Traces were acquired every 25 m along the profile to achieve the lateral coherency needed to observe the various seismic phases. True (green squares) and inverted (red dots) traveltime picks are plotted to show their correspondence with one another and with these synthetic seismic phases.

3.2 Acquisition configuration and data set

The acquisition configuration consists of 36 receivers located at the seafloor and distributed in a squared grid of 6 × 6 receivers with 1 km spacing (Fig. 5a). The 441 sources are spaced 0.25 km and located 10 m below sea level. This configuration produces 15 876 refraction picks, and this same number of reflection picks for each of the two reflectors. Typical active-source real data experiments provide less data than the number of modelling parameters. This data set in combination with the total number of model parameters reproduces a rather optimal experimental situation in which the number of traveltime picks is almost as large as the number of parameters to be estimated.

The synthetic traveltime picks for this acquisition geometry were obtained with the same forward problem solver in TOMO3D. Random noise of ±5 ms was added to these picks to create the final data set in order to make our synthetic application even more realistic.

This traveltime perturbation range of 10 ms was selected to be of the order of magnitude of the RMS of traveltime residuals for this same synthetic application with a noise-free data set (∼1.4 ms for refractions and ∼3.4 ms for reflections); for a satisfactory inversion RMS values must be close to the average traveltime error. We refer to these picks as true traveltime picks, and we plotted some of them on top of their corresponding synthetic record sections generated with the acoustic version of the 2-D finite-differences elastic wave solver by Dagnino *et al.* (2014; Figs 5b–e). The fit between the picks and the seismic phases is satisfactory and illustrates the robustness of our 3-D ray-tracing and traveltime calculation method. Possible mismatches most likely originate from the comparison between 3-D ray tracing and 2-D wave propagation. This acquisition geometry covers the two layers of interest but the second one is sampled mostly by reflections, and ray coverage is nonexistent beneath the deeper reflector. Fig. 6 shows the ray trajectories corresponding to the true traveltime picks in Fig. 5.

3.3 Layer-stripping strategy and test results

The purpose of the layer-stripping strategy (Sallarès *et al.* 2013) is twofold: (1) to invert for more than one reflector by posing successive single-reflector inversion problems, and (2) to recover sharp geological discontinuities that might otherwise appear as smooth velocity gradients. Accomplishing these purposes generally yields models that facilitate geological interpretation. This strategy proceeds by extending the model downwards layer by layer, instead of concurrently inverting all interfaces, which would pose a demanding inverse problem (e.g. codes that invert several reflectors simultaneously would require multiple depth-kernel scaling parameters). The first inversion is devised to recover the velocity within the first, shallowest layer and the depth of the corresponding reflector. Subsequently, the area of the inverted velocity model, limited by the topographic surface at the top and the geometry of the inverted reflector at the bottom, is inserted in a new starting model that extends deeper to include the next reflector. Because we only invert for one reflector at a time, at each step the reflection picks in the data set are replaced by those associated with the new reflector. If the velocity parameters of the previous inversion result are significantly altered but there is reason to believe that the result is nonetheless correct (e.g. in the form of geophysical information that is external to the inversion process), damping can be applied to the previously inverted parameters with respect to those in the new deeper area of the model. In this manner, in order to fit the new picks, the inversion tends to modify the deeper structure rather than the formerly inverted shallower structure.

In the first step of the layer-stripping strategy, the starting velocity model follows the function $v(z) = 1.6 + 0.3 \cdot z$, whereas the starting depth model is represented by $z'(x) = 1 + 0.2 \cdot x$. We set the uppermost velocity to the correct value considering that in a real case we could obtain a reasonably good estimate of that value for instance by forward modelling, drilling or from previous geological knowledge. Likewise, the shallowest portion of the reflecting boundary can be well determined from bathymetry data or seismic reflection experiments, so that we correctly set this value as well (i.e. the seafloor at the trench in real experiments). As shown in Fig. 7, the velocity and depth recoveries are excellent, especially in the central area of the model where the data set offers the best ray coverage. The velocity fit is shown via 2-D vertical and horizontal slices of the 3-D volume. The percentage differences between initial, final and true velocity models show the correctness of the inverted velocity distribution. The poor recovery area observed in Fig. 7(i) that stands out against the rest of the horizontal slice corresponds to the portion of plane that is beneath the first reflector, and will be well retrieved after the second step of layer stripping. Outside this particular area, the highest misfit (Figs 7g–i) values are between 4 per cent and 6 per cent and are limited to small localized zones at the bottom and edges of the model. For the rest of the volume velocity misfit is close to 0 per cent with small areas of ∼2 per cent. The fit between true and final depth model is displayed in Figs 8(a)–(c) and just as for the velocity model the agreement is satisfactory, in particular for the central and shallow areas of the reflector, with misfit values between 0 per cent and 1 per cent (Fig. 8c). The highest misfit values are around 2 per cent and are again found in small local areas at the edges of the deepest part of the reflector. Histograms in Figs 9(a) and (b) show the distribution of refraction and reflection traveltime residuals before and after inversion proving that the iterative inversion produces velocity and depth models that explain the data remarkably well. The combined RMS for this first step is 2.1 ms, with 1.6 ms and 2.5 ms for refraction and reflection picks, respectively.

In the second step, the reflection picks associated with the first reflector are replaced by those associated with the second reflector. Regarding the starting velocity model, the parameters above the first reflector are set to the result of the first inversion step, while the parameters beneath it are set to follow $v(z) = 1.25 + 0.5 \cdot z$ so as to mark a velocity contrast matching this reflector. The starting reflector is represented by $z'(x) = 1.75 + 0.22 \cdot x$. As in the first step certain assumptions have been made. Here we considered that additional available knowledge allows us to roughly estimate the velocity value immediately below the boundary. As mentioned before, in a real experiment this information might be given by some supplementary geophysical exploration or data processing method such as an amplitude analysis of the reflection event in the seismic traces. Similarly, the shallowest position of this second starting reflector could be derived from other techniques or well-established knowledge, in this case on the average oceanic crust. Moreover, based on the observation of the geometry of the first boundary, one could decide to slightly increase the slope of the plane defining the starting depth model for the second boundary.

Fig. 10 displays the result after the second (and in this case final) step of layer stripping. It was not necessary to damp the velocity parameters inverted in the first step in relation to those in the new area of the model because our data set and inversion parameters proved sufficiently robust to keep the shallower structure unchanged, while iteratively improving the deeper layer to fit the newly included reflection data. The recovery of both velocity and depth parameters is not as good as for the previous layer. This could be anticipated simply because recovery generally deteriorates with depth, but also because the new layer is almost only mapped by reflections (Figs 5b–e), which are intrinsically subject to velocity–depth ambiguity. Satisfactory misfit values between 0 per cent and 2 per cent are found for the upper half and central areas of the layer above 2 km depth (Figs 10g–i). At the deeper part of the layer, velocity is not retrieved correctly although misfit is mostly between 4 per cent and 6 per cent with small localized maxima of up to 8–10 per cent. The accumulation of higher velocity errors just above the reflector is characteristic of velocity–depth trade-off. Still, this effect is mitigated by the fact that we are using wide-angle, instead of near-vertical reflections, so that there is an acceptable azimuthal coverage even in this second layer, especially in its upper part. The velocity fit for the previous layer is also improved, especially at the central part and around its bottom reflector where misfit is now

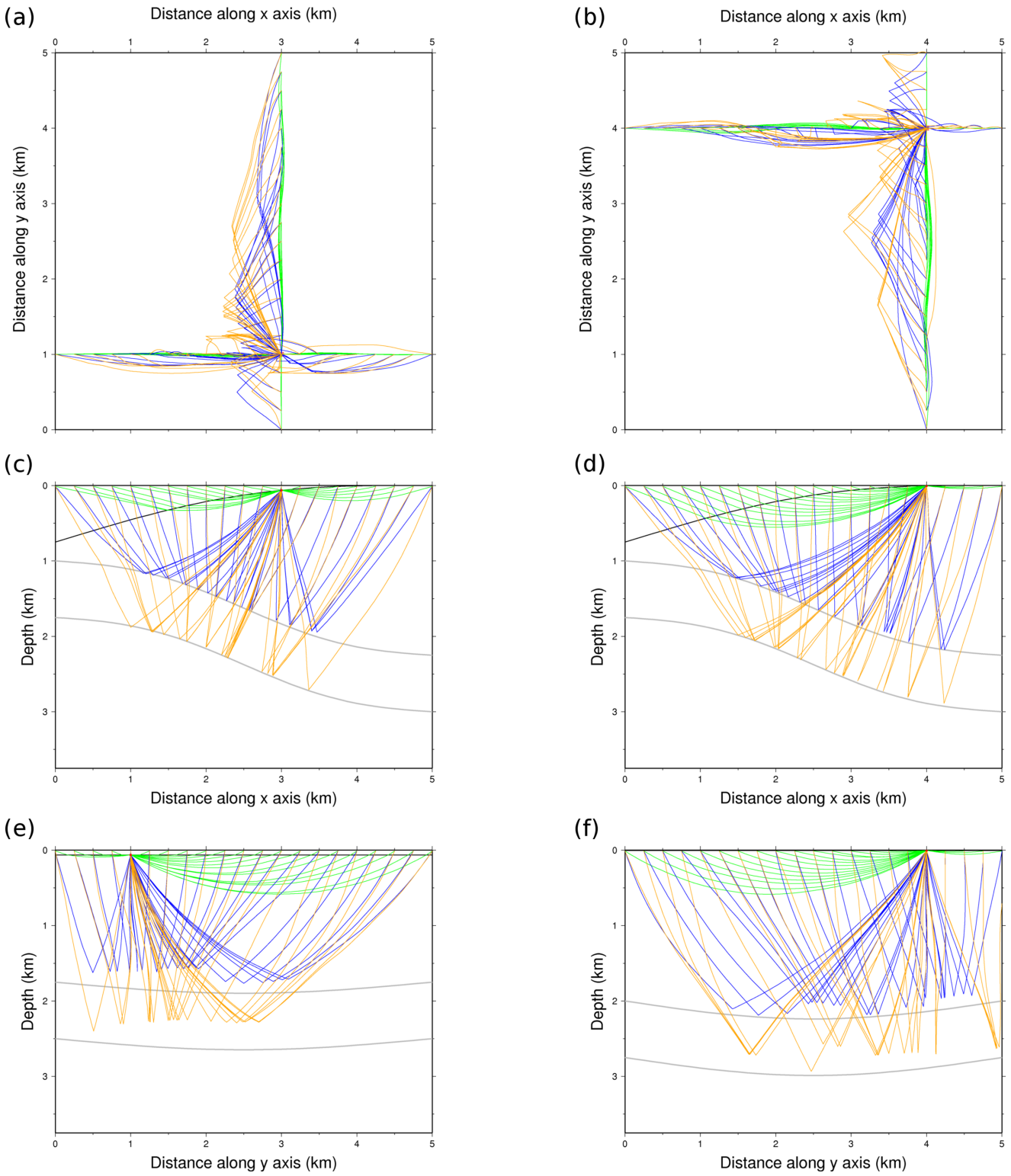

Figure 6. Rays corresponding to the true traveltime picks (green squares in Fig. 5). Refracted rays are plotted in green, reflected rays at the first reflector in blue, and reflected rays at the second reflector in orange. Left and right columns correspond to receivers #20 and #29, respectively. (a, b) Plan view; (c–f) cross-sections along the horizontal axes. Grey lines indicate the locations of the two reflectors. Note that many reflections occur off the selected vertical planes, particularly along the *y* axis.

between 0 per cent and 1 per cent, indicating that the sharp contrast at this first interface is successfully recovered. Moreover, the values at the localized areas of highest misfit after the first step (Figs 7g and h) decreases to between 0 per cent and 2 per cent (Figs 10g and h). More important, the area beneath the first reflector in Fig. 7(i) is well recovered after this second step (Fig. 10i).

Consistently with the overall velocity recovery, the checkerboard pattern is well retrieved for the first layer and partially recovered

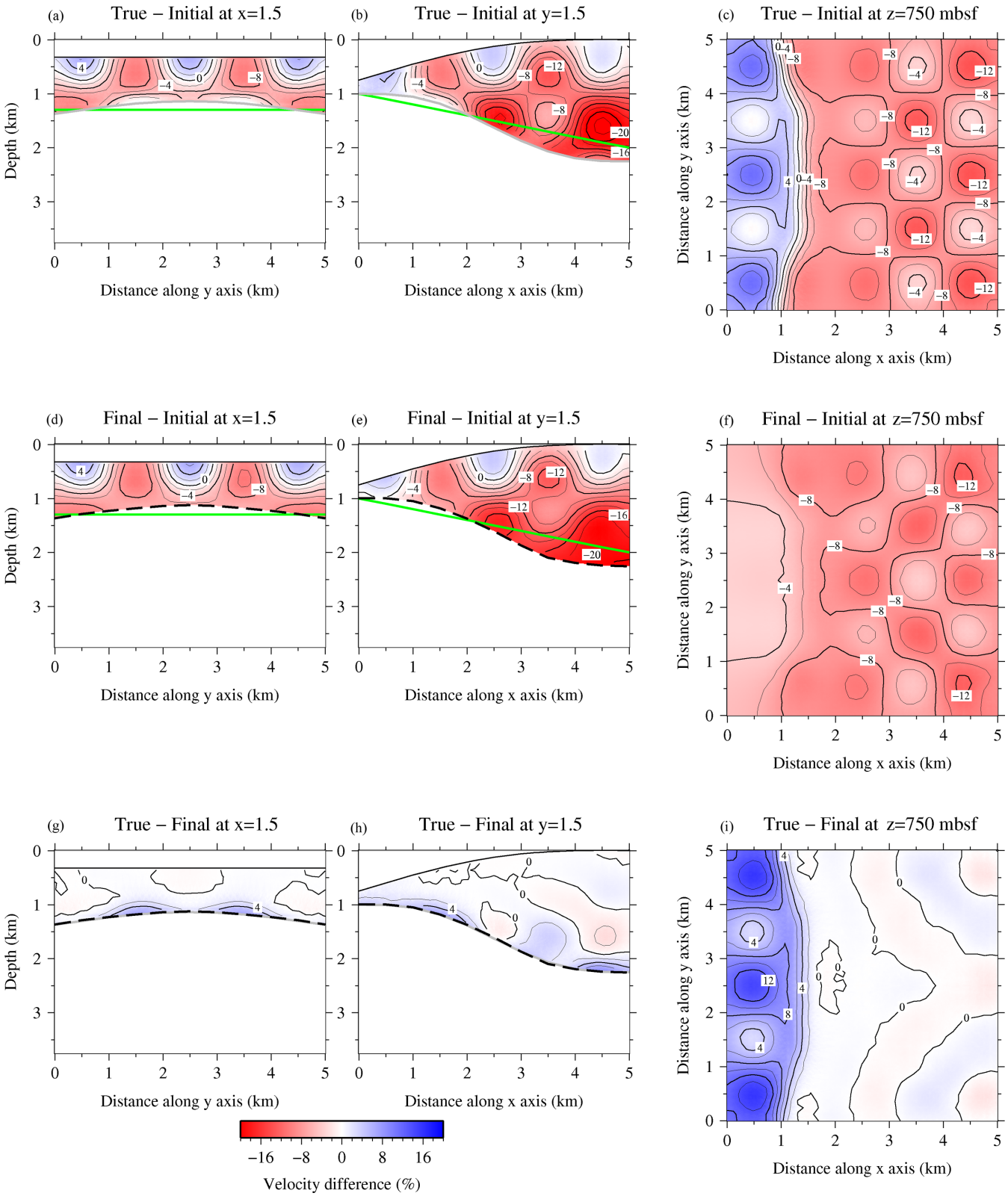

Figure 7. Velocity recovery after layer stripping step 1 illustrated as the percentage differences between 2-D slices of the initial, final and true models. On the vertical slices the corresponding 1-D cuts of the initial (green lines), final (black dashed lines) and true (grey lines) reflectors are also plotted.

for the second one (Fig. 11). Comparison between the recovered checkerboard pattern and the original one illustrates the resolution of the experiment throughout the velocity model. The geometry of the reflector is adequately retrieved, particularly at the shallow central region of the interface where misfit is between 0 per cent and 1 per cent (Fig. 8d–f). The highest misfit values are around 2 per cent and concentrate in small pseudo-circular areas along model edges and at the deeper region of the reflector (Fig. 8f).

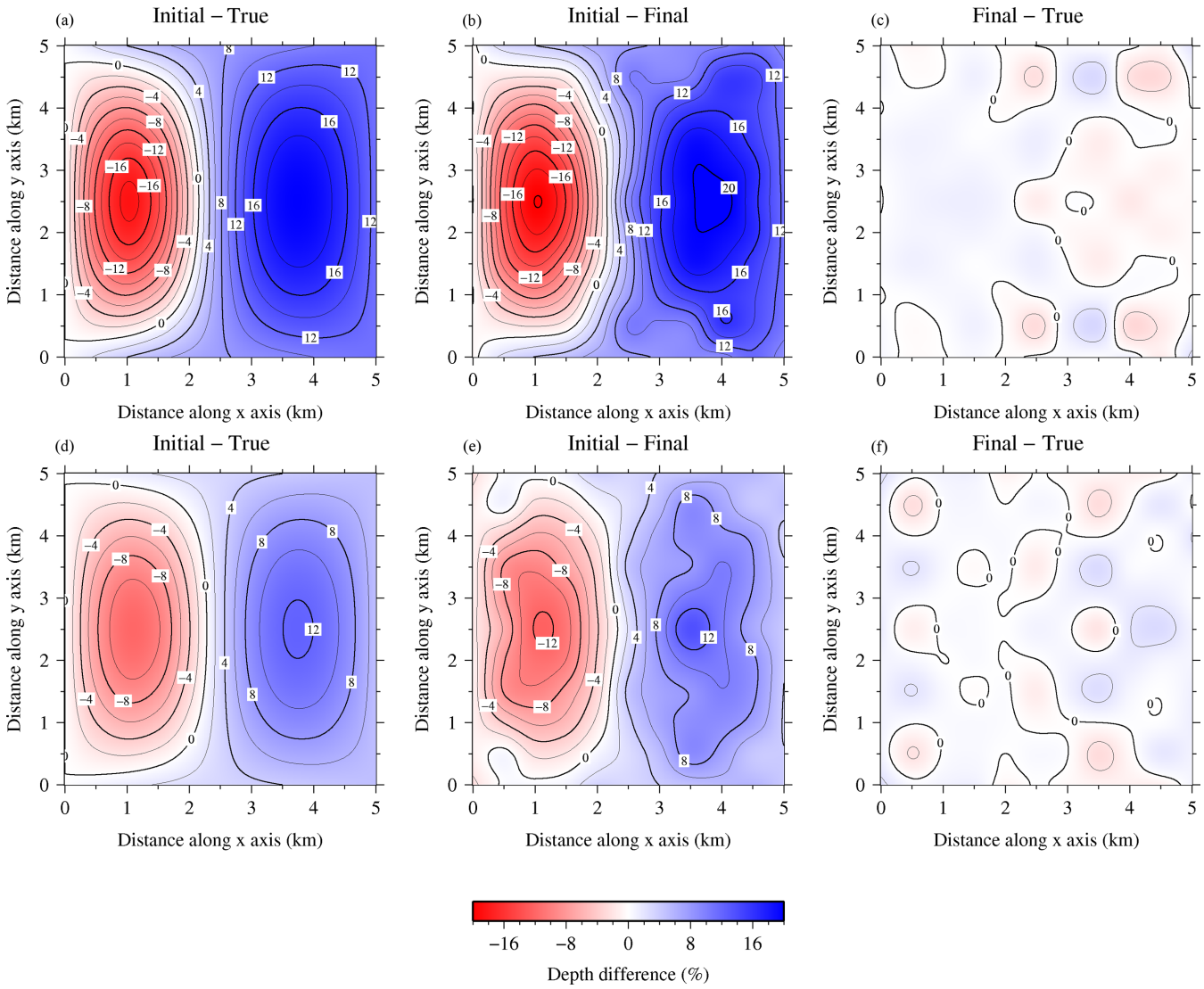

Figure 8. Recovery of interface geometry after layer stripping (a–c) step 1 and (d–f) step 2 illustrated as the percentage differences in depth between initial, final and true reflectors.

Also as in the first step, histograms in Figs 9(c) and (d) show the distribution of refraction and reflection residuals before and after inversion. In this second step we observed how, as expected for a robust inversion solution, refraction residuals remain virtually invariable with respect to the first step. Reflection residuals for the second reflector, however satisfactory, are not as good as for the first one, in accordance with the quality of their respective depth recoveries. The combined RMS for this second step is 3.0 ms, and 1.6 ms and 3.9 ms for refraction and reflection picks, respectively, and the overall combined RMS is 2.7 ms. Examples of the good agreement between true and calculated picks can be seen in Figs 5(b)–(e). The inversion parameters used in the two inversion steps are presented in Table 1.

4 DISCUSSION

The usefulness of TOMO3D has been tested in a complex synthetic case simulating a subduction zone scenario in which it successfully resolved the velocity field of the upper and lower layers as well as the geometry of the boundary and bottom interfaces. First of all, the precision of the forward problem solver was investigated by comparing the synthetic ray paths and traveltimes to the analytical solutions (Fig. 3). The quality of the synthetic data set as well as of the inversion results was illustrated by plotting the picks on top of the synthetic record sections generated with a 2-D acoustic wave propagator showing the satisfactory agreement between inverted and true picks, and with their corresponding seismic phases (Fig. 5).

The successful recovery of the 3-D velocity structure of both layers as well as of the both 2-D depth interfaces is exemplified by the corresponding misfits (Figs 7, 8, 10 and 11) and the traveltime residuals (Fig. 9). Recovery for the lower layer is obviously poorer than for the upper one. However, an acquisition configuration with longer offsets would record refracted arrivals corresponding to rays travelling deeper into the subsurface and improve the recovery of the lower layer. Here, this lower layer is sampled almost only by reflections, which indicates that its recovery would not be possible by refraction traveltime tomography. The symmetry observed in depth misfits is expected, given the symmetry defined by the acquisition configuration, and the true velocity and depth models. Moreover, this suggests that the final velocity model and its misfits also follow an analogous symmetry that can be checked by plotting 2-D slices at

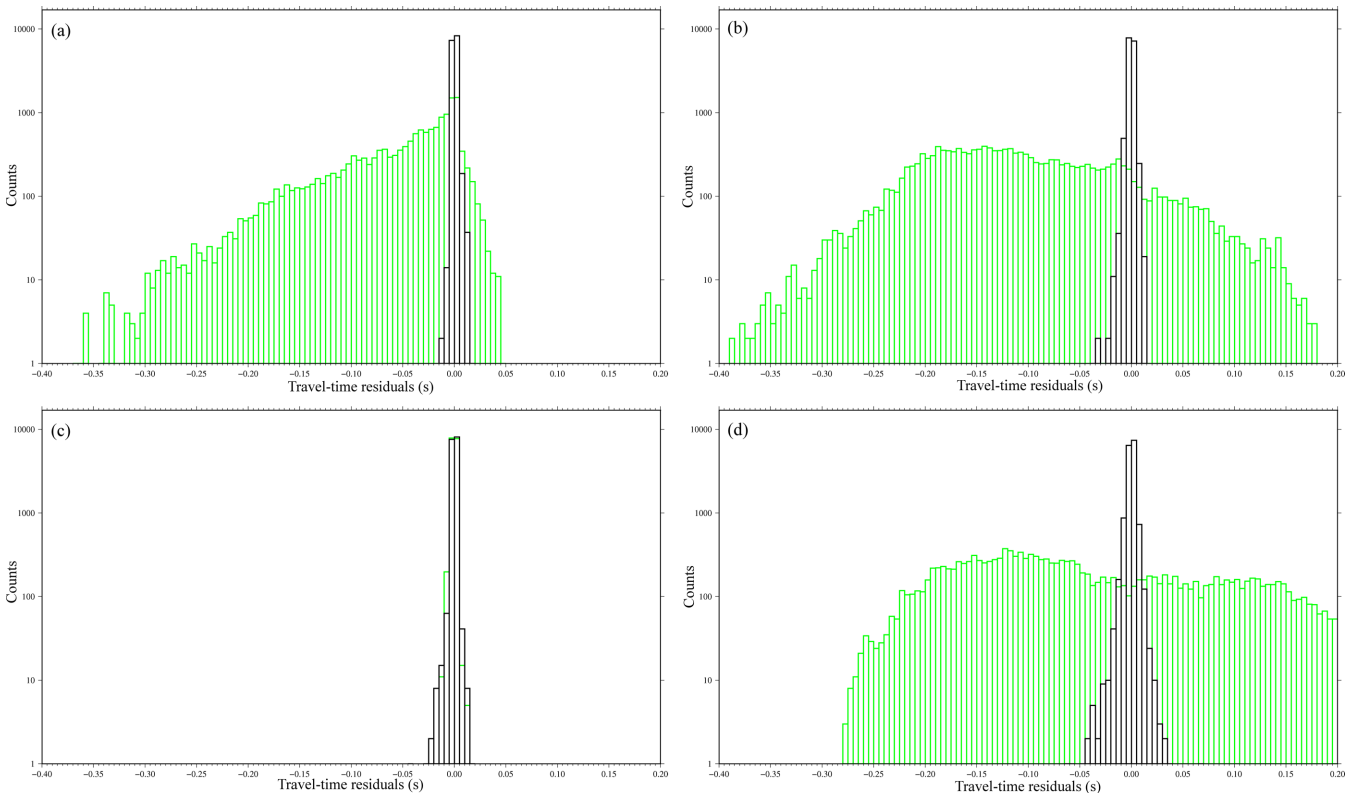

Figure 9. Histograms of initial (green) and final (black) traveltime residuals for (a) refractions and (b) reflections after layer stripping step 1, and (c) refractions and (d) reflections after step 2.

the appropriate x and y positions. Nonetheless, this synthetic test is not as nonlinear as a real data inversion, and in that sense it should be considered as an optimal example of the code's performance. Indeed, even if the velocity and depth functions used to build the initial and true models are different, they are closer to one another than any initial model can be to the actual 3-D velocity structure of the subsurface. Regarding initial models, certain reasonable assumptions were considered here, which would also be valid in a real data study. Ideally though, in a real data case one would like to use an adaptive importance sampling scheme as described in Korenaga & Sager (2012), which statistically evaluates pre-defined ranges of geologically reasonable model parameters. Parallelization and optimization of the code are critical to make these computationally expensive analyses feasible in 3-D.

Additionally, we show our proposed strategy to model impedance contrasts that may appear at the boundaries between geological structures. Traveltime tomography velocity models are characterized by smooth velocity gradients as opposed to layered models obtained from multichannel seismic data processing or from forward modelling of wide-angle seismic data. Layer stripping is devised to introduce and keep such sharp velocity contrasts throughout the entire inversion process. The synthetic record sections in Fig. 5 show how this strategy is mostly an objective modelling process when the different reflected seismic phases can be identified in the record sections. From a technical point of view, inverting for more than one reflector simultaneously is only a matter of making minor modifications to the code. The choice of layer stripping is thus conceptual and seeks to simplify the inverse problem that is posed, which typically produces models that facilitate interpretation. Nevertheless, trade-off effects can still be present in the joint inversion of refraction and reflection data, and its quantification is essential in real data experiments as demonstrated in Korenaga (2011). As with TOMO2D, the degree of velocity–depth trade-off in the data set and its influence on the final velocity and depth models can be evaluated by exploring a range of w values, and to our knowledge, this is the only published procedure to do so. If inversion results show a strong dependence on w, then the data set is severely affected by velocity–depth ambiguity. This sort of assessment becomes crucial in real data inversions where no true model is available for comparison.

5 CONCLUSIONS

TOMO3D is a fully operative, state-of-the-art, parallel code for the joint inversion of refraction and reflection seismic data that produces 3-D velocity models of the subsurface and 2-D depth models representing the geometry of geological reflecting interfaces. Founded on TOMO2D (Korenaga *et al.* 2000), TOMO3D incorporates the same forward and inversion methods extended to work in 3-D. The code can be downloaded from http://barcelona-csi.cmima.csic.es/software/tomo3D, along with user and installation manuals, and it can be used only for academic purposes. Users should cite this paper when presenting their results. The following list summarizes the key points of the work presented here:

(1) The hybrid ray-tracing strategy combining the graph and bending methods provides adequate traveltime and ray-path precision as proven by comparison with analytical solutions and synthetic seismograms.

(2) Parallelization of the forward problem is crucial for the practicality of the program as around 90 per cent of the run time is spent in this part of the code. The achieved speed up in the particular case described here is a factor of ∼9.

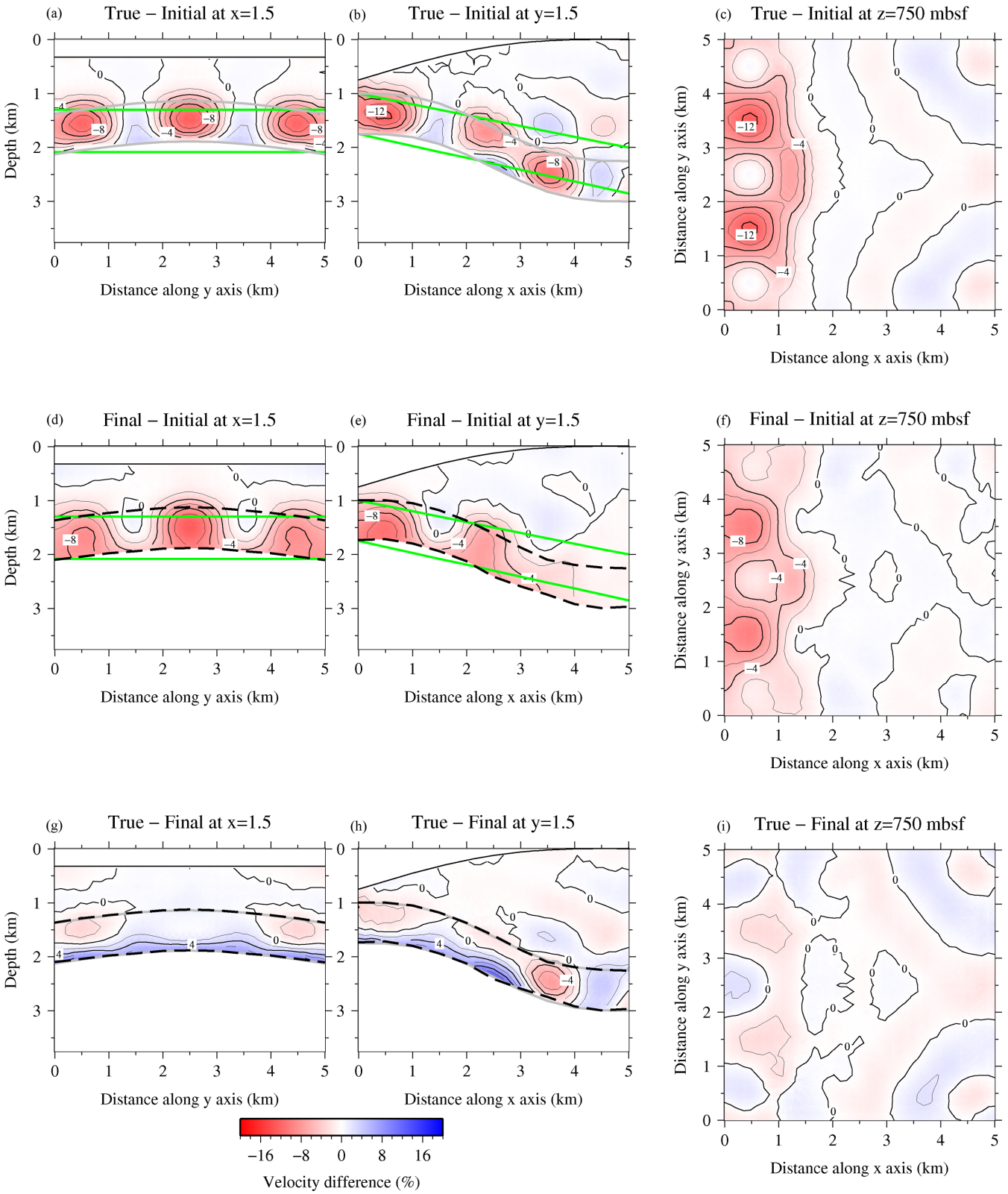

Figure 10. Velocity recovery after layer stripping step 2 illustrated as the percentage differences between 2-D slices of the initial, final and true models. On the vertical slices the corresponding 1-D cuts of the initial (green lines), final (black dashed lines) and true (grey lines) reflectors are also plotted.

(3) The application of layer-stripping strategy in combination with our joint refraction and reflection inversion scheme allows for the successive inversion of several reflecting boundaries adding sharp contrasts to the otherwise smoother velocity distributions produced by classic refraction tomography.

(4) The code can still be improved, for instance by giving the layer-stripping strategy a certain degree of automation. The

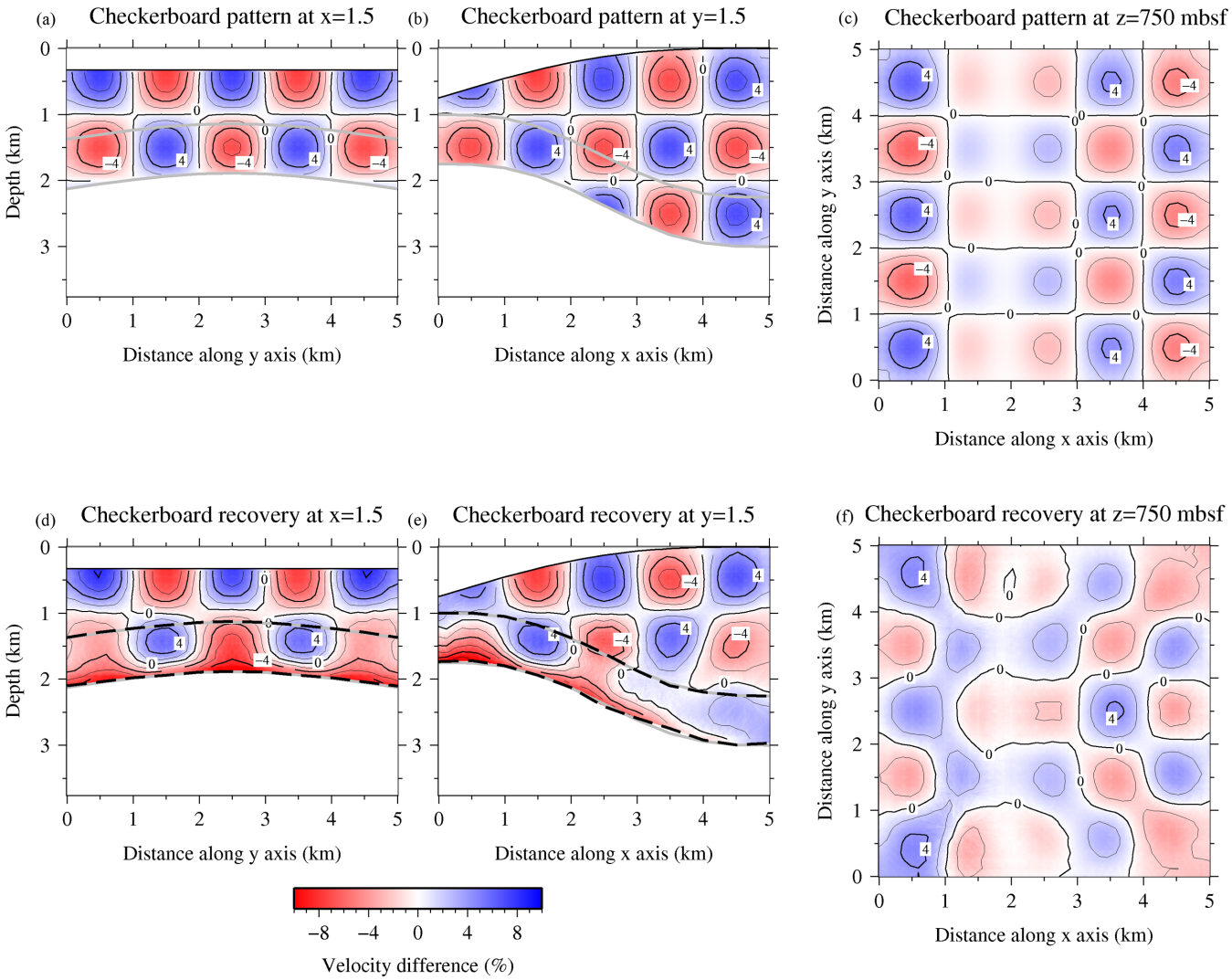

Figure 11. 2-D slices of the checkerboard pattern (a) at $x = 1.5$ km, (b) $y = 1.5$ km and (c) $z = 750$ mbsf with the corresponding 1-D cuts of both true reflectors (grey lines). (d–f) Checkerboard pattern recovery after layer stripping step 2 for the same 2-D slices including the corresponding 1-D cuts of both final reflectors (black dashed lines).

Table 1. Relevant inversion parameters used in the synthetic test (LS stands for layer stripping). Correlation lengths for the velocity model were set at the top and bottom nodes of the velocity grid and interpolated for the rest of the nodes in between. The increase with depth follows from the typical decrease in resolving power. The second step needed longer correlation lengths for the depth model because of the poorer ray coverage, particularly due to the lack of refractions travelling through the deeper parts of the model.

Inversion parameters	LS step 1	LS step 2
FS order (x, y, z) (number of nodes)	(4,4,4)	(4,4,4)
λ_u	15	11
λ_z	1	1
Average velocity perturbation limit (%)	10	10
Average depth perturbation limit (%)	10	10
Top velocity correlation lengths (x, y, z) (km)	(0.5,0.5,0.5)	(0.5,0.5,0.5)
Bottom velocity correlation lengths (x, y, z) (km)	(1,1,1)	(1,1,1)
Depth correlation lengths (km)	(0.5,0.5)	(1,1)

inversion solver could be parallelized incorporating a freely available parallel version of the LSQR algorithm (Lee *et al.* 2013), and the parallelization of the ray-tracing method could be extended to a second level of parallelism in the solution of graph problems.

ACKNOWLEDGEMENTS

AM is funded by Repsol through the CO-DOS collaboration project with CSIC, and the work in this paper was carried at the Grup de Recerca de la Generalitat de Catalunya 2009SGR146: Barcelona

Center for Subsurface Imaging (B-CSI) of the Institut de Ciències del Mar (CSIC). Part of this work was also financed by the MICINN project POSEIDON (CMT2010-21569). We thank all our fellows at the B-CSI for their contribution to this work, as well as Editor Prof Jean Virieux, Reviewer Prof Nicholas Rawlinson, and an anonymous reviewer for their constructive comments.

REFERENCES

Aki, K. & Lee, W.H.K., 1976. Determination of three-dimensional velocity anomalies under a seismic array using first P arrival times from local earthquakes, 1. A homogeneous initial model, *J. geophys. Res.,* **81,** 4381–4399.

Backus, G. & Gilbert, F., 1967. Numerical applications of a formalism for geophysical inverse problems, *Geophys. J. R. astr. Soc.,* **13,** 247–276.

Bai, C.Y., Greenhalgh, S. & Zhou, B., 2007. 3D ray tracing using a modified shortest-path method, *Geophysics,* **72,** T27–T36.

Bai, C.-Y., Huang, G.-J. & Zhao, R., 2010. 2-D/3-D irregular shortest-path ray tracing for multiple arrivals and its applications, *Geophys. J. Int.,* **183,** 1596–1612.

Bishop, T.N. *et al.*, 1985. Tomographic determination of velocity and depth in laterally varying media, *Geophysics,* **50,** 903–923.

Bois, P., La Porte, M., Lavergne, M. & Thomas, G., 1972. Well-to-well seismic measurements, *Geophysics,* **37,** 471–480.

Čeverný, V., 2001. *Seismic Ray Theory,* 1st edn, Cambridge Univ. Press.

Cheng, N. & House, L., 1996. Minimum traveltime calculation in 3-D graph theory, *Geophysics,* **61,** 1895–1898.

Dagnino, D., Sallarès, V. & Ranero, C.R., 2014. Scale- and parameter-adaptive model-based gradient pre-conditioner for elastic full-waveform inversion, *Geophys. J. Int.,* **198,** 1130–1142.

Dijkstra, E.W., 1959. A note on two problems in connection with graphs, *Numer. Math.,* **1,** 269–271.

Dunn, R.A., Lekić, V., Detrick, R.S. & Toomey, D.R., 2005. Three-dimensional seismic structure of the Mid-Atlantic Ridge (35°N): evidence for focused melt supply and lower crustal dike injection, *J. geophys. Res.,* **110,** B09101, doi:10.1029/2004JB003473.

Farra, V. & Madariaga, R., 1988. Non-linear reflection tomography, *Geophys. J.,* **95,** 135–147.

Gallo, G. & Pallottino, S., 1986. Shortest path methods: a unifying approach, *Math. Program. Stud.,* **26,** 38–64.

Hobro, J.W.D., Singh, S.C. & Minshull, T.A., 2003. Three-dimensional tomographic inversion of combined reflection and refraction seismic traveltime data, *Geophys. J. Int.,* **152,** 79–93.

Hole, J.A., 1992. Nonlinear high-resolution three-dimensional travel-time tomography, *J. geophys. Res.,* **97,** 6553–6562.

Iyer, H.M. & Hirahara, K. (eds), 1993. *Seismic Tomography: Theory and Practice,* Springer Science & Business Media.

Kim, S., 2002. 3-D Eikonal solvers: first-arrival traveltimes, *Geophysics,* **67,** 1225–1231.

Kissling, E., 1988. Geotomography with local earthquake data, *Rev. Geophys.,* **26,** 659–698.

Korenaga, J., 2011. Velocity–depth ambiguity and the seismic structure of large igneous provinces: a case study from the Ontong Java Plateau, *Geophys. J. Int.,* **185,** 1022–1036.

Korenaga, J., Holbrook, W.S., Kent, G.M., Kelemen, P.B., Detrick, R.S., Larsen, H.-C., Hopper, J.R. & Dahl-Jensen, T., 2000. Crustal structure of the southeast Greenland margin from joint refraction and reflection seismic tomography, *J. geophys. Res.,* **105,** 21 591–21 614.

Korenaga, J. & Sager, W.W., 2012. Seismic tomography of Shatsky Rise by adaptive importance sampling, *J. geophys. Res.,* **117,** B08102, doi:10.1029/2012JB009248.

Koulakov, I., 2009. LOTOS code for local earthquake tomographic inversion: benchmarks for testing tomographic algorithms, *Bull. seismol. Soc. Am.,* **99,** 194–214.

Lee, E.-J., Huang, H., Dennis, J.M., Chen, P. & Wang, L., 2013. An optimized parallel LSQR algorithm for seismic tomography, *Comput. Geosci.,* **61,** 184–197.

Lutter, W.J. & Nowack, R.L., 1990. Inversion for crustal structure using reflections from PASSCAL Ouachita experiment, *J. geophys. Res.,* **95,** 4633–4646.

McCaughey, M. & Singh, S.C., 1997. Simultaneous velocity and interface tomography of normal-incidence and wide-aperture seismic traveltime data, *Geophys. J. Int.,* **131,** 87–99.

Meléndez, A., 2014. *Development of a New Parallel Code for 3-D Joint Refraction and Reflection Travel-Time Tomography of Wide-Angle Seismic Data – Synthetic and Real Data Applications to the Study of Subduction Zones,* Institut de Ciències del Mar (CSIC), Universitat de Barcelona.

Moser, T.J., 1991. Shortest path calculation of seismic rays, *Geophysics,* **56,** 59–67.

Moser, T.J., Nolet, G. & Snieder, R., 1992a. Ray bending revisited, *Bull. seism. Soc. Am.,* **82,** 259–288.

Moser, T.J., Van Eck, T. & Nolet, G., 1992b. Hypocenter determination in strongly heterogeneous earth models using the shortest path method, *J. geophys. Res.,* **97,** 6563–6572.

Nakanishi, I. & Yamaguchi, K., 1986. A numerical experiment on nonlinear image reconstruction from first-arrival times for two-dimensional island arc structure, *J. Phys. Earth,* **34,** 195–201.

Nolet, G., 2008. *A Breviary of Seismic Tomography: Imaging the Interior of the Earth and Sun,* Cambridge Univ. Press.

Paige, C.C. & Saunders, M.A., 1982. LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Softw.,* **8,** 43–71.

Papazachos, C. & Nolet, G., 1997. *P* and *S* deep velocity structure of the Hellenic area obtained by robust nonlinear inversion of travel times, *J. geophys. Res.,* **102,** 8349–8367.

Pereyra V., Lee W.H.K. & Keller H.B., 1980. Solving two-point seismic ray tarcingt problems in a heterogeneous medium. Part 1. A general adaptive finite difference method, *Bull. seism. Soc. Am.,* **70,** 79–99.

Pullammanappallil, S.K. & Louie, J.N., 1993. Inversion of seismic reflection travel times using a nonlinear optimization scheme, *Geophysics,* **58,** 1607–1620.

Rawlinson, N. & Sambridge, M., 2003. Seismic traveltime tomography of the crust and lithosphere, *Adv. Geophys.,* **46,** 81–197.

Rawlinson, N. & Sambridge, M., 2004. Wave front evolution in strongly heterogeneous layered media using fast marching method, *Geophys. J. Int.,* **156,** 631–647.

Rawlinson, N. & Urvoy, M., 2006. Simultaneous inversion of active and passive source datastes for 3-D seismic structure with application to Tasmania, *Geophys. Res. Lett.,* **33,** L24313, doi:10.1029/2006GL028105.

Sallarès, V., Meléndez, A., Prada, M., Ranero, C.R., McIntosh, K. & Grevemeyer, I., 2013. Overriding plate structure of the Nicaragua convergent margin: relationship to the seismogenic zone of the 1992 tsunami earthquake, *Geochem. Geophys. Geosyst.,* **14,** 3436–3461.

Sambridge, M.S., 1990. Non-linear arrival time inversion: constraining velocity anomalies by seeking smooth models in 3-D, *Geophys. J. Int.,* **102,** 653–677.

Shaw, P.R. & Orcutt, J.A., 1985. Waveform inversion of seismic refraction data and applications to young Pacific crust, *Geophys. J. R. astr. Soc.,* **82,** 375–414.

Sheriff, R.E. & Geldart, L.P., 1995. *Exploration Seismology,* 2nd edn, Cambridge Univ. Press.

Taillandier, C., Noble, M., Charius, H. & Calandra, H., 2009. First-arrival traveltime tomography based on the adjoint-state method, *Geophysics,* **74,** doi:10.1190/1.3250266.

Thurber, C.H., 1983. Earthquake locations and three-dimensional crustal structure in the Coyote Lake area, central California, *J. geophys. Res.,* **88,** 8226–8236.

Toomey, D.R. & Foulger, G.R., 1989. Tomographic inversion of local earthquake data from the Hengill-Grensdalur Central Volcano Complex, Iceland, *J. geophys. Res.,* **94,** 17 497–17 510.

Toomey, D.R., Solomon, S.C. & Purdy, G.M., 1994. Tomographic imaging of the shallow crustal structure of the East Pacific Rise at 9°30′N, *J. geophys. Res.,* **99,** 24 135–24 157.

Um, J. & Thurber, C., 1987. A fast algorithm for two-point seismic ray tracing, *Bull. seism. Soc. Am.,* **77,** 972–986.

Van Avendonk, H.J.A., Harding, A.J., Orcutt, J.A. & Holbrook, W.S., 2001a. Hybrid shortest path and ray bending method for traveltime and raypath calculations, *Geophysics,* **66,** 648–653.

Van Avendonk, H.J.A., Harding, A.J., Orcutt, J.A. & McClain, J.S., 1998. A two-dimensional tomographyic study of the Clipperton transform fault, *J. geophys. Res.,* **103,** 17 885–17 899.

Van Avendonk, H.J.A., Harding, A.J., Orcutt, J.A. & McClain, J.S., 2001b. Contrast in crustal structure across the Clipperton transform fault from travel time tomography, *J. geophys. Res.,* **106,** 10 961–10 981.

Vidale, J., 1988. Finite-difference calculation of travel times, *Bull. seism. Soc. Am.,* **78,** 2062–2076.

Zelt, C.A. & Barton, P.J., 1998. Three-dimensional seismic refraction tomography: a comparison of two methods applied to data from the Faeroe Basin, *J. geophys. Res.,* **103,** 7187–7210.

Zelt, C.A. & Smith, R.B., 1992. Seismic travel time inversion for 2-D crustal velocity structure, *Geophys. J. Int.,* **108,** 16–34.

Zhang, J., ten Brink, U.S. & Toksöz, M.N., 1998. Nonlinear refraction and reflection travel time tomography, *J. geophys. Res.,* **103,** 29 743–29 757.

Zhang, J. & Toksöz, N., 1998. Nonlinear refraction traveltime tomography, *Geophysics,* **63,** 1726–1737.

SUPPORTING INFORMATION

Additional Supporting Information may be found in the online version of this paper:

Figure S1. Polygonal paths and ray trajectories for refractions and reflections calculated with TOMO3D for the traveltime and ray-path accuracy test in Fig. 2. Polygonal paths are obtained with the graph method and then used to build initial guess paths for the bending method, which produces the refined ray trajectories. Polygonal paths for refractions are plotted in (a, c, e), and the corresponding refracted paths after bending are plotted in (b, d, f). Analogously, polygonal paths for reflections are plotted in (g, i, k), and the corresponding reflected paths after bending are plotted in (h, j, l). The velocity model follows a 1-D linear gradient with depth which, together with the horizontal reflector, explains the symmetry exhibited by the ray paths.

Figure S2. Forward problem parallelization scheme. The number of sources (n) is compared to the number of available processors (k). If $k \geq n$, all sources are processed simultaneously. When that is not the case, the k processors start by taking care of the first k sources. When processor i has solved the graph problem for its current source i, it moves on to bend the rays of the corresponding m_i receivers equidistributing them among its j_i cores. Immediately after this task is completed, a new unsolved source $k + 1$ is assigned to processor i and so on so forth. This is repeated until the forward problem is solved for the n sources. Finally, the partial kernels for the n sources are added to calculate the total kernel. (http://gji.oxfordjournals.org/lookup/suppl/doi:10.1093/gji/ggv292/-/DC1).

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the paper.