Multiple Graph Realizations method: Improving the accuracy and the efficiency of the Shortest Path Method through random sampling.

Petros Bogiatzis*[1], Catherine A. Rychert[1] and Nicholas Harmon[1]

[1] Ocean and Earth Science, National Oceanography Centre Southampton, University of Southampton, Southampton, UK.

Corresponding author: Petros Bogiatzis (p.bogiatzis@soton.ac.uk)

Abstract

We present a new implementation of the shortest path method that calculates accurate travel times in arbitrarily large model spaces without the requirement of large computational times and large amounts of memory, an inherent problem of the Dijkstra's-like algorithms. The Multiple Graph Realizations method is based upon multiple sampling of the model space, using numerous random graphs. The performance of this new method is compared against the conventional way to improve the accuracy of shortest path method, which is to use denser grids and connectivity stencils of higher order. Our results suggest that although for relatively small models, single runs of the shortest path method are more suitable to achieve the desired accuracy, in large models, and after a certain level of desired accuracy, this approach becomes inefficient or even unfeasible, as the requirements in memory and computational time increases dramatically. On the contrary our method can achieve the desired accuracy with linear impact in computational time and negligible impact in required memory.

1. Introduction

Although full-waveform modelling has become much more efficient owing to increased computational power and particularly GPU based high performance computers, the rapid calculation of travel times in complex media remains among the most important tasks in seismology. This is particularly valuable for high frequency waves travelling over large regions, where solving for the full waveform remains computationally expensive. It is similarly

useful in that classical tomography approaches provide better starting models for full-waveform inversions, in turn reducing convergence times to reasonable models by several fold. Over the years many well-known approaches have been used to tackle the ray tracing problem, including: shooting method (e.g., Julian & Gubbins 1977, Červený 1987, Iwasaki 1988, Virieux & Farra 1991, Bulant 1996), bending (e.g., Julian & Gubbins 1977, K. Koketsu *et al.* 1991) and pseudo-bending (e.g., Pereyra *et al.* 1980, Um & Thurber 1987, Prothero *et al.* 1988, Edelsbrunner, Tan, *et al.* 1990, Moser *et al.* 1992, Kazuki Koketsu & Sekine 1998, Dahlen *et al.* 2000). All these methods attempt to find rays that solve the ray equation or satisfy Fermats's principle with the minimum computational cost.

Grid-based techniques have gained attention and become more popular over traditional ray tracing due to their robustness, efficiency and their ability to calculate travel times from a source to all the points of a grid that samples a medium, instead of just a receiver. They can be divided in two types of approaches, those that solve the eikonal equation explicitly and those that are based on the shortest path method (SPM).

The first type of grid-based approach was introduced by Vidale (1988, 1990) and later further advanced by others (e.g., Fuhao Qin *et al.* 1992, Schneider *et al.* 1992, Hole & Zelt 1995, Kim & Cook 1999). The work of Podvin & Lecomte (1991) introduced additional operators appropriate on high contrast velocity models. A crucial development was the introduction of the Fast Marching Method (FMM; Sethian 1996, 2001) for the robust and fast solution of the eikonal equation yielding successful applications in seismology (e.g., Rawlinson & Sambridge 2004, Lelièvre *et al.* 2011). FMM solves the eikonal equation by tracking an evolving interface only within a narrow band of grid-nodes around the wave front. Thus, it keeps the nodes behind this interface fixed and considers new nodes as the front propagates using an upwind entropy scheme. Recently, the Fast Sweeping Method (FSM) has also utilized to solve the eikonal equation (e.g., Tsai *et al.* 2003, Zhao 2004). The FSM successively "sweeps" the grid representing the model using an upwind finite-difference scheme, following a specific order. It performs Gauss-Seidel iterations in alternating directions and carries out as many traversals as necessary until the travel time in each cell has converged. It has similar accuracy with the FMM however its efficiency is largely related with the complexity of the velocity model (e.g., Chacon & Vladimirsky 2012, Gomez *et al.* 2019). Thus, it is generally faster than FMM in long wavelength, smooth models and slower than FMM in highly heterogeneous velocity fields. Recent developments include the acceleration of FMM and FSM methods using various parallel architectures for regular (Jeong et al., 2008; Weber et al., 2008) or irregular (Fu et al., 2013; Genellari et al., 2018) computational grids, reporting accelerations

by factors of $40\times$ to $200\times$. Other works focus on better handling the errors due to the singularity that appears in the case of a point source, by means of factorization of the traveltime function in the eikonal equation (e.g., Luo & Qian 2012; Treister & Haber, 2016).

The second type of grid-based approach, SPM, is grounded on the Dijkstra's algorithm (Dijkstra 1959) and was introduced by Nakanishi & Yamaguchi, (1986) and later by Saito, (1989, 1990) and Moser, (1991). In principle, it differs significantly from the fast-marching method, as it does not explicitly solve the eikonal equation, even if their practical algorithmic implementations are very similar, both imitating Huygens' principle. The SPM aims to find a shortest path on a graph, that is a network of grid points or nodes. Every node is connected only to a finite number of other nodes in its direct neighbourhood. Each connection, also known as edge, is characterized by a weight, which in the raytracing problem corresponds to the travel time between the two nodes. Graph nodes represent the discretization samples of the model space velocity or slowness, and then the shortest path is equivalent to the seismic ray path, by satisfying Fermat's principle of stationary time. Among its advantages are that it is unconditionally stable (e.g., Cheng & House, 1996; Dijkstra, 1959), and like the eikonal solvers, it provides travel times and ray paths from the source to all the nodes of the grid, allowing modelling of head and diffracted wave, first arrivals even in the presence of multipathing and waves in shadow zones. In addition, having the complete travel time field has other benefits such as the calculation of seismic amplitude (e.g., Vidale & Houston 1990). Moreover, the ability to model secondary phases has also been implemented (e.g., Moser 1991, Toomey *et al.* 1994, Zhang & Toksöz 1998, C.-Y. Bai *et al.* 2009, C. Bai *et al.* 2012), as well as versions that take into account anisotropic and viscoelastic media (e.g., Franklin, 1997; Li et al., 2020) The accuracy of the method can be improved by increasing node density and by including a larger number of connections (e.g., Klimeš & Kvasnička 1994, Mak & Koketsu 2011). Recent works atempt to increase the accuracy of the ray paths and overcome the subsequent computational burden by summing the edges vectorially as opposed to measuring the edge lengths and then scalarly summing them (e.g., Campen et al., 2013). It is generally comparable or even better than the FMM in terms of its speed and accuracy (e.g., Bai et al., 2009). One shortcoming relative to FMM is that SPM requires substantially more memory (e.g., Boisvert, 2010). Another limitation arises from the inherent sequential nature of Dijkstra's algorithm, which makes its parallel implementation challenging. Recent ongoing research (Malony *et al.* 2016, Monil *et al.* 2018) attempts to bypass this problem by using the Bellman-Ford-Moore (Ford 1956, Bellman 1958, Moore 1959) algorithm instead of Dijkstra's

method that allows scalability through the decomposition of the computational domain across processing nodes in a distributed high performance computing environment which can accommodate the increased memory and computational time requirements. Considerable effort has been devoted to improving the accuracy of the method by keeping its efficiency in acceptable levels in terms of computational time but mostly memory by optimizing the search method of the Dijkstra's algorithm, sparsifying or densifying the grid according to the structure of the model, optimizing the azimuthal coverage of connections or combining graph theory with the pseudo-bending method to fine tune the ray paths (Moser *et al.* 1992, Fischer & Lees 1993, Nolet & Moser 1993, Klimeš & Kvasnička 1994, Cheng & House 1996, Papazachos & Nolet 1997, Zhang & Toksöz 1998, Gruber & Greenhalgh 1998, Van Avendonk *et al.* 2001, C. Bai *et al.* 2007, C.-Y. Bai *et al.* 2009, Mak & Koketsu 2011, Bogiatzis *et al.* 2019)

In this work we compare the performance of SPM using a typical regular grid against a completely unstructured irregular grid of points that generated using a Delaunay-triangulation refinement method and a source-based polar concentric grid with and without rotation. Furthermore, we introduce a new approach, which utilizes multiple grid realizations (MGR) that gradually builds a shortest path "memory" in the model. We demonstrate that this approach reduces the memory requirements without significantly affecting the calculation time. We also compare this approach to finite difference waveform simulations and the FMM. We selected FMM over the FSM as it is widely used in seismology, they both have similar accuracy and furthermore its performance does not depend on the complexity of the velocity field. The MGR method allows the upscaling of the shortest path method in large domains with the desired accuracy, without running out of memory, while it is highly parallelizable.

2. The shortest path method

In the SPM, the model is described through a network, or a graph, which consists of nodes and connections between them. Each connection is characterized from a weight that defines the travel time required for the wave front to travel the internode distance. Following the common practice (e.g., Moser, 1991, Klimeš & Kvasnička 1994), for each edge, the slowness is calculated as the average between the two end points, assuming straight ray paths and an isotropic medium. Consequently, the delay between connected nodes is approximated by the numerical quadrature of slowness, with respect to the internode distance, using the trapezoidal rule. we approximate this travel time using the numerical quadrature of slowness with respect

to the internode distance, utilizing the trapezoidal rule. Then we define the connections of each node, also known as "Forward Star" (Moser 1991), by a distance threshold. The selection of this threshold is usually based on the maximum distance that a straight path can be considered an acceptable approximation of the true ray path. A rigorous analysis on the best practices for selecting this threshold can be found in Klimeš & Kvasnička (1994). All nodes and their connections are stored in a sparse matrix, **A**, also known as the adjacency matrix, so that $A(i,j)$ represents the time delay of the seismic wave when traveling from the node i to the node j. Due to reciprocity, the adjacency matrix is symmetric and thus only the upper or the lower triangle is required to be stored. After forming **A**, the travel times and, optionally, the ray paths are calculated through the application of Dijkstra's algorithm. In this work we use the implementation included in MATLAB 9.8 (R2020a), as the "graphallshortestpaths" function which is an interface for the pre-compiled 64bit MEX-file "graphalgs" that implements the Dijkstra's method. Other possibly faster codes (e.g., Steyvers & Tenenbaum 2005; "dijkstra_shortest_paths" from the boost C++ libraries, https://www.boost.org) can also be used and they are fully interchangeable as long as they are based upon the concept of the adjacency matrix. This formulation allows the application of the algorithm, without any modifications to either 2-D or 3-D problems but also to arbitrary grid types and geometries.

3. A review of three different grid-based graphs

Here we present three different grid types either commonly used with SPM or for potential use with SPM, namely a regular rectangular grid, a completely unstructured irregular and a concentric polar grid with and without rotation. We compare the performance of SPM with each one, using a medium of constant velocity, for which the analytical solution can be calculated.

3.1. Regular grid

If a grid has periodicity in space, i.e., a pattern which is repeated throughout the model, then it is considered regular. The most common regular grid is the rectangular one (Figure 1), which among other advantages allows the rapid determination of the surrounding grid nodes for any arbitrary point of the model. This feature allows the application of Dijkstra's method without requiring constructing the adjacency matrix in full and in advance, but instead by

directly finding the neighbouring nodes and the corresponding travel times on the fly, during the run of the algorithm with some additional computational cost due to the redundancy of repeated calculations but dramatic savings in memory.

However, periodicity of regular grids also results in systematic errors that propagate with increasing distance. Klimeš & Kvasnička (1994) showed that the maximum error as a result of the angle discretization will occur if the ray path systematically follows the most erroneous direction throughout its length. This is most likely in the case for a constant velocity medium in combination with a regular grid, where azimuthal gaps systematically occur at particular azimuths. Thus, the errors add-up with increasing distance from the source along these directions. The presence of heterogenicity makes the above scenario less probable because at least a portion of the ray path is likely to follow the true direction.

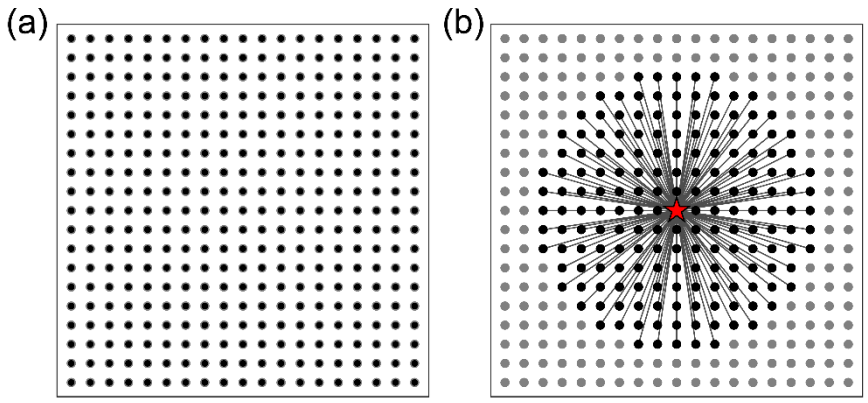

Figure 1: (a) Portion of a rectangular grid. (b) connectivity stencil based on distance threshold for an arbitrary node of the grid (star). Black nodes correspond to connected neighbours and grey to nodes with no direct connection (black lines) to the central node.

3.2. Irregular grid

To mitigate the azimuthal gap problem of a regular grid, a non-periodic irregular grid can be used instead (Figure 2). Such grids can be constructed by calculating Delaunay triangulations or equivalently Voronoi diagrams (e.g., Thacker 1980; Edelsbrunner *et al.* 1990; Bern & Eppstein 1992; Ruppert 1995, Si 2015; Engwirda & Ivers 2016). An optimal irregular grid avoids thin triangles, i.e. triangles with very small and consequently also large angles. In

a rectangular model with homogeneous grid-spacing this is usually a trivial task. However, when the model contains regions of strong heterogeneity such as a complex model boundary or discontinuity or topography, variable spacing is more suitable. Then the use of mesh-generating algorithms is necessary. Typically, these algorithms optimize the grid by maximizing the minimum angle of each triangle (e.g., Thacker 1980) or by minimizing the maximum angle of each triangle (e.g., Edelsbrunner *et al.* 1990). Either approach yields grids that generally reduce the maximum angular gap in the connections of any arbitrary grid point with its neighbours and thus minimize the error due to the take-off angle discretization (Moser 1991, Klimeš & Kvasnička 1994). It should be noted that satisfying the aforementioned Delaunay triangulation quality criteria comes under the primary requirement of non-periodicity of the grid that was mentioned earlier. For instance, a Delaunay triangulation of the 2-D regular grid that was examined in the previous section, yields a Delaunay triangulation with only equilateral triangles, thus the best possible angular quality, but it fails the criterion of non-periodicity. In this work we utilize the algorithm of Engwirda & Ivers (2016) to produce non-periodic high-quality grids.

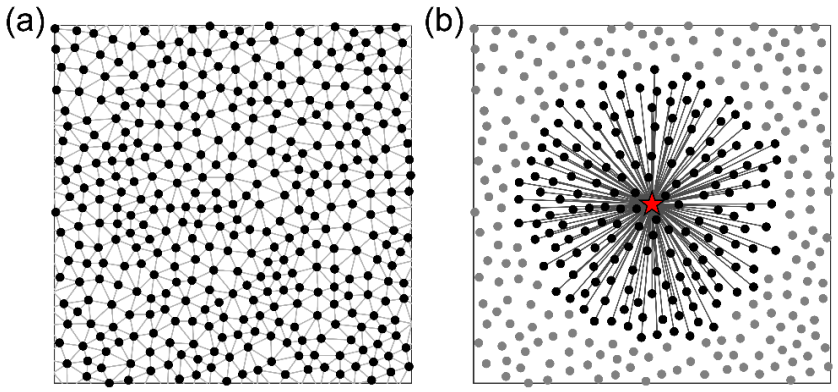

Figure 2: (a) Irregular grid and the underlying triangulation produced with the method of Engwirda & Ivers (2016). (b) Connectivity stencil based on distance threshold of an arbitrary node of the grid (star). Black points correspond to nodes with direct connection (black lines), while grey points to points without direct connection from the central point.

3.3. Polar, source-centred grid, with and without rotation

Another type of grid that we examine is a source-centered, concentric, polar grid. As in the two previous cases we construct a grid with a uniform distribution of grid nodes i.e., each node has a similar distance to all its neighbours. A demonstration of the connectivity stencil at the

source and an arbitrary point are shown in Figure 3. This grid can be readily constructed by warping concentric squares into concentric circles (e.g., Shirley & Chiu 1997). This approach avoids azimuthal oversampling at close distances from the source and under sampling at larger distances as for example happens when both polar coordinates i.e., angle and distance are sampled using a constant step (Figure S1b). More grid points can be added at the locations of receivers for improved accuracy. In this approach systematic azimuthal gaps are narrower than those of the regular grid, although they still exist (Figure S1c). We also present a case in which we mitigate this problem by applying a rotational deformation to the grid points. Working in polar coordinate system we change the angular coordinate φ, by $\theta\left(\frac{r}{r_{max}}\right)^{\gamma}$, where θ is an arbitrary angle, r is the radial coordinate, and γ controls how distortion scales with distance from the source (Figure S1d).

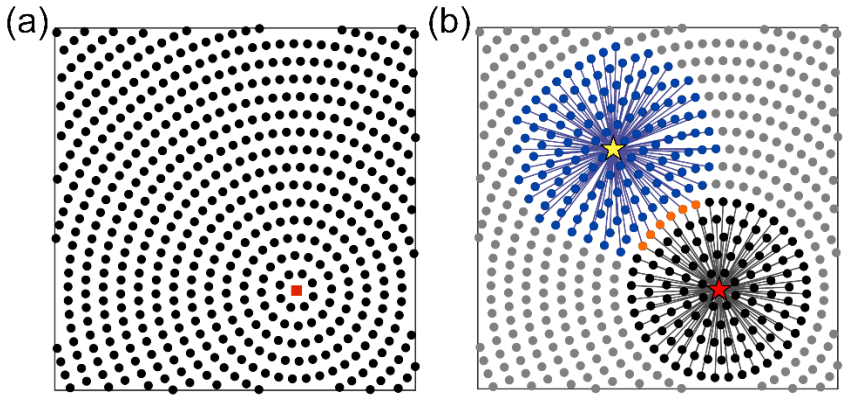

Figure 3: (a) Polar concentric grid around the source (red square). (b) Connectivity stencils based on the same distance threshold for the source (red star) and an arbitrary node (yellow star). Black nodes correspond to those with direct connection (black lines) with the source, blue nodes correspond to those with direct connection (blue lines) from the yellow star, while orange nodes have direct connections to both stars. Grey nodes do not share direct connections to any of the two stars.

3.4. Comparison in homogeneous space against the analytical solution

Next, we compare the SPM on the grids presented above with the analytical solution for a medium with constant velocity of 7 km/s. The model geometry is a 2-D rectangular area with $-300\ km \leq x, z \leq 300\ km$. The rectangular grid is generated by putting nodes every ~5.4 km

across both dimensions of the model. This yields a total of 12544 nodes. The irregular grid is calculated using a constraint of approximate 6 km spacing which yielded a total of 12366 nodes. Finally, the polar source-centered grid is constructed in a similar approach and radial spacing of ~6 km, yielding a total of 12387 nodes. All three grids have a similar number of nodes, with the rectangular regular grid having marginally the greatest number of nodes. All grids sample the model space in a uniform manner. The Adjacency matrix for all grids is calculated by defining for each node direct connections to all its neighbors that lie within distance of 20 km. This yields an Adjacency matrix with 267958 nonzero elements (connections) for the regular grid, 248047 elements for the irregular grid and 255627 elements for the polar source-centered grid. Again, the rectangular grid has the largest number of elements (although marginally), followed by the source-centered polar grid and finally the irregular grid. A source is assumed approximately at the middle of the model making sure that a node exists at this location for all the different grids. For the comparison we bilinearly interpolate the travel times calculated with each method in a denser rectangular grid of points with 1 km spacing in both dimensions.

The regular grid is the worst performer among the three methods, despite having the largest number of points and connections (Figure 4a and b). The pattern of errors presents prominent cross-, and x-shape error bands along the systematic azimuthal gaps of the neighborhood connectivity stencil (Figure 4a). The histogram of errors presents a wide distribution, extending to up to more than 0.55 s for some points of the medium. Along specific azimuths (e.g., 0^o, 90^o) the accuracy is nearly perfect where the nodes align perfectly with the ray path in homogenous media. This accuracy can only be fully exploited in special cases such as homogeneous earth and in very specific azimuths, as it deteriorates in the presence of arbitrary heterogeneities.

The irregular grid presents significantly better performance with errors that do not exceed ~0.15 s (Figure 4c, d). It is possible that azimuthal gaps exist in the irregular grid as well; however, due to the variations of the relative location of the grid points, they do not occur systematically at the same azimuths, as with the regular grid, and therefore the cumulative error over large distances is kept relatively low.

The unrotated polar grid presents performance comparable to the regular grid, although with lower maximum errors, < 0.47 s in comparison to 0.55 s (Figure 4f). The rotated polar source-centred grid shows similar performance with the irregular grid, with errors that are lower than 0.15 s (Figure 4e and f).

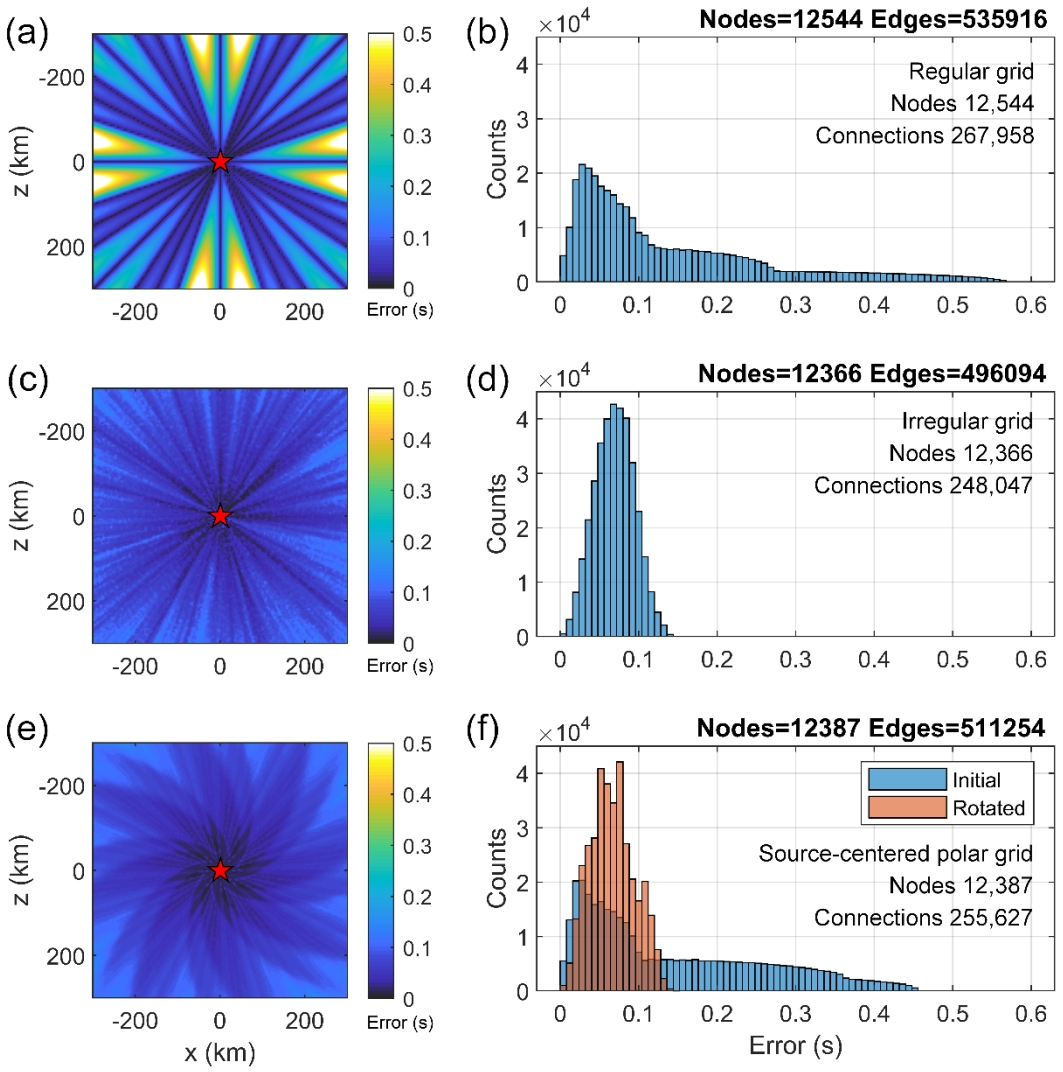

Figure 4: Comparison of the travel times between the different grid types and the analytical solution. (a) Error in (s) of the regular grid calculated as the difference between the travel times calculated with the regular grid against the analytical travel times and (b) the relative histogram. (c, b) Error similar to that in (a, b), but for the irregular grid. (e) Error of the travel times calculated in a rotated source-centred grid similar to that of Figure 3d, against the analytical travel times. (f) Histogram of the errors against the analytical travel times for the fixed (initial) and the rotated source centred grids. The red star indicates the location of the source.

Our results confirm various works (Moser 1991, Klimeš & Kvasnička 1994) showing the role of azimuthal gaps in the take-off angle of the shortest path method. When such gaps occur

systematically, as in the case of periodic grids, the error accumulates and gradually increases with increasing distance from the source. Non-periodic irregular grids avoid systematic azimuth gaps; however, the problem of the discretization of the take-off angle remains. The problem can be reduced by increasing the density of the grid points as well as the connectivity distance. However an important consideration is that although the angular resolution associated to the stencil improves with increasing connectivity distance, the scale of heterogeneity of the model poses an upper limit where the ray path between to neighboring nodes can be effectively approximated using the trapezoidal rule (e.g., Klimeš & Kvasnička 1994). Moreover, increasing the density of the grid and the connectivity distance is computationally expensive. For example, the number of connected neighbors per grid point, in the case of uniformly spaced points, will grow as $O\left(\left(\frac{r}{h}\right)^d\right)$, with r being the connectivity distance, h the average node spacing and d the spatial dimensions of the model. The complexity of Dijkstra's algorithm generally, increases as $O(log(N) * E)$, where N and E are the number of points of the grid and the total number of connections between the points, respectively (Barbehenn 1998). As mentioned earlier, probably the most significant problem is the required memory, especially in 3D applications, which poses a limit on the density of a grid as well as the number of connections and consequently to the accuracy of the method.

4. Multiple Graph Realizations (MGR) Method

Here we examine an alternative approach. Let us assume a reference grid of points that discretize the medium, in which we would like to calculate the travel time field from a hypothetical source. Using graph theory, the travel times can be approximated by applying Dijkstra's or - similar - algorithm using a graph which can be constructed as shown previously, to include the reference grid of points. Note that the nodes of the graph do not need to coincide with the reference grid points, and the travel times to these points can be estimated through interpolation from the estimated traveltimes at the nodes of the graph. Between a source and a receiver, a sequence of connected nodes would define an approximation to the real ray path. Assume now a series of graphs that are constructed by perturbing the location of the nodes of the initial graph except of the nodes that coincide with the source and the receiver. It is likely that there would be a graph where the nodes alignment would be more favorable to the real ray path and thus the travel time would be smaller, according to Fermat's principle (Figure 5a). Of course, the graph that happens to be more appropriate for one ray path may not be optimal for

another that corresponds to different receiver. Following this reasoning we apply Dijkstra's algorithm to multiple graph realizations, that are rapidly produced as randomly perturbed versions of the initial graph. Initially, the travel times from the source to all grid points are set to infinity. Each time that Dijkstra's algorithm is applied to a new graph realization, we interpolate the travel times to the reference grid of points by means of bilinear (trilinear in 3-D) interpolation, and we update the value of each point only if the new travel time improves the current one. Therefore, in the reference grid is stored the best score from all previous runs. However, to exploit this information in every new run, we also introduce a global connections stencil that connects the source directly with all nodes of each graph (Figure 5b). This translates in fully completing the row of the graph's adjacency matrix that corresponds to the index of the source. Note that the travel times of the global connections are not calculated with the method described in section 2 for connections between local neighboring nodes, but they are assigned directly with the best travel times from the source to the locations of all the nodes of the current graph calculated up until that point. We calculate these travel times again by means of bilinear/trilinear interpolation from the reference grid. The algorithmic steps of the Multiple Graph Realizations method are the following:

MGR Algorithm:

1: Construct a reference grid of N nodes and set $T_i^{ref} \rightarrow +\infty$, $i = 1, \ldots N$

2: Create new graph and its adjacency matrix

3: Interpolate T_i^{ref} to the nodes of the graph and update the adjacency matrix' global connections.

4: Apply Dijkstra's algorithm to get the travel times.

5: Interpolate the travel times back to the reference grid T_i^n.

6: Update reference grid nodes as, $T_i^{ref} = \min\left(T_i^{ref}, T_i^n\right)$

7: If termination criterion is met, stop. Else, go to 2.

As a termination criterion, one can monitor the improvement between successive iterations and use a threshold in either the average or maximum travel-time subsequent improvement. Alternatively, a predefined number of iterations can be used, based upon previous testing. The additional memory requirement for the MGR method is only marginal as we demonstrate later, and proportional to the cardinality of the graph. It requires storing the optimal travel times from the source to the nodes of the reference grid, which are updated in each iteration. It also requires the full completion of one row of the adjacency matrix, which normally is partially completed

with the local connections of the source. The computational time scales linearly with the number of realizations.

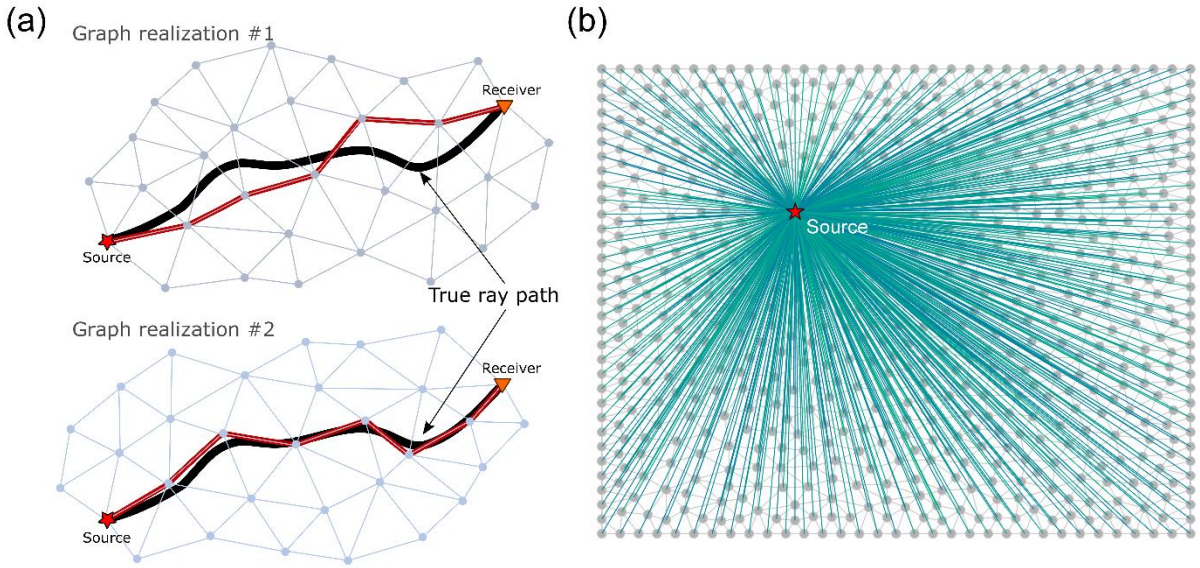

Figure 5: Schematic showing the two basic concepts of the MGR method. (a) Two versions of the shortest path (red lines) between the same source (star) and a receiver (triangle) pair, corresponding in two different graph realizations (top and bottom) that describe the same background velocity model. (b) Global connections stencil than connects directly the source (star) to all nodes of the graph (edges colored with shades of green), on top of the typical local connectivity of the graph (gray edges).

4.1. Performance Evaluation of the MGR method

We demonstrate this approach using the irregular grid architecture as it is described above. The initial grid is built as explained in section 3.2. However, for efficiency, the rest of the grid realizations are derived by applying random perturbations on the initial coordinates of each grid point, rather than constructing new irregular grids from the beginning. Therefore, the perturbed grids are not explicitly optimized for quality, and instead we rely on the relatively small amplitude of the perturbations and the large number of realizations. The new coordinates do not exceed the initial average internode distance and the source coordinates are fixed in all realizations.

We test the method initially in a constant velocity model and we compare against the analytical solution (Figure 6). The travel times present significant improvement after the first ~30 realizations and then the rate gradually drops, reaching RMS error bellow 0.01 s, after approximately 130 iterations.

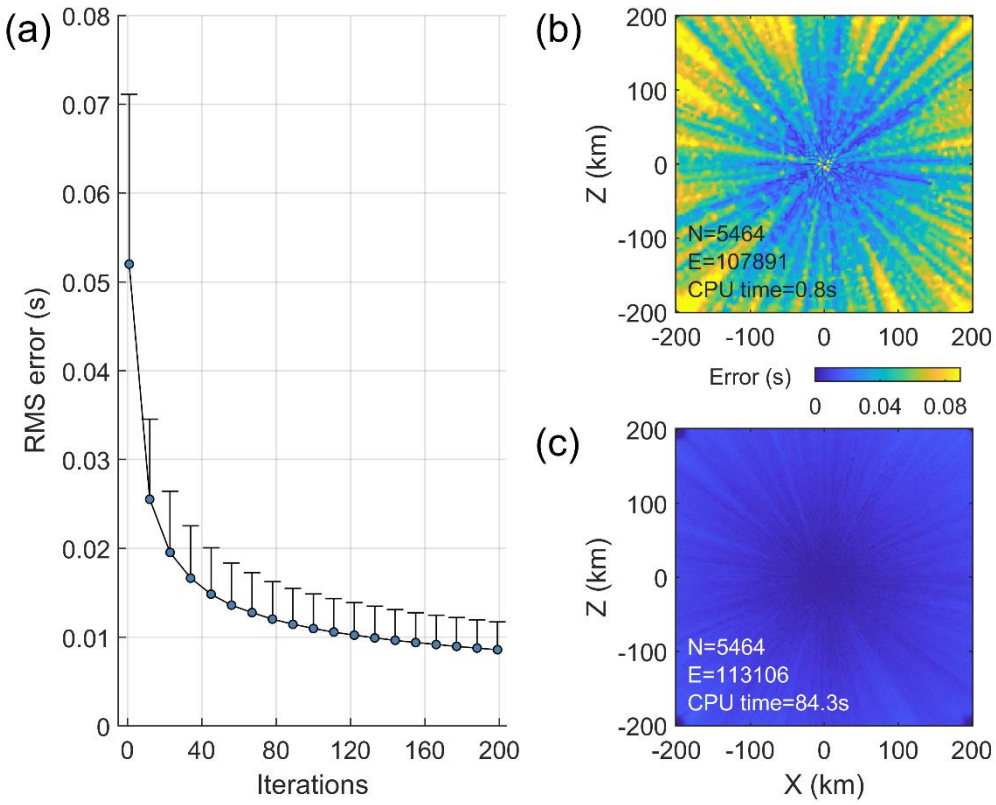

Figure 6: Results of the MGR method in a medium of constant velocity, 7 km/s. (a) Root mean square error (s) as a function of iterations. Error bars correspond to one standard deviation. (b) Error after the first iteration (s) and (c) error (s) after 200 iterations. In both (b) and (c) are also shown the number of graph nodes (N), the number of connections (E), and the average computation time.

Next, we test the performance of the MGR method in a complex model, which simulates a typical subduction zone (Figure 7). The initial grid has average node distance of 7 km and the connectivity threshold is 20 km. This yields 5547 nodes and 87439 connections. A hypothetical source is located at the top of the subducting plate at a depth of 104 km (Figure 7).

We also compare the traveltimes from MGR and SPM to two other methods for modelling wavefields and their traveltimes given the absence of an analytical solution. The first is a fast marching eikonal solver that uses second order derivatives (Hassouna & Farag 2007, Dirk-Jan Kroon 2020), applied in a grid with spacing of 100 m in both directions. The second is a finite difference full waveform elastic simulation algorithm (SOFI2D) (Bohlen *et al.* 2015), utilizing a sixth order finite difference stencil in space and 2nd order in time with a grid spacing of 138 m in both directions. For the simulation we use an explosive source described by a Ricker wavelet of central frequency ~1.5 Hz and maximum frequency ~3Hz. To calculate the travel times from the finite difference simulation we compute seismograms for 476 receivers which uniformly sampled the modelling domain in both dimensions every 20.7 km and pick the onset of the first arrivals in normalized seismograms to mitigate the geometrical spreading effect. The onset time of the first arrival is determined as the time that each seismogram exceeds an amplitude threshold that separates the signal from the numerical noise. We determine this threshold as the value that minimizes the error with the analytical solution in a homogeneous model. The RMS picking error for the homogeneous case is 0.001 s. The arrival times from the finite difference simulation are compared to the MGR method at the locations of the receivers. We compare all travel time results against the initial reference graph.

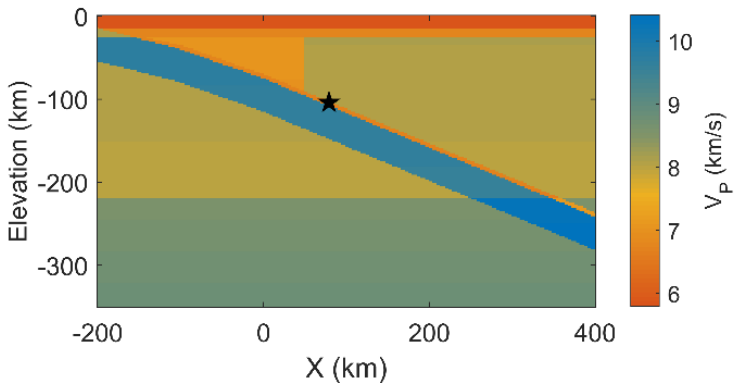

Figure 7: Heterogeneous model of V_P velocity that represents a typical subduction structure. The model is produced with PREM (Dziewonski & Anderson 1981) as a background, by adding a ~20% faster subducting slab. The model also includes a slow mantle wedge, 15% slower and a thin low velocity layer on top of the slab.

According to Fermat's principle, under the high frequency approximation, a smaller travel time from the source to an arbitrary point of the medium is a closer to the one corresponding to the real ray path, which is our goal here. For this reason, we use as a measure of the improvement the mean signed travel time difference from the initial graph. A total of 200 grid realizations are calculated according to the MGR method and the results are shown in Figure 8. All computations for the MGR method are performed using one core of an intel i7-8850U 1.8Ghz CPU, in double precision and 16 GB of main memory. The calculations are performed using the software described in section 2. The minimum memory requirement in bytes to store the adjacency matrix is calculated as (Gilbert *et al.* 1992), $E \times (i + d) + (N + 1) * i$, where N is the number of nodes in the graph (cardinality), E is the number of connections, i is the size of an integer variable and d the size of a double precision variable in bytes.

In this work the calculations take place in a 64-bit system therefore $i = d = 8$ bytes. In high level languages such as MATLAB where the user does not fully control the allocation procedure, often the real allocated space can be somewhat larger. Following this concept, the adjacency matrix takes ~1.4 Mb of main memory and the total average computation time for 200 iterations is ~100 s, including ~1s for the construction of the initial grid, which corresponds to ~0.5 s for each iteration. The average traveltime improvement relative to the initial coarse grid is better than >0.3 s at the end of the 200 iterations.

Next, we test a series of single evaluations of Dijkstra's algorithm with reduced internode distances of the graph (Figure 8; Table 1). The connectivity distance threshold is kept unchanged at 20 km, which yields an implicit increase of the number of connections as the nodes' distance decreases. Other combinations of node-spacing and connectivity distances have also been tested showing similar behaviour (e.g., Figure S2). We report the improvement in travel time accuracy, as well as the impact in computational time and the required memory for the adjacency matrix.

Initially, we decrease the node distance to 2.6 km. The improvement in average travel times is ~0.22 s, similar with the result of the MGR method after 50 iterations. The computation time for the single run is just ~10 s, less than half compared to the ~24 s of the MGR approach (Figure 8). However, the required memory for storing the adjacency matrix jumps to ~63 MB as the number of nodes (38,069) and connections (4,107,633) increases. A further decrease of the node distance at 1.8 km, decreases the average travel time difference to ~0.26 s, which compares with the performance of the MGR method after 100 iterations. The computational time of the single run is ~54 s that is comparable to the ~47s of the MGR method, suggesting

the break-even point between the two methods is near this node distance. The size of the adjacency matrix further increases to 275 MB which two orders of magnitude larger than the memory required from the MGR method. The following decrease in node distance to 1.3 km decreases the average travel times difference to 0.28 s, matching the performance of the MGR method after 140 iterations. However, now the computational time of the single run is more than 7 minutes (436 s), about 7 times more than the MGR approach. The required memory for storing the adjacency matrix has also increased significantly to ~ 1 GB. A further reduction in nodes' distance to 1.0 km marginally improves the average travel time difference from the starting model to 0.29 s. The computational time is 1 hour 49 minutes and 32 s. The adjacency matrix requires at least 2.80 GB of memory. MGR method achieves similar results in only 77 s, without requiring additional memory. Further reduction of the internode distance to 0.9 km failed due to running out of memory. In Figure 8b and c, we compare the best possible result from a single run of the shortest path method against the MGR approach. Note that the spatial distribution of the differences in travel times is generally similar between the two different approaches. A summary of the comparison between the MGR method and the single runs of the shortest path method with different internode distances can be found in Table 1.

The full-waveform finite difference method also produces travel times that are in good agreement with to the SPM and MGR methods, verifying the accuracy of the approximation. The finite difference method has a mean signed differences of ~ -0.04 s and a histogram of the differences is shown in Figure S4. The fact that travel times are not as low as those reached by MGR is likely owing to finite frequency effects. A full waveform simulation is the best approximation to the seismic wavefield, as it accounts for finite frequency effects that are not modelled in the high frequency approximation and ray theory (e.g., Hung et al., 2000; Spetzler & Snieder, 2004) which create differences in the first arrival times due to waveform interference, scattering, etc. In a full waveform simulation, the finite frequency content of the source wavelet senses the medium's structure differently, compared with the high frequency approximation, presenting a wavelength-dependent averaging of the elastic properties in the case of full waveform simulation (Lomax 1994, Snieder & Lomax 1996). The total runtime for the finite difference simulation took 4 hours and 26 minutes on a workstation with 32 CPU threads for 60 s of wave propagation and reduced the travel time to a mean signed difference of -0.04 s (Figure 8, blue dashed line). The computational cost of the finite difference method is significantly higher than either FMM or SPM or MGR even with high density spatial sampling making it less attractive for generating initial velocity models.

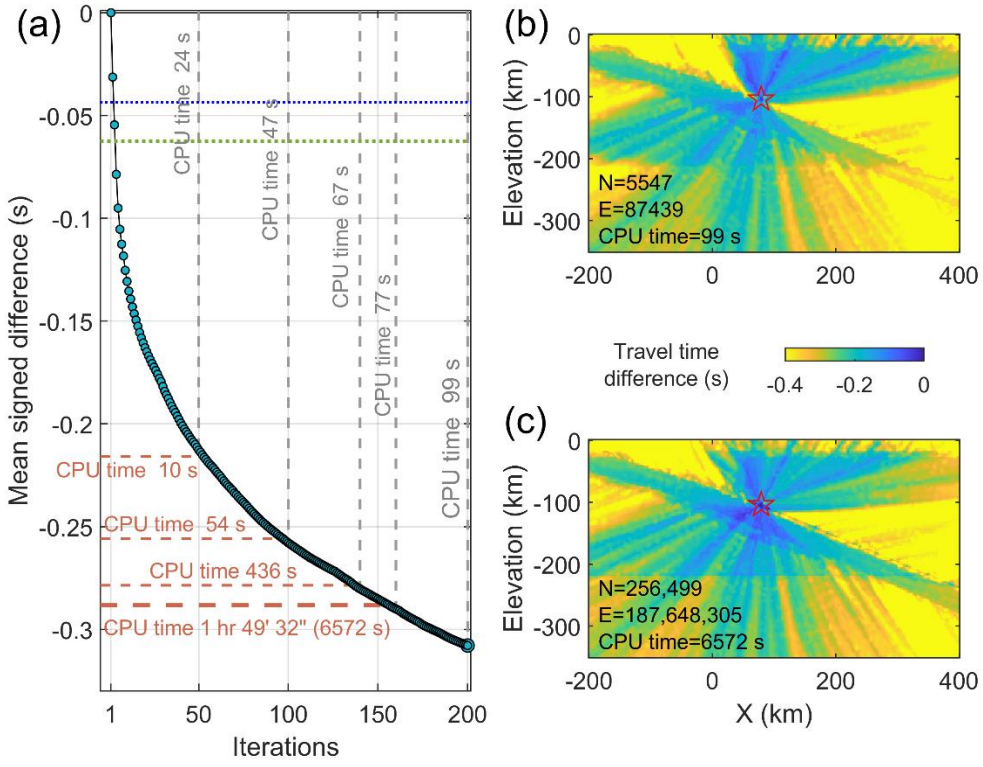

Figure 8: Results of the MGR method for the heterogeneous model shown in Figure 7. (a) Mean signed difference from the initial graph as a function of the iteration number. Vertical dashed lines indicate the computation time of the MGR method for various iterations. The horizontal dashed orange lines show the corresponding computational time needed for achieving similar improvement in travel times by one application of the shortest path method in a denser graph. From top to bottom, for average node distance of 2.6 km, 1.8 km, 1.3 km and 1.0 km. The horizontal blue dotted line corresponds to the comparison of the average travel time difference of a full waveform finite difference simulation and the horizontal green dotted line corresponds to the same result from the fast marching approach (b) Travel time improvement (s) compared to the initial graph derived with the MGR method after 200 iterations. (c) Travel time Improvement using a graph with average node distance of 1.0 km that corresponds to the lowermost horizontal dashed line in (a). In both panels (b) and (c) the number of graph nodes (N), the number of connections (E), and the average computation time (CPU time) are reported.

The FMM reduced traveltimes similar to MGR and SPM, although the reduction is more limited owing to several factors including, in the end, the fact that the storage limit was reached (Figure 8. The FMM reduced the travel time relative to the reference grid by a mean signed difference of -0.06 s. (Figure 8, green dashed line). The reduction is greater than that of the finite difference, as it is for MGR and SPM, again given the ray theoretical, implicit infinite frequency approximation. There are several other factors that contribute to the difference in the final MGR travel times and the FMM result. This could be in part due to the regular grid structure of the FMM relative to the complicated structure of the subduction zone model, which even though there is dense spatial sampling the MGR method effectively samples the space at an even higher density. Furthermore, the maximum internode distance of the second order stencil used from the FMM algorithm, in the case of 100 m spacing extends up to 200 m and drops to 100 m for 50 m spacing. On the contrary in the case of the SPM method the maximum connectivity length is ~20 km allowing a larger proportion of ray paths in relatively homogeneous parts of the model to be approximated with smoother trajectories. The computational time for the FMM with 100 m spacing is ~12 s, a few seconds greater than the MGR method, with lower memory requirements, for a similar level of accuracy. Reducing the spacing at 50 m improves the accuracy only marginally (average difference of ~0.001 s), while further reduction at 10 m is not possible as it causes exceedance of the available memory (Figure S3). Figure S4 shows histograms with the traveltime comparisons of the full waveform and the FMM results against with the starting SPM graph and the MGR approach. It is worth noting that as it is shown the computational requirements of conventional SPM but also FMM can be significant even in the 2-D case when the number of nodes and the order of the stencil increase to achieve accuracy above a certain level. This level can be surpassed easier in the 3-D case.

We test in a similar manner our method in a 3-D model, which is constructed from the aforementioned 2-D model by extending it along the strike of the subduction zone (i.e., y-direction). The results (Figure S5; Table S1) are generally similar to the 2-D case however, the accuracy level where the conventional SPM efficiency starts degrading is significantly lower. The connectivity distance threshold is kept at 20 km and the for the MGR method the grid is constructed with maximum internode distance of ~15 km. For internode distances larger or equal ~8 km single runs of the shortest path method are more efficient for achieving the same improvement in traveltimes. However, for further improvement, the MGR method becomes significantly more efficient (Table S1; Figure S5). The minimum internode distance for a single run we manage to achieve before running out of memory was 4.4km which took ~2.2 hours to

run. The MGR method got to the same result within ~444 seconds without requiring additional memory. The time for the construction of the initial grid was ~3.5 s. After 200 iterations that required 936 s the average improvement was 2.659 s.

***Table 1**: Summary of results of MGR method against single runs with increasing grid density.*

Method	**Grid spacing (km)**	**Nodes**	**Connections**	**Adjacency matrix size (MB)**	**Average travel time reduction (s)**	**CPU time (s)**
MGR 200 iterations	7.0	5,547	81,949	1	0.31	99
Single run	2.6	38,069	4,107,633	63	0.22	10
	1.8	79,731	17,977,854	275	0.26	54
	1.3	151,661	65,703,490	1,004	0.28	436
	1.0	256,499	187,648,305	2,865	0.29	6,572

5. Conclusions

In this work we compare three different grid architectures as the basis for the shortest path method. We see that irregular grids with limited periodicity perform better in homogeneous media compared with the analytical solution. The main advantage over a regular grid is that the absence of spatial periodicity prohibits systematic errors from accumulating along specific azimuths.

To further reduce the systematic errors in the shortest path method, we introduce the Multiple Graph Realizations method, which takes advantage of multiple random perturbations of an initial grid to further improve the travel times calculation. During the procedures the algorithm remembers the best scores and directly passes them into Dijkstra's algorithm by updating the adjacency matrix. Application in both homogeneous and heterogeneous media showed that the method can significantly reduce the travel times relative to a coarse grid and in accordance with Fermat's principle for ray theory. Compared to the typical alternative

approach of reducing the grid spacing and using higher numbers of connections in the forward star, the MGR method requires significantly less memory as the working grid can be quite sparse. For a relatively small number of nodes and connections the MGR method takes more time to achieve the same improvement as one would get with a denser grid and a larger forward star. However, as the number of nodes and connections increases, due to a requirement for denser or larger grids or in 3-D domains, the improvement of the MGR method to the computational time and the memory requirements are dramatic (e.g., Figure 8; Table 1; see also Figure S5 and Table S1 for the 3-D case). In the subduction example in this paper it was not feasible to decrease the node distance less than 1.0 km because of memory requirements. Even with larger available memory, the computational time is several orders of magnitude larger than the one achieved by the MGR approach. MGR travel times are similar to those from finite different elastic waveform simulations in the subduction zone test we performed here, although MGR requires less computational time and memory. The MGR travel times are also similar to those from FMM, although MGR requires less computational time and memory and also has the potential to achieve higher accuracy. It should be noted that typical discretization of eikonal equation solvers including the FMM implementation used here for the comparison, in Cartesian grids suffer from bias along the coordinate axes and near the region near the point source where the solution is singular (e.g., Figure S6). Recent works use source factorisation in an attempt to overcome this problem (e.g., Luo & Qian 2012; Treister & Haber, 2016). The FMM used here does not include this, and therefore comparisons to FMM including source factorisation will be a topic of future work.

An interesting point and a future direction of this research is to examine different strategies in grid realizations during the iterations of the MGR algorithm. Although in this case random perturbations proved to be effective, optimized grid perturbations and realizations could further improve the required number of iterations and the overall accuracy of the method. For example, switching between different grid architectures throughout iterations, one can take advantage of their strong points in different regions of the model and different azimuths. Another direction is dynamically adjusting the sampling region of the graph. For instance, during the first iterations only in a limited region around the source and gradually expanding throughout the model space.

Finally, it is worth investigating the parallelization possibilities of the MGR algorithm. In a possible CPU-based implementation each CPU can be assigned with a graph realization. One consideration is the requirement that all workers should be able to asynchronously access and

update the memory from previous runs, that is the reference grid. Nevertheless, the communication overhead is not overly prohibitive, as this translates to only N floating precision numbers before and after each individual run. The transmission of the reference grid coordinates ($N \times 2$ in 2D or $N \times 3$ in 3D floating point numbers) is required only once and takes place at the beginning of the procedure.

Acknowledgements

We would like to thank the two anonymous reviewers for their constructive comments that helped improve this manuscript. We would also like to acknowledge funding from the Natural Environment Research Council (NE/M003507/1) and the European Research Council (GA 638665). No data have been used in this work.

References

Avendonk, H.J.A. Van, Harding, A.J., Orcutt, J.A. & Holbrook, W.S. (2001) Hybrid shortest path and ray bending method for traveltime and raypath calculations. *Geophysics*, **66**, 648–653. doi:10.1190/1.1444955

Bai, C.-Y., Tang, X.-P. & Zhao, R. (2009) 2-D/3-D multiply transmitted, converted and reflected arrivals in complex layered media with the modified shortest path method. *Geophys. J. Int.*, **179**, 201–214. doi:10.1111/j.1365-246X.2009.04213.x

Bai, C., Greenhalgh, S. & Zhou, B. (2007) 3D ray tracing using a modified shortest-path method. *Geophysics*, **72**, T27–T36. doi:10.1190/1.2732549

Bai, C.Y., Li, X.L., Wang, Q.L. & Peng, J.B. (2012) Multiple arrival tracking within irregular triangular or tetrahedral cell model. *J. Geophys. Eng.*, **9**, 29–38. doi:10.1088/1742-2132/9/1/004

Barbehenn, M. (1998) A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. *IEEE Trans. Comput.*, **47**, 263. doi:10.1109/12.663776

Bellman, R. (1958) On a routing problem. *Q. Appl. Math.*, **16**, 87–90. doi:10.1090/qam/102435

Bern, M. & Eppstein, D. (1992) Mesh generation and optimal triangulation, pp. 23–90.

doi:10.1142/9789814355858_0002

Bogiatzis, P., Ishii, M. & Davis, T.A. (2019) The Dulmage–Mendelsohn permutation in seismic tomography. *Geophys. J. Int.*, **218**, 1157–1173. doi:10.1093/gji/ggz216

Bohlen, T., Daniel, K. & Jetschny, S. (2015) SOFI2d , seismic modeling with finite differences, 2D - acoustic and viscoelastic version, Users Guide. *Karlsruhe Inst. Technol.*

Boisvert, J.B. (2010) On the Use of Fast Marching Algorithms for Shortest Path Distance Calculation. *Cent. Comput. Geostatistics Rep. **12**, 113.*, Alberta. Retrieved from http://www.ccgalberta.com/ccgresources/report12/2010-113_fmm_for_shortest_distance.pdf

Bulant, P. (1996) Two-point ray tracing in 3-D. *Pure Appl. Geophys. PAGEOPH*, **148**, 421–447. doi:10.1007/BF00874574

Campen, M., Heistermann, M. & Kobbelt, L. (2013) Practical Anisotropic Geodesy. *Comput. Graph. Forum*, **32**, 63–71. doi:10.1111/cgf.12173

Červený, V. (1987) Ray tracing algorithms in three-dimensional laterally varying layered structures. in *Seismic Tomography*, pp. 99–133, Dordrecht: Springer Netherlands. doi:10.1007/978-94-009-3899-1_5

Chacon, A. & Vladimirsky, A. (2012) Fast Two-scale Methods for Eikonal Equations. *SIAM J. Sci. Comput.*, **34**, A547–A578. doi:10.1137/10080909X

Cheng, N. & House, L. (1996) Minimum traveltime calculation in 3-D graph theory. *Geophysics*, **61**, 1895–1898. doi:10.1190/1.1444104

Dahlen, F.A., Hung, S.-H. & Nolet, G. (2000) Fréchet kernels for finite-frequency traveltimes-I. Theory. *Geophys. J. Int.*, **141**, 157–174. doi:10.1046/j.1365-246X.2000.00070.x

Dijkstra, E.W. (1959) A note on two problems in connexion with graphs. *Numer. Math.* doi:10.1007/BF01386390

Dirk-Jan Kroon. (2020) Accurate Fast Marching, Mathworks. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/24531-accurate-fast-marching

Dziewonski, A.M. & Anderson, D.L. (1981) Preliminary reference Earth model. *Phys. Earth*

Planet. Inter., **25**, 297–356, Elsevier. doi:10.1016/0031-9201(81)90046-7

Edelsbrunner, H., Seng Tan, T. & Waupotitsch, R. (1990) O(n2 log n) time algorithm for the minmax angle triangulation. doi:10.1137/0913058

Edelsbrunner, H., Tan, T.S. & Waupotitsch, R. (1990) An O(n 2 log n) time algorithm for the MinMax angle triangulation. *Proc. sixth Annu. Symp. Comput. Geom. - SCG '90*, pp. 44–52, New York, New York, USA: ACM Press. doi:10.1145/98524.98535

Engwirda, D. & Ivers, D. (2016) Off-centre Steiner points for Delaunay-refinement on curved surfaces. *Comput. Des.*, **72**, 157–171. doi:10.1016/j.cad.2015.10.007

Fischer, R. & Lees, J.M. (1993) Shortest path ray tracing with sparse graphs. *Geophysics*, **58**, 987–996. doi:10.1190/1.1443489

Ford, L.R. (1956) Network flow theory, Santa Monica, Calif. : Rand Corp.

Franklin, J. (1997) Minimum traveltime calculations in anisotropic media using graph theory. SEG Tech. Progr. Expand. Abstr. 1997, 1517–1520, Society of Exploration Geophysicists. doi:10.1190/1.1885704

Fu, Z., Kirby, R.M. & Whitaker, R.T. (2013) A Fast Iterative Method for Solving the Eikonal Equation on Tetrahedral Domains. *SIAM J. Sci. Comput.*, **35**, C473–C494. doi:10.1137/120881956

Ganellari, D., Haase, G. & Zumbusch, G. (2018) A massively parallel Eikonal solver on unstructured meshes. *Comput. Vis. Sci.*, **19**, 3–18. doi:10.1007/s00791-018-0288-z

Gilbert, J.R., Moler, C. & Schreiber, R. (1992) Sparse Matrices in MATLAB: Design and Implementation. *SIAM J. Matrix Anal. Appl.*, **13**, 333–356. doi:10.1137/0613024

Gomez, J. V., Alvarez, D., Garrido, S. & Moreno, L. (2019) Fast Methods for Eikonal Equations: An Experimental Survey. *IEEE Access*, **7**, 39005–39029. doi:10.1109/ACCESS.2019.2906782

Gruber, T. & Greenhalgh, S.A. (1998) Precision analysis of first-break times in grid models. *Geophysics*, **63**, 1062–1065. doi:10.1190/1.1444384

Hassouna, M.S. & Farag, A.A. (2007) MultiStencils Fast Marching Methods: A Highly Accurate Solution to the Eikonal Equation on Cartesian Domains. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**, 1563–1574. doi:10.1109/TPAMI.2007.1154

Hole, J.A. & Zelt, B.C. (1995) 3-D finite-difference reflection travel times. *Geophys. J. Int.*, **121**, 427–434. doi:10.1111/j.1365-246X.1995.tb05723.x

Hung, S.-H., Dahlen, F.A. & Nolet, G. (2000) Fréchet kernels for finite-frequency traveltimes-II. Examples. *Geophys. J. Int.*, **141**, 175–203. doi:10.1046/j.1365-246X.2000.00072.x

Iwasaki, T. (1988) Ray-tracing Program for Study of Velocity Structure by Ocean Bottom Seismographic Profiling. *Zisin (Journal Seismol. Soc. Japan. 2nd ser.)*, **41**, 263–266. doi:10.4294/zisin1948.41.2_263

Jeong, W.-K. & Whitaker, R.T. (2008) A Fast Iterative Method for Eikonal Equations. *SIAM J. Sci. Comput.*, **30**, 2512–2534. doi:10.1137/060670298

Julian, B.R. & Gubbins, D. (1977) Three-dimensional seismic ray tracing. *J. Geophys.*, **43**, 95–113.

Kim, S. & Cook, R. (1999) 3-D traveltime computation using second-order ENO scheme. *Geophysics*, **64**, 1867–1876. doi:10.1190/1.1444693

Klimeš, L. & Kvasnička, M. (1994) 3-D network ray tracing. *Geophys. J. Int.*, **116**, 726–738. doi:10.1111/j.1365-246X.1994.tb03293.x

Koketsu, K., Kennett, B.L.N. & Takenaka, H. (1991) 2-D reflectivity method and synthetic seismograms for irregularly layered structures-II. Invariant embedding approach. *Geophys. J. Int.*, **105**, 119–130. doi:10.1111/j.1365-246X.1991.tb03448.x

Koketsu, Kazuki & Sekine, S. (1998) Pseudo-bending method for three-dimensional seismic ray tracing in a spherical earth with discontinuities. *Geophys. J. Int.*, **132**, 339–346. doi:10.1046/j.1365-246x.1998.00427.x

Lelièvre, P.G., Farquharson, C.G. & Hurich, C.A. (2011) Computing first-arrival seismic traveltimes on unstructured 3-D tetrahedral grids using the Fast Marching Method. *Geophys. J. Int.*, **184**, 885–896. doi:10.1111/j.1365-246X.2010.04880.x

Li, X.-W., Zhou, B., Bai, C.-Y. & Wu, J.-L. (2020) Seismic complex ray tracing in 2D/3D viscoelastic anisotropic media by a modified shortest-path method. *Geophysics*, **85**, T331–T342. doi:10.1190/geo2020-0113.1Lomax, A. (1994) The Wavelength-Smoothing Method For Approximating Broad-Band Wave Propagation Through Complicated Velocity Structures. *Geophys. J. Int.*, **117**, 313–334. doi:10.1111/j.1365-

246X.1994.tb03935.x

Luo, S. & Qian, J. (2012) Fast Sweeping Methods for Factored Anisotropic Eikonal Equations: Multiplicative and Additive Factors. J. Sci. Comput., 52, 360–382. doi:10.1007/s10915-011-9550-y.

Mak, S. & Koketsu, K. (2011) Shortest path ray tracing in cell model with a second-level forward star. *Geophys. J. Int.*, **186**, 1279–1284. doi:10.1111/j.1365-246X.2011.05103.x

Malony, A.D., Monil, M.A.H., Rasmusen, C., Huck, K., Byrnes, J. & Toomey, D. (2016) Towards Scaling Parallel Seismic Raytracing. *2016 IEEE Intl Conf. Comput. Sci. Eng. IEEE Intl Conf. Embed. Ubiquitous Comput. 15th Intl Symp. Distrib. Comput. Appl. Bus. Eng.*, pp. 225–233, IEEE. doi:10.1109/CSE-EUC-DCABES.2016.189

Monil, M.A.H., Malony, A.D., Toomey, D. & Huck, K. (2018) Stingray-HPC: A Scalable Parallel Seismic Raytracing System. *2018 26th Euromicro Int. Conf. Parallel, Distrib. Network-based Process.*, pp. 204–213, IEEE. doi:10.1109/PDP2018.2018.00035

Moore, E.F. (1959) The shortest path through a maze. *Int. Symp. Theory Switch. Part II*.

Moser, T.J. (1991) Shortest path calculation of seismic rays. *Geophysics*, **56**, 59–67. doi:10.1190/1.1442958

Moser, T.J., Nolet, G. & Snieder, R. (1992) Ray bending revisited. *Bull. - Seismol. Soc. Am.*, **82**, 259–288.

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. doi:10.4294/jpe1952.34.195

Nolet, G. & Moser, T.-J. (1993) Teleseismic Delay Times In A 3-D Earth and A New Look At the S Discrepancy. *Geophys. J. Int.*, **114**, 185–195. doi:10.1111/j.1365-246X.1993.tb01478.x

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. Solid Earth*, **102**, 8349–8367. doi:10.1029/96JB03730

Pereyra, V., Lee, W.H.K. & Keller, H.B. (1980) Solving Two-Point Seismic-Ray Tracing Problems In A Heterogeneous Medium. *Bull. Seismol. Soc. Am.*, **70**, 79–99.

doi:10.1017/CBO9781107415324.004

Podvin, P. & Lecomte, I. (1991) Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools. *Geophys. J. Int.*, **105**, 271–284. doi:10.1111/j.1365-246X.1991.tb03461.x

Prothero, W.A., Taylor, W.J. & Eickemeyer, J.A. (1988) A fast, two-point, three-dimensional raytracing algorithm using a simple step search method. *Bull. - Seismol. Soc. Am.*, **78**, 1190–1198.

Qin, F., Luo, Y., Olsen, K.B., Cai, W. & Schuster, G.T. (1992) Finite-difference solution of the eikonal equation along expanding wavefronts. *Geophysics*, **57**, 478–487. doi:10.1190/1.1443263

Rawlinson, N. & Sambridge, M. (2004) Wave front evolution in strongly heterogeneous layered media using the fast marching method. *Geophys. J. Int.*, **156**, 631–647. doi:10.1111/j.1365-246X.2004.02153.x

Ruppert, J. (1995) A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *J. Algorithms*, **18**, 548–585. doi:10.1006/jagm.1995.1021

Saito, H. (1989) Traveltimes and raypaths of first arrival seismic waves: Computation method based on Huygens' principle. *1989 SEG Annu. Meet.* doi:10.1190/1.1889578

Saito, H. (1990) 3-D ray-tracing method based on Huygens' principle. *SEG Tech. Progr. Expand. Abstr. 1990*, pp. 1024–1027, Society of Exploration Geophysicists. doi:10.1190/1.1889897

Schneider, W.A., Ranzinger, K.A., Balch, A.H. & Kruse, C. (1992) A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities. *Geophysics*, **57**, 39–50. doi:10.1190/1.1443187

Sethian, J.A. (1996) A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci.*, **93**, 1591–1595. doi:10.1073/pnas.93.4.1591

Sethian, J.A. (2001) Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. *J. Comput. Phys.*, **169**, 503–555. doi:10.1006/jcph.2000.6657

Shirley, P. & Chiu, K. (1997) A Low Distortion Map Between Disk and Square. *J. Graph.*

Tools, **2**, 45–52. doi:10.1080/10867651.1997.10487479

Si, H. (2015) TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Softw.*, **41**, 1–36. doi:10.1145/2629697

Snieder, R. & Lomax, A. (1996) Wavefield smoothing and the effect of rough velocity perturbations on arrival times and amplitudes. *Geophys. J. Int.*, **125**, 796–812. doi:10.1111/j.1365-246X.1996.tb06024.x

Spetzler, J. & Snieder, R. (2004) The Fresnel volume and transmitted waves. *Geophysics*, **69**, 653–663. doi:10.1190/1.1759451

Steyvers, M. & Tenenbaum, J.B. (2005) The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cogn. Sci.*, **29**, 41–78. doi:10.1207/s15516709cog2901_3

Thacker, W.C. (1980) A brief review of techniques for generating irregular computational grids. *Int. J. Numer. Methods Eng.*, **15**, 1335–1341. doi:10.1002/nme.1620150906

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. Solid Earth*, **99**, 24135–24157. doi:10.1029/94JB01942

Treister, E. & Haber, E. (2016) A fast marching algorithm for the factored eikonal equation. *J. Comput. Phys.*, **324**, 210–225. doi:10.1016/j.jcp.2016.08.012Tsai, Y.-H.R., Cheng, L.-T., Osher, S. & Zhao, H.-K. (2003) Fast Sweeping Algorithms for a Class of Hamilton--Jacobi Equations. *SIAM J. Numer. Anal.*, **41**, 673–694. doi:10.1137/S0036142901396533

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

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

Vidale, J.E. (1990) Finite-difference calculation of traveltimes in three dimensions. *Geophysics*, **55**, 521–526. doi:10.1190/1.1442863

Vidale, J.E. & Houston, H. (1990) Rapid calculation of seismic amplitudes. *Geophysics*, **55**, 1504–1507. doi:10.1190/1.1442798

Virieux, J. & Farra, V. (1991) Ray tracing in 3-D complex isotropic media: An analysis of the problem. *Geophysics*, **56**, 2057–2069. doi:10.1190/1.1443018

Zhang, J. & Toksöz, M.N. (1998) Nonlinear refraction traveltime tomography. *Geophysics*, **63**, 1726–1737. doi:10.1190/1.1444468

Zhao, H. (2004) A fast sweeping method for Eikonal equations. *Math. Comput.*, **74**, 603–628. doi:10.1090/S0025-5718-04-01678-3