

A hybrid approach to compute seismic travel times in 3D tetrahedral meshes

Maher Nasr<sup>1,3</sup>, Bernard Giroux<sup>1</sup> and J. Christian Dupuis<sup>2</sup>

1-Institut national de la recherche scientifique, Centre Eau Terre Environnement, 490 rue de la Couronne, Québec (Québec) Canada G1K 9A9

2- Université Laval, Département de Géologie et de Génie Géologie.

3- maher.nasr@ete.inrs.ca ; T: +1 418 654 4677 (ext 4481)

**Abstract:**

We propose an optimized method to compute travel times for seismic inversion problems. It is a hybrid method combining several approaches to deal with travel time computation accuracy in unstructured meshes based on tetrahedral elementary cells. As in the linear travel time interpolation method, the proposed approach computes travel times using seismic ray paths. The method operates in two sequential steps: At a first stage, travel times are computed for all nodes of the mesh using a modified version of the shortest path method. The difference with the standard version is that additional secondary nodes (called tertiary nodes) are added temporarily around seismic sources in order to improve accuracy with a reasonable increase in computational cost. During the second step, the steepest travel time gradient method is used to trace back ray paths for each source-receiver pair. Travel times at each receiver are then recomputed using slowness values at the intersection points between the ray path and the traversed cells. A number of numerical tests with an array of different velocity models, mesh resolutions and mesh topologies have been carried out. These tests showed that an average relative error in the order of 0.1% can be achieved at a computational cost that is suitable for travel time inversion.

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the [Version of Record](#). Please cite this article as [doi: 10.1111/1365-2478.12930](https://doi.org/10.1111/1365-2478.12930).

This article is protected by copyright. All rights reserved.

**Keywords:** Travel time- Seismic modelling- Ray-tracing.

## Introduction

Numerical wave modelling based on ray tracing is often considered as a good approximate simulation of wave propagation (Grunberg, Genaud and Mongenet 2004; Cerveny 2005). The applications range from migration and seismic tomography in exploration geophysics (Symes *et al.* 1994; Yoon *et al.* 2003; Cerveny 2005; Majdański *et al.* 2006; Göktürkler and Balkaya 2010) to earth structure determination and earthquake hypocentre localization in seismology (Zhang and Thurber 2003; Trampert and van der Hilst 2005; Doser 2006). In addition to tracing ray-paths, computing first-arrival travel times is equally important for most applications. The numerical evaluation of arrival times can be achieved through different modes of domain discretization and ray tracing methods. Considering grid-based methods, the first step consists in defining a 2D or 3D velocity or slowness model that is subsequently discretized. Conventionally, a ray-tracing method is then applied to propagate the wave front from the seismic source to all domain nodes giving the travel time value at each one. In order to calculate seismic travel times as efficiently as possible, many approaches have been proposed and improved over time. Despite a plethora of solutions and philosophies for solving this problem, a careful study allows to identify some common threads. For instance, in order to compute travel times, whatever method is used, two components are required: (1) an efficient strategy to visit the domain nodes and (2) an update mechanism to compute travel times at the cell scale. Independent from mesh type (regular or unstructured), the first component aims to conserve causality during wave front propagation. For instance, the Shortest Path Method (SPM) introduced by Nakanishi and Yamaguchi (1986) and the Fast Marching Method (FMM) implemented by Sethian (1999 a), both rely on a strategy derived from Dijkstra's algorithm (Dijkstra 1959). During wave front propagation, domain nodes are maintained into three complementary and

disjoint sets: the upwind set, the downwind set and the narrow band (Moser 1991; Sethian 1999 a).

The upwind set contains nodes where travel time values are already calculated and assigned while the downwind set includes non-visited nodes and the narrow band represents nodes directly connected to the upwind node set. At each iteration, causality is ensured by considering only narrow band nodes with the smallest travel time ( $T_{min_i}$ ) as the new upwind nodes to update neighbouring vertices. In order to avoid costly searches, narrow band nodes should be sorted according to their travel time values and kept in that sorted-order.

The Group Marching Method (GMM), proposed by Kim (2001), uses a similar strategy but offers an improvement by considering many active sources at the same time. In order to ensure causality, the updated group of nodes should be selected such that at every iteration, narrow band nodes with travel times less than  $T_{min_i} + \frac{h \cdot s_{min}}{\sqrt{3}}$  are placed in the upwind set. Note that  $h$  is the spatial step and  $s_{min}$  is the narrow node minimum slowness. Neighbouring nodes are then updated and displaced to the narrow band set (Kim 2001).

The Fast Sweeping Method (FSM), introduced by Zhao (2004), avoids the requirement to maintain a sorted list of nodes which can be time consuming and resource intensive. The method relies on Gauss-Seidel iterations to propagate the wave front. At each iteration, all the domain nodes are visited and convergence is reached for nodes along characteristic curves parallel to sweeping directions (Zhao 2004; Qian, Zhang and Zhao 2007). The causality is ensured by using several Gauss-Seidel iterations with different directions so that all characteristic curves are scanned.

Jeong and Whitaker (2008) proposed the Fast-Iterative Method (FIM) an easily parallelizable ray-racing method. In this approach, all the narrow band nodes, named here active list nodes, are updated at each iteration. Convergence is assumed to be reached for unchanged travel time nodes.

Accepted Article

Thereby, they are removed from the active list and their neighbouring nodes are added to active list. The task is repeated until the active list is empty (Jeong and Whitaker 2008). To preserve causality, all the neighbours of the converged nodes are displaced to active list even those already converged at previous iterations (Capozzoli *et al.* 2014).

Shifting to travel time update mechanisms at cell scale, three different schemes have been described. The first, often associated with the SPM, is based on Huygens's principle (Leidenfrost *et al.* 1999). It considers a transmitted wave through cell interfaces so that the node update is done by taking the neighbour vertex with the smallest travel time and increasing it by the time needed for the wave front to propagate between the two nodes (Moser 1991; Leidenfrost *et al.* 1999; Bai, Li and Tang 2011). This mechanism is recognized as the diffraction mode propagation (Podvin and Lecomte 1991; Lelièvre, Farquharson and Hurich 2011). For accuracy purposes, secondary nodes can be added on cell edges and faces to take into account the transmission of the wave through the cell (Gruber and Greenhalgh 1998). The second mechanism, called the Godunov upwind scheme, is derived from a finite-differences solution of the Eikonal equation (Sethian 1999 a; Zhao 2004; Capozzoli *et al.* 2014). This scheme is only suitable for regular grids. For unstructured meshes, generalized local solvers based on Fermat's principle have been proposed by several authors (Fomel 1997; Sethian 1999 b; Qian *et al.* 2007; Huang, Zhang and Liu 2011; Lelièvre *et al.* 2011). The update mechanism consists of finding face or edge points so that the travel time to the updated node is smallest. To find such a point, some authors rely on a geometric solution (Sethian 1999 b; Qian *et al.* 2007; Fu, Kirby and Whitaker 2013) while others consider an optimization problem solution based on spatial derivatives and linear travel time interpolation (Fomel 1997; Lelièvre *et al.* 2011; Zhang *et al.* 2011). Asakawa and Kawanaka (1993) used the latter type of solver to trace back ray paths from receivers to the corresponding seismic source. This allowed to compute the travel times along the

ray. This method, known as the Linear Travel time Interpolation Method (LTIM), can be implemented in both regular grid and unstructured meshes (Cardarelli and Cerreto 2002; Huang *et al.* 2011; Zhang *et al.* 2011).

In general, the accuracy and performance of all these methods depend on mesh resolution, slowness model, cell shape and neighbourhood size (Hysing and Turek 2005; Jeong and Whitaker 2008; Giroux 2014). The choice of the wave front propagation strategy has the greatest incidence on computing cost while the update mechanism has the most influence on the accuracy of the results. As such, many hybrid approaches have been proposed to enhance travel time computation efficiency, such as the combination of the FMM, the GMM or the SPM with the LTIM (Zhang, Chen and Xu 2004; Huang *et al.* 2011; Zhang *et al.* 2011).

Seismic travel time computations underpin inversion problems and as such authors often tend to increase mesh resolution in order to get the highest accuracy possible when computing the forward problem (Bai *et al.* 2018). The task may be expensive for nonlinear problems like velocity inversion and hypocentre location since many iterations of a quadratic ( $O(n^2)$ ) to quasilinear ( $O(n \log(n))$ ) complexity are required to reach convergence. Repeatedly calculating travel times using high resolution meshes incurs long computation times and affects the overall inversion performance (Bai *et al.* 2018). This motivates the development of algorithms that provide efficient travel time computation with high accuracy and at reasonable computational cost. Such an approach is the aim of the present work. Based on the results of a large number of tests, we found that the proposed hybrid approach provides the sought trade-off between accuracy and time cost. Similar to LTIM, the presented approach relies on the computation of travel times along ray paths and combines these with other methods such as SPM and FMM. Briefly, this approach operates in two stages: (1) travel times are calculated for all domain nodes using the SPM or the FMM and (2) the ray paths joining

Accepted Article

receivers to corresponding seismic source are traced back and travel times are recalculated along the seismic rays. When ray paths are not explicitly needed, the second step can be avoided by interpolating travel times stored at grid nodes, at the receiver locations. Furthermore, in the case of the SPM, ray paths can be deduced directly at the first step, by storing in memory, for each node, the index of the source node yielding the final value of travel time (the so-called “parent” node). This procedure, however, results in ray path segments that necessarily pass at nodes, yielding somewhat discontinuous ray trajectories. Figure 1 illustrates this effect for a case in 2D for a homogeneous velocity model. Increasing the number of secondary nodes reduces this effect, at the expense of increased computational cost. There is therefore a balance to find between avoiding the second step, which reduces computation time, and increasing the number of secondary nodes.

In this paper, the two steps, i.e. travel time computation and ray path determination, are analysed for unstructured meshes and different options that can be included at each stage are tested to enhance travel time computation accuracy and computational cost. Several experiments have been realized in order to find the most efficient combination between different approaches available at each stage. Smooth velocity models similar to those derived from inversion processes are used. In addition to hybrid combinations, the standard LTIM version is implemented for real tetrahedral meshes. To our knowledge, this is the first time that the LTIM is tested with that kind of unstructured meshes. Its performance is evaluated and compared to hybrid and standard approaches. Accuracy and computational cost for the optimal combination are then evaluated with different complexity slowness models and compared to two classic methods: the SPM and the LTIM.

## Conceptual approach

The main concept behind the proposed strategy consists in calculating travel times along ray paths. This technique was introduced by Asakawa and Kawanaka (1993) as a part of LTIM on regular 2D grids. Since then, several researchers have looked at ways to enhance the accuracy and the performance of the method (Li and Ulrych 1993; Cardarelli and Cerreto 2002; Vanelle and Gajewski 2002; Zhou, Zhang and Chen 2004; Zhang *et al.* 2015; Li, Liu and Zhang 2019). The method was extended to 3D regular grids by Zhang *et al.* (2011) and to unstructured mesh by Huang *et al.* (2011). In their work, Huang *et al.* (2011) relied on a hybrid mesh to discretize the study domain: a cubic elementary cell is the initial kernel and the cell shape has been slightly perturbed to fit undulating interfaces. Implementing the LTIM on unstructured meshes based on tetrahedral elements is trickier compared to a cubic grid due to the triangular shape of cell faces. As a first hurdle, local solvers based on linear travel time interpolation are less accurate because only three nodes are available instead of four. Thus, while the bilinear interpolation in a quadrilateral face is equivalent to a 4-term polynomial with a linear travel time gradient:

$$T(x, y) = a_0 + a_1x + a_2y + a_3xy, \quad (1)$$

the travel time for a triangular face is represented with only a 3-term polynomial and a constant travel time gradient:

$$T(x, y) = a_0 + a_1x + a_2y. \quad (2)$$

The second hurdle is that causality is not always respected in the case of obtuse angle triangles (Qian *et al.* 2007; Lelièvre *et al.* 2011; Fu *et al.* 2013) and can generate an additional error. The standard LTIM therefore appears to be theoretically less accurate and stable when used with tetrahedral meshes. Given these limitations, LTIM should be combined with other methods to enhance its performance.

The strategy that is proposed still requires two sequential steps, similar to current LTIM implementations. Consider a tetrahedral mesh in which slowness is defined at the vertices of the tetrahedra, i.e. at the so-called primary nodes. During the first step that is commonly known as the

forward process, travel times are computed for all mesh nodes using either SPM or FMM. The choice of method depends on the accuracy that can be reached for the given problem. The second step, often called the backward process, consists in tracing the ray paths for all source-receiver pairs. Travel times ( $T$ ) can be then recalculated, during this second step, along the seismic rays (Huang *et al.* 2011).

$$T(x, y, z) = \sum_{i=1}^{n-1} \frac{1}{2} (s_{i+1} + s_i) \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}, \quad (3)$$

where  $s_i$ ,  $x_i$ ,  $y_i$ ,  $z_i$  and  $(n-1)$  are respectively the slowness, the coordinates and the number of segments constituting the ray joining the source node to the receiver, while  $x$ ,  $y$  and  $z$  are the receiver coordinates. The segments are defined by the intersection of the ray path with the faces of the traversed tetrahedra, and slowness values at intersection points on the interfaces ( $s_i$ ) are interpolated from the values stored at the primary nodes (Figure 1). This second stage is referred to be the backward process because computation is done going from the receiver back to the source (Asakawa and Kawanaka 1993).

### Forward step

During the forward step, wave fronts are propagated from seismic source to the whole domain so that travel times can be computed for all nodes. In theory, this step can be considered as a classic travel time computation problem. Any method previously discussed such as the SPM, the GMM or the FMM should work for this stage. One should note however that the update mechanism can impact the computational accuracy of the solution. To illustrate this effect, we propose to compare the accuracy of the SPM update mechanism (Giroux 2014) to the accuracy of the LTI local solver (Lelièvre *et al.* 2011) used in the FMM. Solutions based on Godunov schemes that are hinged on finite-differences solution of Eikonal equation were discarded since they are inappropriate for unstructured meshes as shown by Fomel (1997) and Sethian (1999b). These tests consist of

calculating travel times for all primary nodes of the domain using the SPM and the FMM. A model with a vertical velocity gradient of  $v(z) = 1.5 + 4.5z$  was considered with four different mesh resolutions. The study domain is cubic with an edge of 1 km. The receivers, at a number of 1000, were randomly placed within the domain while seismic source was placed at the centre. All the meshes that were used during these tests were prepared using Gmsh version 4, an open-source software originally proposed by Geuzaine and Remacle (2009). Following the approach of Gruber and Greenhalgh (1998), the SPM was used in this example with three secondary nodes on each cell edge and three on each cell face. The distribution of these secondary nodes inside cell face is illustrated below, in Figure 2. The absolute and relative errors are then calculated for each primary node and the travel times computed for each method were compared to the analytic solution given by Cerveny (2005) and Fomel, Luo and Zhao (2009). The results obtained, shown in Figure 3, were examined to determine the most suitable update mechanism for tetrahedral meshes. These results suggest that, in terms of accuracy, SPM outperforms FMM regardless of the number of primary nodes and hence can be considered as the most accurate method in this case. On the other hand, the SPM computational cost is somewhat higher than for the FMM, even when using few secondary nodes like in the present example (3 secondary nodes). We propose below a modification to the SPM to reduce its computational cost and preserve accuracy.

The conclusions regarding accuracy of the methods have previously been reached by Bai, Huang and Zhao (2010) and Giroux (2014) who showed that SPM provides more accurate solutions than FMM based on the LTI solver. This emphasizes the fact that the LTI local solver appears to be less suitable for 3D unstructured meshes, especially those based on tetrahedral elementary cells, because the number and the configuration of nodes make travel time computations less accurate in comparison to a regular grid. Recall that a good approximation of the Eikonal equation solution at cell scale requires at least two nodes at each quadrant (Gavete, Gavete and Benito 2003). Furthermore, cells of domain presenting triangular faces with an obtuse angle could affect the accuracy of LTI solver and cause additional errors as previously discussed (Qian *et al.* 2007; Lelièvre *et al.* 2011). The generated errors at the vicinities of such cells can propagate to other

nodes along characteristic curves and induce more travel time computation errors sometimes further at other cells. This is a common occurrence for real unstructured meshes created by most generators such as Gmsh or Tetgen, even with optimization option turned on. In contrast, the SPM updater does not show any causality problems or cell geometry dependency. This also holds when secondary nodes are used and it is considered to be efficient (Giroux 2014). A common parametrization strategy is to associate slowness values at the nodes of the mesh instead of within the cells. A velocity model is then interpolated at secondary nodes which allows to take slowness variations into account and thus provide a better approximation of the seismic ray shape inside the cell. This strategy that uses secondary nodes is also useful to enhance the performance of the inversion algorithm. In order to have a reasonable number of model parameters, the mesh for inversion problems are usually coarser in comparison to forward problems. In the forward case, it is important to increase the mesh resolution in order to get better travel time accuracy. Commonly, two meshes must be defined and stored separately in order to achieve reasonable accuracy and acceptable numerical performance (Serretti and Morelli 2011; Penz 2012). This is the case for the majority of raytracing methods such as FMM, GMM and FSM. Such a double meshing can be avoided, however, when SPM is used. This is because the inversion problem can be cast such that it works only with the primary nodes. During the ray tracing step, the mesh is refined by adding secondary nodes to the list of primary nodes. Consequently, this makes SPM a suitable choice for inversion algorithms since it saves memory and reduces computation cost: less nodes are stored and less slowness interpolations are done (Table 1).

At cell scale, the LTI local solver assumes homogeneous slowness values which therefore implies a straight ray paths within each cell. This assumption induces further error contrary to the SPM solver which insures a better approximation of seismic ray shape via the addition of secondary nodes. The LTI local solver also has another important weakness in that it violates the planar wave hypothesis near seismic sources. This results in high travel time errors in the vicinity of the sources. The common workaround for LTI is to refine the mesh around the source nodes. This is possible on most mesh generators. However, in the case of nonlinear inversion problems, source location may change from iteration to another and therefore the whole domain should be discretized anew at every iteration to get the refinement at the source positions. It soon becomes apparent that this approach is cumbersome to implement and can be computationally intensive. In comparison, SPM allows for an efficient alternative that we propose in this paper. The idea is to create a mesh of homogeneous resolution and preserve its topology for all iterations, and perform mesh refining at each iteration using temporary nodes placed in the vicinity of the source nodes. These dynamic nodes are added on cell edges and faces with the approach of Gruber and Greenhalgh (1998) and are removed once travel times are computed for the corresponding source. Three

node types are thus defined for SPM as illustrated in Figure 2: (1) primary nodes, located at the tetrahedron vertices, that define the inversion model parameters, (2) secondary nodes added on the cell edges and faces to increase mesh resolution for the whole domain and (3) temporary tertiary nodes placed within a given radius of the seismic sources to refine the mesh locally. Inversions are carried out considering only primary nodes, while the ray tracing step operates with all nodes. The slowness model is defined at primary nodes and interpolated at both secondary and tertiary node sets. The total number of nodes within a given cell is given by:

$$N = N_p + N_s + N_T, \quad (4)$$

where  $N_p$  is equal to 4, the number of primary nodes.  $N_s$  and  $N_T$  are respectively the number of secondary and tertiary nodes.  $N_s$  and  $N_T$  are calculated as follow:

$$N_s = 2ns \cdot (ns + 2), \quad (5)$$

$$N_T = 2nt \cdot (nt + 2) \cdot (ns + 1)^2, \quad (6)$$

where  $ns$  represents the number of secondary nodes added on each cell edge and  $nt$  is the number of tertiary nodes added between two successive secondary nodes. While the number of primary and secondary nodes is always invariant, the number of tertiary nodes may change during the inversion processes. The total number of tertiary nodes  $N_T$  depends on the local mesh resolution, the source position, and the radius around the source where these nodes are added. This makes the whole mesh resolution dynamic. For this reason, we call this modified SPM the dynamic shortest-path method (DSPM). In further sections, DSPM\_ $s_t$  refers to the DSPM with  $s$  secondary nodes and  $t$  tertiary nodes (Figure 2).

**-The DSPM vs the SPM:** As stated previously, we use accuracy and computational time as metrics to compare DSPM and SPM. As such, the relative errors at primary nodes as a function of distance from seismic source are illustrated in Figure 4 for three different levels of discretization. Each domain in this example contains 11,026 primary nodes. The velocity model was derived using a vertical gradient model ( $v(z) = 1.5 + 4.5z$ ). The domain under study is still a  $1 \text{ km}^3$  cube with average cell edges length of approximately 57 m. The receivers and seismic source were kept at the same

positions as the example shown in Figure 3. Three different schemes are tested, with parameters given in Table 2. In the first discretisation, 9 secondary nodes are placed on every edge and 36 inside each face (SPM\_9). For the second discretization, 7 secondary edge nodes and 21 face nodes were added to every edge and face respectively (SPM\_7). The last discretization considered has one tertiary node between edge nodes (DSPM\_7\_1), for cells distributed uniformly around the seismic source in a sphere with a radius of 150 m. This scheme increased the overall number of nodes by 4% in comparison to SPM\_7 bringing total number of nodes close to  $3.0 \times 10^6$ , a reduction of 37% relative to the overall number of nodes in SPM\_9.

The modelling results, shown in Figure 4, reveal that the accuracy of SPM is improved when tertiary nodes are introduced. These results also show that despite a significant reduction in the number of nodes, the results from DSPM\_7\_1 have lower relative error than the more resource intensive SPM\_9 discretization. In fact, the maximum recorded relative error decreased from 3.4 % and 3.5% respectively when homogeneous mesh resolution models SPM\_9 and SPM\_7 were considered, to 1.6% when higher mesh resolution was used around seismic sources (DSPM\_7\_1). It is interesting to note that the improvements are not restricted to the regions proximal to the nodes that are coincident with the source points but also at greater distances in the model. This is can be explained by the fact that error at nodes close to the source often propagates over all domain nodes: travel times calculated at a far location from source depend on the travel time of near nodes. The global average error was evaluated to 0.66% and 0.86% for SPM\_9 and SPM\_7 respectively and to 0.62% after applying the mesh-refinement process of DSPM\_7\_1. The strategy of increasing mesh resolution locally represents an interesting alternative to deal with the planar wave approximation problem: the DSPM ensures more accuracy with a reasonable mesh resolution. Avoiding mesh refinement of the whole domain can enhance the performance of the method by saving memory

and reducing computation costs while still maintaining a reasonable relative error. As an example, the computation time was decreased by 45 % when SPM\_7 was used instead of SPM\_9 and by 50% when DSPM\_7\_1 was used instead of SPM\_9. It is interesting to note that, although the total number of nodes is 7% higher for DSPM\_7\_1 relative to SPM\_7, the computation time for the latter is 8% higher. This is explained by differences in implementation that relates to the information stored for each node. For SPM, the index of the parent node is saved to allow raytracing during the forward step. This incurs additional memory access during computations and therefore slows down the solution. For the DSPM, the raytracing is performed at a backward step (as detailed in section 2.2 below) and thus the forward step is not affected by the same memory access overhead.

**-Tertiary node Subdomain:** The extent of the subdomain around sources where tertiary nodes are used plays a crucial role for the DSPM accuracy. In this work, the subdomain is assumed to be spherical with a radius  $R$  and centred on the seismic source. Since many factors influence its choice, the criteria to set the radius are examined in this section. These factors are the initial mesh resolution and quality, the number of secondary and tertiary nodes and the used velocity model. The radius has to be set in order to increase mesh resolution in the source vicinity, where the wave front curvature is more pronounced in comparison to the size of the cells. Recall that the SPM is based on Huygens's principle, which states that every point on a wave front is itself the source of a spherical wave, and these waves which sum up to propagate the wave front in space. By increasing the density of nodes close to the source, the number of potential secondary source is increased and a highly curved wave front will be better represented.

To illustrate the influence of the choice of the radius  $R$ , we consider the previous example with a mesh of about 11 thousand nodes. The mesh has a relatively narrow cell size distribution, with mean cell edge length  $\langle e_i \rangle$  of  $57 \pm 14$  m. Tests are performed for five increasing values of radius  $R$  normalized

by  $\langle e_i \rangle$ , for six discretization schemes: DSPM\_1\_1, DSPM\_1\_2, DSPM\_2\_1, DSPM\_2\_2, DSPM\_3\_1, and DSPM\_3\_2. The results are presented in Figure 5, which shows the median of the relative error computed for bins of increasing distance to the source, as well as computational cost. For comparison, the results for SPM\_5 are also presented. We can observe that the effect of radius  $R$  on accuracy remains proportionally constant as the number of secondary nodes  $n_s$  increases. On the other hand, the effect of  $R$  is more pronounced if the number of tertiary nodes is increased. In most cases, the benefit of using a radius larger than  $3\langle e_i \rangle$  appears marginal, especially considering that the computation cost increases substantially for  $R > 3\langle e_i \rangle$ . A second series of tests were performed with a finer mesh (results not shown), and the same trends were observed.

### Backward process

After calculating the travel times for all domain nodes, a backward step is started. The objective of this step is to get the ray paths connecting each receiver to the corresponding source (Asakawa and Kawanaka 1993; Zhou *et al.* 2004; Huang *et al.* 2011; Zhang *et al.* 2011). The ray is approximated by using a finite number of line segments that are defined by points of intersection between the maximum travel time gradient and mesh cells. The methods used to trace back seismic rays fall into one of two camps: the first relies on the LTI local solver to trace the ray from the receiver back to the seismic source (Asakawa and Kawanaka 1993; Huang *et al.* 2011; Zhang *et al.* 2011) while the second is based on a local calculation of the steepest travel time gradients (Zhang *et al.* 2013; Ding *et al.* 2016). Accuracy and performance are compared for both approaches after the brief introduction that follows below.

### The LTI local solver

LTI updaters were already introduced for solving the forward step problem. At any given receiver (R), the solver provides the point on the cell face (P) that minimizes travel time. The line joining R to P is considered as a segment of the ray path and P is then assumed to be a new receiver. The task is

repeated until the source node is reached. To calculate the coordinates of P, we follow the approach presented by Lelièvre *et al.* (2011). For this, we suppose that a point receiver R is located inside a tetrahedron cell (ABCD). We then assume that P1, P2, P3 and P4 are respectively the points that minimize travel times on face ABC, ABD, BCD and ACD (Figure 6). According to Lelièvre *et al.* (2011), the position of P<sub>1</sub> is given by its relative coordinates ( $\xi$ ,  $\zeta$ ) in the base (A, **b**, **c**):

$$\mathbf{AP}_1 = \xi \cdot \mathbf{c} + \zeta \cdot \mathbf{b}, \quad (7)$$

where  $\xi$  and  $\zeta$  are calculated as following:

$$\xi = \xi_0 + \frac{|b^2(T_B - T_A) - (T_C - T_A) \mathbf{b}^t \cdot \mathbf{c}|}{\varphi w} \rho_0, \quad (8 \text{ a})$$

$$\zeta = \zeta_0 + \frac{|c^2(T_C - T_A) - (T_B - T_A) \mathbf{b}^t \cdot \mathbf{c}|}{\varphi w} \rho_0. \quad (8 \text{ b})$$

Here  $\xi_0$  and  $\zeta_0$  are the relative coordinates of R' the orthogonal projection of R onto face ABC and  $\rho_0$  is the distance from the point R to R' (Figure 6). The norm of cross product between vectors **b** and **c** is given by  $\varphi$  ( $\varphi = || \mathbf{b} \times \mathbf{c} ||$ ). The transpose operator is  $t$  while  $T_A$ ,  $T_B$  and  $T_C$  are respectively the travel times at nodes A, B and C computed during the forward step. The 3 nodes are labelled such that the travel time at node A is the smallest ( $T_A < T_B$  and  $T_A < T_C$ ).

The difference between cell slowness ( $s$ ) and travel time gradient is denoted as  $w$  (Lelièvre *et al.* 2011), i.e.

$$w^2 = s^2 \varphi^2 - (T_B - T_A)^2 b^2 - (T_C - T_A)^2 c^2 + 2(T_B - T_A)(T_C - T_A) \mathbf{b}^t \cdot \mathbf{c}. \quad (9)$$

Note that the slowness value is assumed to be constant inside the cell. Obviously,  $w^2$  should be positive to calculate  $\xi$  and  $\zeta$ . Moreover, the point P<sub>1</sub> has to be inside the face ABC. The condition is fulfilled by testing the following expressions:

$$0 < \xi < 1, \quad (10 \text{ a})$$

$$0 < \zeta < 1, \quad (10 \text{ b})$$

$$0 < \xi + \zeta < 1. \quad (10 \text{ c})$$

In addition to the transmitted waves inside cells, head waves travelling along the edges should be considered (Sethian 1999 a; Qian *et al.* 2007; Lelièvre *et al.* 2011). For this task we need to determine points that minimize travel time onto edges AB, AC and BC. The pseudo-code that meets this requirement is shown in Algorithm 1 below. To investigate how this algorithm works, we start by considering the case for the edge AB where the position of the point sought is given by the normalized distance  $\xi$  from node A such that:

$$\mathbf{AP}_1 = \xi \cdot \mathbf{c}. \quad (11)$$

The value of  $\xi$  that minimize travel time along this edge is calculated as shown below:

$$\xi = \xi_0 + \frac{(T_B - T_A)\rho_0}{\tilde{w}c}. \quad (12)$$

The task is similar for the 3D case, where  $\rho_0$  is defined as the distance to R from its orthogonal projection  $R''$  onto line AB (Figure 6).  $\xi_0$  is the normalized distance between A et  $R''$  while  $\tilde{w}$  is computed as:

$$\tilde{w}^2 = s^2c^2 - (T_B - T_A)^2. \quad (13)$$

When  $\tilde{w}^2$  is negative or P falls outside the line segment AB ( $\xi > 1$  or  $\xi < 0$ ), the solution is given by the node that minimizes the head wave travel time along the edges AR and BR ( $\min(T_A + s_{AR}, T_B + s_{BR})$ ).

For each face around the receiver, 7 points have to be computed such that travel time values are minimized for transmitted and head waves. These points fall onto edges and faces of the cells surrounding receiver R (Lelièvre *et al.* 2011; Algorithm 1; Algorithm 2). Considering face ABC for example, the first point, if it exists, is due to the transmitted wave traveling inside the volume RABC computed using equation 8a and 8b. The second corresponds to head waves travelling inside faces RAB, RAC and RBC and along edges RA, RB and RC. The only point that minimizes travel times to the

receiver is preserved while the others are discarded. The task is repeated for all faces around the receiver (faces ABD, BCD and ACD in Figure 6).

Finally, for the  $n$  faces around the receiver, point  $P$  is chosen such that it has the smallest travel time to the receiver (Algorithm 2):

$$P_i | \min(T_{P_i} + s_i RP_i). \quad (14)$$

Depending on the receiver position, the number of faces  $n$  can take different values. Thus, for a receiver falling inside a cell, the four faces that belong to the cell have to be considered. The number of faces to consider increases to 6 when the receiver is located on a face, between two cells. It is also possible to get yet a greater number of faces when the receiver is located on a cell vertex or an edge. Faces belonging to cells sharing the vertex or the edge have to be tested. The overall number for each receiver position will therefore depend strongly on the local mesh topology.

#### The Steepest travel time gradient methods

According to Fermat's principle, the steepest travel time gradient is always normal to the wave front and tangent to the seismic ray at a local scale (Zhang *et al.* 2013; Wang *et al.* 2014; Ding *et al.* 2016).

Thus, ray paths can be traced following the travel time gradient. The negative direction of the gradient should be taken since the ray is traced back from receivers to the source. Similar to the LTI solver approach, the method operates iteratively to find the intersections between ray paths and mesh cells. The ray is traced by connecting the segments defined by travel time gradient intersections with the cell faces.

Many methods have been proposed to approximate travel time gradient on unstructured meshes. Some approaches rely on a first order linear gradient reconstruction such as the Averaging-Based method (ABM), the Regression-Based method (RBM), the Green-Gauss method and the Raycasting

(ter Braak and Juggins 1993; Ledergerber *et al.* 2008; Correa, Hero and Ma 2011). For such methods, travel times are approximated at each point  $\mathbf{x}_0(x_0^1, x_0^2, x_0^3)^t$  using a first order Taylor expansion:

$$T(\mathbf{x}_0 + \mathbf{h}) \approx T(\mathbf{x}_0) + \mathbf{h}^t \cdot \nabla T(\mathbf{x}_0), \quad (15)$$

where  $\nabla T(\mathbf{x}_0)$  is the travel time gradient and  $\mathbf{h}$  is the discretization step. Some approaches can be extended to higher order gradient reconstruction, such as the Regression-Based method (RBM). A second order approximation of the travel times is used in this case:

$$T(\mathbf{x}_0 + \mathbf{h}) \approx T(\mathbf{x}_0) + \mathbf{h}^t \cdot \nabla T(\mathbf{x}_0) + \frac{1}{2} \mathbf{h}^t \cdot \mathbf{H}(\mathbf{x}_0) \cdot \mathbf{h}, \quad (16)$$

where  $\mathbf{H}(\mathbf{x}_0)$  is the travel time Hessian matrix.

Three methods were tested in this study. The first is the ABM. The second is the RBM with a first order approximation of travel time gradient and the last one is the second order version of the RBM.

**-The Averaging-Based method:** For this method, the travel time gradient is calculated at a given cell vertex as a weighted linear combination of gradients computed at each neighbouring cell (Correa *et al.* 2011). The gradient at point  $\mathbf{x}_0$  is given by

$$\nabla T(\mathbf{x}_0) = \sum_{i=1}^{nc} f_i \nabla T(i), \quad (17)$$

where  $\nabla T(i)$ ,  $f_i$  and  $nc$  are respectively the travel time gradient, the cell weighting factor and the number of neighbouring cells. The travel time gradient  $\nabla T(i)$  inside each cell  $i$  is assumed constant and is computed by solving (Correa *et al.* 2011).

$$\begin{bmatrix} (\mathbf{x}_1 - \mathbf{x}_0)^t \\ (\mathbf{x}_2 - \mathbf{x}_0)^t \\ (\mathbf{x}_3 - \mathbf{x}_0)^t \end{bmatrix} \cdot \nabla T = \begin{bmatrix} T(\mathbf{x}_1) - T(\mathbf{x}_0) \\ T(\mathbf{x}_2) - T(\mathbf{x}_0) \\ T(\mathbf{x}_3) - T(\mathbf{x}_0) \end{bmatrix}, \quad (18)$$

where  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$  are the other three vertices of the cell  $i$ . The left-hand side is a  $3 \times 3$  matrix with full rank in the case of non-degenerated tetrahedron. Thereby, the system can be solved exactly using direct methods.

Different representations were proposed to express weighting factors appearing in equation 17.

- 1- A uniform function: in this case, all cells have the same weights. The method is recommended for meshes showing homogeneous cell size.
- 2- A volumetric function: each cell is weighted using its relative volume. The method is equivalent to the Green-Gauss Method (Correa *et al.* 2011).
- 3- An inverse centroid distance function: the travel time gradient at each cell is weighted using inverse distance from vertex  $\mathbf{x}_0$  to correspondent cell centroid.

Correa *et al.* (2011) compared different weighting functions and proved that the inverse centroid distance gives the highest accuracy compared to the other methods. Thus, the ABM with inverse centroid distance weighting was chosen for this work. This approach is only valid to calculate travel time gradient at cell vertices and thus if the receiver falls on an edge, on a face, or inside a cell, travel time gradients must be computed at each neighbouring vertex. The gradient components are then approximated at the receiver position using respectively 1D, 2D or 3D linear interpolation.

**-The Regression-Based method:** The method is also known as the Least-squares gradient reconstruction (Dahoe and Cant 2004) or the method of Moving Least-Squares (Gavete *et al.* 2003; Ledergerber *et al.* 2008). The travel time gradient computed makes use of multidimensional regressions. For  $M$  nodes surrounding the point  $\mathbf{x}_0$ , equation 15 can be written  $M$  times and gives the following linear system of equations which is written in matrix notation as:  $\mathbf{A} \cdot \nabla T(\mathbf{x}_0) = \mathbf{b}$ , where:

$$\mathbf{A} = \begin{bmatrix} (\mathbf{x}_1 - \mathbf{x}_0)^t \\ (\mathbf{x}_2 - \mathbf{x}_0)^t \\ \vdots \\ (\mathbf{x}_i - \mathbf{x}_0)^t \\ \vdots \\ (\mathbf{x}_M - \mathbf{x}_0)^t \end{bmatrix}, \quad (19 \text{ a})$$

$$\mathbf{b} = \begin{bmatrix} T(\mathbf{x}_1) - T(\mathbf{x}_0) \\ T(\mathbf{x}_2) - T(\mathbf{x}_0) \\ \vdots \\ T(\mathbf{x}_i) - T(\mathbf{x}_0) \\ \vdots \\ T(\mathbf{x}_M) - T(\mathbf{x}_0) \end{bmatrix}. \quad (19 \text{ b})$$

An  $M \times M$  diagonal matrix  $\mathbf{W}$  can be inserted in the system to give different weights to nodes. The system can be written as:

$$\mathbf{W} \cdot \mathbf{A} \cdot \nabla T(\mathbf{x}_0) = \mathbf{W} \cdot \mathbf{b}. \quad (20)$$

The weight coefficients can follow a uniform or an inverse distance function. Tests conducted by Mavriplis (2003) and Correa *et al.* (2011) have shown that better accuracy is achieved with inverse distance weighting and thus this weighting scheme was retained for this study. The linear system of equations described by equation 20 is usually overdetermined and can therefore be solved using the least-squares method.

One should note that RBM is easily extended to second order gradient reconstruction by adopting the second order Taylor expansion (the equation 16). The system of equations to be solved then becomes:

$$\begin{bmatrix} (\mathbf{x}_1 - \mathbf{x}_0)^t & vect((\mathbf{x}_1 - \mathbf{x}_0) \cdot (\mathbf{x}_1 - \mathbf{x}_0)^t) \\ (\mathbf{x}_2 - \mathbf{x}_0)^t & vect((\mathbf{x}_2 - \mathbf{x}_0) \cdot (\mathbf{x}_2 - \mathbf{x}_0)^t) \\ \vdots & \vdots \\ (\mathbf{x}_i - \mathbf{x}_0)^t & vect((\mathbf{x}_i - \mathbf{x}_0) \cdot (\mathbf{x}_i - \mathbf{x}_0)^t) \\ \vdots & \vdots \\ (\mathbf{x}_M - \mathbf{x}_0)^t & vect((\mathbf{x}_M - \mathbf{x}_0) \cdot (\mathbf{x}_M - \mathbf{x}_0)^t) \end{bmatrix} \cdot \begin{bmatrix} \nabla T(\mathbf{x}_0) \\ vect(\mathbf{H}(\mathbf{x}_0)) \end{bmatrix} = \begin{bmatrix} T(\mathbf{x}_1) - T(\mathbf{x}_0) \\ T(\mathbf{x}_2) - T(\mathbf{x}_0) \\ \vdots \\ T(\mathbf{x}_i) - T(\mathbf{x}_0) \\ \vdots \\ T(\mathbf{x}_M) - T(\mathbf{x}_0) \end{bmatrix}, \quad (21)$$

where  $vect(\cdot)$  is an operator transforming a matrix to a vector by vertically appending its columns in order (Erem and Brooks 2011). The system is solved the same way as for equation 19, using least-squares. The travel time gradient components are stored in the first 3 elements of the solution vector.

## Accuracy and performance

### Backward process

The averaging and regression-based methods as well as the LTI local solver ray-tracing were evaluated in terms of their accuracy and computing time requirements. All of the results presented were obtained using a 3 GHz 8-Core Intel Xeon machine running macOS Mojave 10.14.1 with 64 Go 1866 MHz DDR3 RAM. All algorithms have been implemented in C++ and compiled with the clang compiler version 10.0 with level 3 optimization turned on. The source code is available on github (Giroux and Nasr 2018). Although these implementations have been parallelized following a task-based approach (Giroux and Larouche 2013), parallel performance is not assessed in this paper.

The accuracy of the methods was the first metric to be investigated by using the same mesh resolution, slowness model and forward step solution (the DSPM). The only modification was the method used for the backward process. The analysed approaches include the LTI solver, the steepest gradient reconstruction using the ABM, the first order regression-based method (FO-RBM) and the second order regression base method (SO-RBM). Travel times obtained after interpolation of the forward-step travel times at receiver locations are also presented to help quantify the benefits of the different processes used for computing the backward step. For all tests in this section, the domain dimensions, the velocity model as well as the receivers and the seismic source remain the ones described in section 2.1. Thus, a vertical gradient velocity model, a point seismic source placed at the centre of the cubic domain and one thousand randomly distributed receivers were used. In this first test, the number of primary nodes is still 11,026 and one secondary node per edge was added for all domain cells. The total number of nodes was increased to 3 edge nodes and 3 face nodes (DSPM\_1\_1 according to the nomenclature presented in Figure 2) within a radius of 0.1 km around the source. This brings the total number of nodes to 82,172. Since the smoothness of this model is

comparable to velocity models that are typically obtained from inversion, this seems an appropriate choice. For this numerical experiment, the seismic source was placed at the centre of the domain.

The absolute and relative travel time errors were computed for each receiver and are shown in Figure 7 and Table 3.

Errors computed for different methods show that the LTI solver is less accurate than the steepest gradient methods (the ABM, the FO-RBM and the SO-RBM) by one to two orders of magnitude. To explain the low accuracy of the LTI solver, three hypotheses can be postulated. The first is that the method operates at the scale of one cell and uses only 4 nodes to determine ray path intersection with mesh cells, whereas the steepest gradient methods relies on a cloud of nodes 5 to 8 times greater for this mesh resolution. The number of nodes used is therefore a crucial factor to better approximate the ray path. Indeed, tests that used the FO-RBM with travel time gradient computed at the scale of one cell (only the 4 cell nodes are used) showed accuracy that, albeit slightly less, is of the same order of magnitude as was obtained with the LTI solver. This is to say that the LTI solver approach is somewhat equivalent to the steepest gradient methods operating at the scale of one cell but it gets left behind when more nodes are available.

The second hypothesis is that the LTI solver depends more on the geometry of the cells because faces with obtuse angles have a significant deleterious effect on the accuracy of the LTI solver.

Unfortunately, unlike for the forward process, there are no ingenious modifications that can be brought to bear this issue. Consequently,  $w^2$  can have a negative value and thus it is impossible to calculate a point that minimizes the travel time at this position (equation 9; Lelièvre *et al.* 2011). It is also conceivable to have a positive value of  $w^2$  that leads to a point that falls outside face borders. Under both scenarios, the ray path is forced to pass through cell vertices. This deviation increases considerably the travel time error along the ray especially with low mesh resolution. As a consequence, this means that the LTI solver accuracy depends more on cell density than the

steepest gradient methods. The final hypothesis to explain the poor performance of LTI is that the solver operates poorly near source nodes. The violation of the plane wave approximation at source vicinities is likely the main culprit.

For the steepest gradient methods, the tests showed that SO-RBM gives the highest travel time accuracy, as can be seen in Figure 7 and Figure 8. The second order approximation of travel times based on the computation of the Hessian Matrix  $\mathbf{H}(\mathbf{x}_0)$  ensures higher accuracy for the travel time gradient calculation. However, the RBM based on first order approximation (FO-RBM) seems to be less accurate by one order of magnitude compared to other methods. Indeed, with increasing neighbouring node number, matrix  $\mathbf{A}$  of equation (19 a), an overdetermined system, becomes more ill-conditioned, which implies more travel time gradient error. On the other hand, the ABM shows an absolute and a relative error with the same order of magnitude as the SO-RBM despite the fact that it calculates travel time gradient using a first order approximation. Contrary to the FO-RBM that uses the same approximation order, the ABM provides an exact solution to a series of 3 by 3 well-posed systems (equation 18) for non-degenerated neighbouring tetrahedron and does not suffer consequently from conditioning problems as does the FO-RBM.

The second evaluation metric evaluated for these backward step processes is computation time. For assessing this metric, numerical experiments with different mesh resolutions were performed. The results reveal that CPU time for all methods are closely matched as shown in Figure 8. FO-RBM remains the fastest and the SO-RBM remains the slowest for all the tests. The relative difference between the fastest and slowest method tends to decrease as mesh resolution increases. It is possible to see that the ABM has a slight advantage over the SO-RBM in terms of computation speed, but the increase in accuracy for the latter is significant. For this reason, and because computation time difference is rather small, we opt to favour the SO-RBM.

Looking at the spatial distribution of errors, there is no clear correlation with distance to the source for the backward process which is at odds with the results of the forward step performed in section 2.1. Figure 9 shows the absolute and relative errors for LTI and SO-RBM as examples. Results for the ABM and FO-RBM are comparable. There is no evidence that absolute errors are higher further away from the source or that relative errors are higher closer to the source. After close inspection it seems that higher errors are aggregated in some portions of the model. We have examined the source of this aggregation to see if there is a correlation with mesh quality or error at primary nodes, but there is no obvious cause at this stage. Nonetheless, this aggregation is likely caused by a combination of factors such as: mesh quality, accuracy of gradient computation or LTI solver, and propagation of error at primary nodes.

In another series of tests, the steepest gradient method has been compared to the standard SPM. Figure 10 shows the errors and CPU time for the DSPM\_1\_1 with SO-RBM presented above with errors and CPU time for standard SPM. Two different tests were done with SPM, the first with travel times calculated at the forward step (SPM fw, top figures) and the second with travel times calculated with the SO-RBM (SPM bw, bottom figures).

The first observation that can be made is that computing travel times during the backward step has a tremendous impact on accuracy. It effectively reduces the errors by more than one order of magnitude. This effect is most pronounced at finer mesh resolution. Comparing results for the SPM alone, this improvement in accuracy is obtained at the cost of an increase in computation time ranging between 5% and 45% (27% on average). The second observation that can be made is that the level of error with DSPM is in the same order of magnitude as the SPM with SO-RBM, albeit slightly higher. On the other hand, using the DSPM is always faster by a significant proportion.

In the final series of tests for the backward step process, we examined the influence of the number of secondary and tertiary nodes on the performance of DSPM. Three configurations were compared to the DSPM\_1\_1 (i.e. DSPM\_1\_2, DSPM\_2\_1 and DSPM\_1\_3). All of these tests used the SO-RBM. Moreover, we also used the same domain, velocity model, seismic source and receiver positions described in section 2.1. Errors and CPU times are shown in Figure 11. As expected, increasing the number of nodes improves accuracy. It is interesting to note, however, the difference between DSPM\_1\_1 and DSPM\_1\_2, where we can see that an increase in the number of tertiary nodes improves accuracy at the cost of a very slight increase in computation time (Figure 11). The use of tertiary nodes thus appears to be an efficient way to improve accuracy.

To sum up, it seems clear that the DSPM at the forward step and the SO\_RBM at the backward process both ensures better accuracy compared to similar methods. The hybrid approach proposed in this work is the combination of these two methods.

### Accuracy for complex media

Many alternative approaches were examined in the previous sections for forward and the backward processes. Based on the tests, the proposed solution to achieve the best accuracy while limiting computing cost is to combine the following two elements: (1) use DSPM to propagate the wave front during the forward process and (2) use SO-RBM to trace back the ray path from each receiver to the corresponding source. This approach ensures the best compromise between accuracy and computational cost. This solution can be considered as a hybrid combination between a modified version of the SPM and the steepest gradient raytracing method based on the SO-RBM.

In order to further investigate its accuracy, the proposed approach was used to calculate travel times for other velocity models. Tests were done to see if the performance metrics observed in the initial

phase of this work could withstand more complex slowness models. The new tests used the same basic model that consists of a  $1 \text{ km}^3$  volume that contains 11,026 primary nodes. The seismic source and receiver positions also remained unaltered. For these tests, the only change was the inclusion of more complex slowness models. In all, a total of 4 slowness models were created, as shown in Figure 12. The first one (Model A) is a vertical constant gradient of slowness squared:  $s^2(z)=0.83+0.16z$  or  $V(z) = \frac{1}{\sqrt{0.83+0.16z}}$ . The second is a 3 D linear velocity gradient model. Velocity values are expressed at each point as a first order polynomial of position coordinates:  $V(x, y, z) = 2.04+0.15x+0.35y+0.95z$ . Model C is a radial model in which velocity values decrease linearly as a function of distance  $d$  from the seismic source:  $V=3.52-1.25d$ . The last model (Model D) was built using random slowness values taken from a uniform distribution of velocity values in the interval 2000-3400 m/s. To get a smooth velocity aspect similar to those obtained with inversion process, a low-pass moving-average filter was applied to this last model. The filter window contains for each node their first and second neighbours.

In order to test the accuracy for each model, true travel time values have been determined for each receiver using one of two strategies. For models for which an analytical solution exists, which is the case for the first three models, the true values were computed using the travel time analytic expression. The analytical solution is given by Fomel *et al.* (2009) for Model A and Model B and by Cerveny (2005) for Model C. When no analytical solution can be considered, as for Model D, an approximate solution is determined by computing travel times to receivers using the standard SPM and a high mesh resolution. Refinement was ensured here by adding about 1920 secondary nodes at each cell (SPM\_30). A domain of nearly 52 million nodes was thus built. The travel times were calculated at the backward step with the SO-RBM scheme and are considered as the true values to be compared to those obtained using the proposed solution.

Results are presented in Figure 13. Note that only relative errors were considered to evaluate each model accuracy. The absolute errors were excluded here since it depends on the slowness distributions along ray paths which are not the same for all models.

Model A shows a median relative error of 0.03%, a relative RMSE of 0.05% and maximum relative error of 0.22%. The same values were recorded for Model B. Error for Model C is slightly higher: recorded values reach 0.22% for the median relative error, 0.3% for the RMSE error and 0.9% for the infinity norm. Note that this high error is more strongly influenced by the slowness interpolation process at points constituting ray path rather than to the raytracing step. Comparison of interpolated and true slowness values for this model shows a significant deviation specially near the seismic source, with a difference reaching 1.11%. Furthermore, travel times calculated along ray paths using the true slowness values show error closer to the previous two models: a median relative error of 0.03%, a relative RMSE of 0.05% and a maximum relative error of 0.52% were calculated.

The random slowness model (Model D) leads to some negative differences with the reference model: the travel times computed using the proposed method are smaller than those derived from the standard SPM. Since the reference travel time values were calculated based on the SPM with a high density of nodes, the process can be seen as a comparison between the two methods more than an accuracy test. Qualitatively travel times computed using the proposed method can be considered as acceptable: they are slightly different than those given by the SPM. For Model D, the median relative difference is about 0.02% while the relative RMSE and the maximum difference are estimated respectively to 0.04% and 0.2%.

The different tests showed that the combination of the SPM method with dynamic nodes and the SO-RBM for raytracing can work efficiently for a number of different slowness models. Although the accuracy varies from one model to another, the proposed method is still able to calculate seismic travel times with negligible errors even when a relatively coarse mesh is used. For example, with Model D, the proposed solution, using only 84,014 nodes (primary, secondary, and tertiary), ensures accuracy within the same order of magnitude as the standard SPM operating with more than 618 times the number of nodes. The proposed solution is therefore less time consuming since only 4.3 s are required to calculate travel times for all receivers while the same task is just below 875 times longer with the classic SPM at the required mesh resolution.

#### Accuracy for a steep topography model

Unstructured meshes are usually used to better approximate geological domains that present steep topography or internal discontinuity like faults, cavities or stratigraphic unconformities. These meshes generally outperform regular meshes when fitting irregular borders or lithologic contacts. For that reason, we present in this section performance tests for such particular domains. We considered for this purpose the case of an irregular topography model, representative of a cliff. The horizontal dimensions are 450 m by 530 m and the relative elevation of the cliff ranges from 50 m to 330 m (Figure 14). To mimic a hypocentre relocation problem, we used in this test one receiver and 1000 seismic sources randomly placed in the model. The receiver is placed in the centre of the model at the surface. The domain was discretized using about 12,000 primary nodes and 56,000 cells. A variable mesh density is adopted in this test, with cells 10 times finer at the top of the model to better approximate the topology. The choice is also justified by our intention to test the proposed approach on mesh with variable density, another advantage of tetrahedral meshes widely exploited. A total of 16 secondary nodes are used to refine the mesh everywhere in the whole domain while

144 tertiary nodes are added in each cell locally around seismic sources at a radius of 100 m (scheme DSPM\_2\_2). This raises the number of primary and secondary nodes close to 277,000, and the number of tertiary nodes is about 180,000 on average.

We performed this test with a homogeneous velocity having a value of 3.5 km/s (Figure 14). Using a constant velocity model allows observing the effect of topography on the seismic rays. For instance, most rays outgoing from seismic sources on the surface of the model are forced to follow topographic irregularities to reach the receiver.

As for the random velocity model, there is no analytic solution to calculate the true travel times for this model. To evaluate the efficiency of the proposed approach, approximative travel time values calculated via the classical SPM after update at the backward step are taken as a reference solution.

To further improve accuracy, the mesh used with the DSPM method was refined by splitting the cells, which allows preserving the topographic surface. The number of nodes used to get this solution is over 289 million nodes, of which over 85,000 are primary (SPM\_25). The error generated by this high-resolution mesh is expected to be negligible, and the obtained travel times are considered as the true solution.

The errors are calculated for each seismic source and shown in Figure 15. For the absolute error, the average value is about 0.05 ms, the maximum does not exceed 0.32 ms and the RMSE is of 0.07 ms.

These values correspond to an average relative error in the order of 0.08%, a relative RMSE of 0.12% and a maximum of 0.76%. Thus, deviations between travel times calculated via SPM and the proposed method remain negligible for this model also. This indicates that the approach operates well in the case of steep topography and concave domains. Furthermore, these results are similar to

errors obtained in the case of models without topography and show that the topography has generally little effect on the performance of our approach.

## Discussion

Through this paper, various tests and comparisons have been made to find an efficient approach to compute seismic travel times. In this section, we discuss the innovative sides of the work as well as the interconnections that may exist with previous studies. Firstly, we presented in this work a new SPM version called the DSPM. The concept behind this solution is simple to set up, proved to increase accuracy and reduce significantly the computation cost. Instead of adding many secondary nodes homogeneously over the study domain as suggested by Gruber and Greenhalgh 1998, the DSPM operates by increasing the density of nodes in the vicinity of the source using tertiary nodes. It was shown that this approach permits enhancing travel time accuracy not only in the vicinity of sources but further for the entire domain. Using the DSPM implies specifying the subdomain radius where the tertiary nodes are added. A series of simple tests on a mesh with a narrow cell size distribution indicates that the benefits of using a radius larger than three times the average cell edge length are marginal, even at low mesh resolution and node density. These observations should, however, be reassessed for meshes with wide cell size distribution.

The second point to discuss is the accuracy of the first order LTI solver on tetrahedral meshes. For the forward step, our results match those obtained by Lelièvre *et al.* (2011): the first order LTI solver suffers from some accuracy problems to ensure an efficient travel time computation at low and moderate mesh resolutions. Moreover, we tested this LTI solver to trace back seismic ray paths in the case of tetrahedral meshes. We proved that this mechanism is still not enough even when calculating travel times at a backward step along ray paths. Thus, we join to Lelièvre *et al.* (2011) to

emphasize the importance of creating a second order LTI solver in order to enhance the solver accuracy. Until conceived that solver, the hybrid approach presented in this work is still among the most efficient ways to calculate seismic travel time on tetrahedral meshes. Despite its high accuracy and lower computation cost, this approach finds its application limits when it remains mainly recommended for smooth velocity models. The stationarity of velocity models is needed here since the gradient is calculating locally using many cells. With models showing sharp variations, the travel time gradients calculating at these vicinities will have different directions and cannot be correctly summed to get a representative average. Approximating the ray path using the average gradient is no longer valid in this case. This issue is more common to low-resolution meshes and can be solved by refining local domain regions where sharp variations are expected. However, this solution supposes that the positions of the sharp velocity variations are well known which is not always the case. Finally, we have presented an application of the proposed approach on a steep topography model. Results showed that projection the seismic rays upon the irregular domain borders can be an efficient solution to calculate travel times in such cases. This avoids to make any transformation into curvilinear coordinates or other systems as made in some previous approaches aiming to calculate travel time in irregular topography domains (Lan and Zhang 2013).

## Conclusion

In this work, an efficient method to compute seismic travel times for unstructured tetrahedral meshes is presented. The method uses a combination of approaches carefully chosen to reach maximum precision and efficiency. The proposed solution adopts the strategy of calculating travel times along the ray paths as was introduced in the LTIM. The method is justified since seismic rays represent a good approximation of the paths along which the seismic waves traveled. This allows for better

accuracy without the necessity of increasing mesh resolution and thus reduces memory and computational requirements.

Similar to the LTIM, the proposed solution operates in two separate steps: (1) a forward process to propagate the wave front from the seismic sources to the whole domain and (2) a backward process to back-trace ray paths for all source-receiver pairs and compute final travel times. Many alternatives were tested for each step. Results from our tests for the forward process showed some advantages of the SPM update mechanism compared to the LTI local solver. Indeed, the LTI solver, widely stable when a regular grid is used, suffers from accuracy issues when used on unstructured meshes that are based on tetrahedral elementary cells. With such cell shape, neither the node number nor their arrangement are suitable for linear interpolation of the travel times which leads to a rapid degradation of the accuracy of results. Achieved tests confirmed that the precision of the LTI solver depends significantly on the mesh topology. The results from the tests involving the SPM updater demonstrated that this mechanism works really well with tetrahedral cells and that it can be an alternative solution for the cases where there is a violation of the planar wave approximation proximal to the source nodes or when the causality conditions are no longer valid near cell faces with obtuse angles. The results also showed that adding secondary nodes dynamically around sources can enhance travel time computation accuracy everywhere in the domain and especially near source nodes. The DSPM approach was thus adopted for the forward process.

For the backward process, two categories of methods were tested: (1) the method based on the LTI solver and (2) those using the steepest travel time gradient to approximate the ray paths from receiver to corresponding seismic source. The LTI solver mechanism proved less accurate for the same reasons as enumerated in the discussion of the forward process. The steepest gradient approach using the second-order regression-based method (SO-RBM) was therefore chosen in order to compute the travel time gradient. This approach provides the best compromise between accuracy and computation time and works well with relatively coarse meshes, which is a big advantage for

inversion problems. The method has been parallelized, another advantage for inversion problems. Note that the presented approach emphasizes the major role of the hybridization between different raytracing methods to enhance accuracy and computational cost when calculating seismic travel times. A hybrid approach such as the proposed one can operate more efficiently than the standard versions of SPM and of LTIM.

The method was tested on various models that include relatively sharp velocity gradients and irregular topography, and results have shown an acceptable error and computation time. The results of the tests also emphasize the crucial role of the slowness interpolation method and the error that can be introduced at this step. Future work includes testing more accurate local solvers such as those based on cubic spline interpolation (Zhang *et al.* 2013) and using the proposed solution as part of inversion codes. More accurate slowness interpolation approaches for the backward step will also be tested.

#### **Acknowledgements**

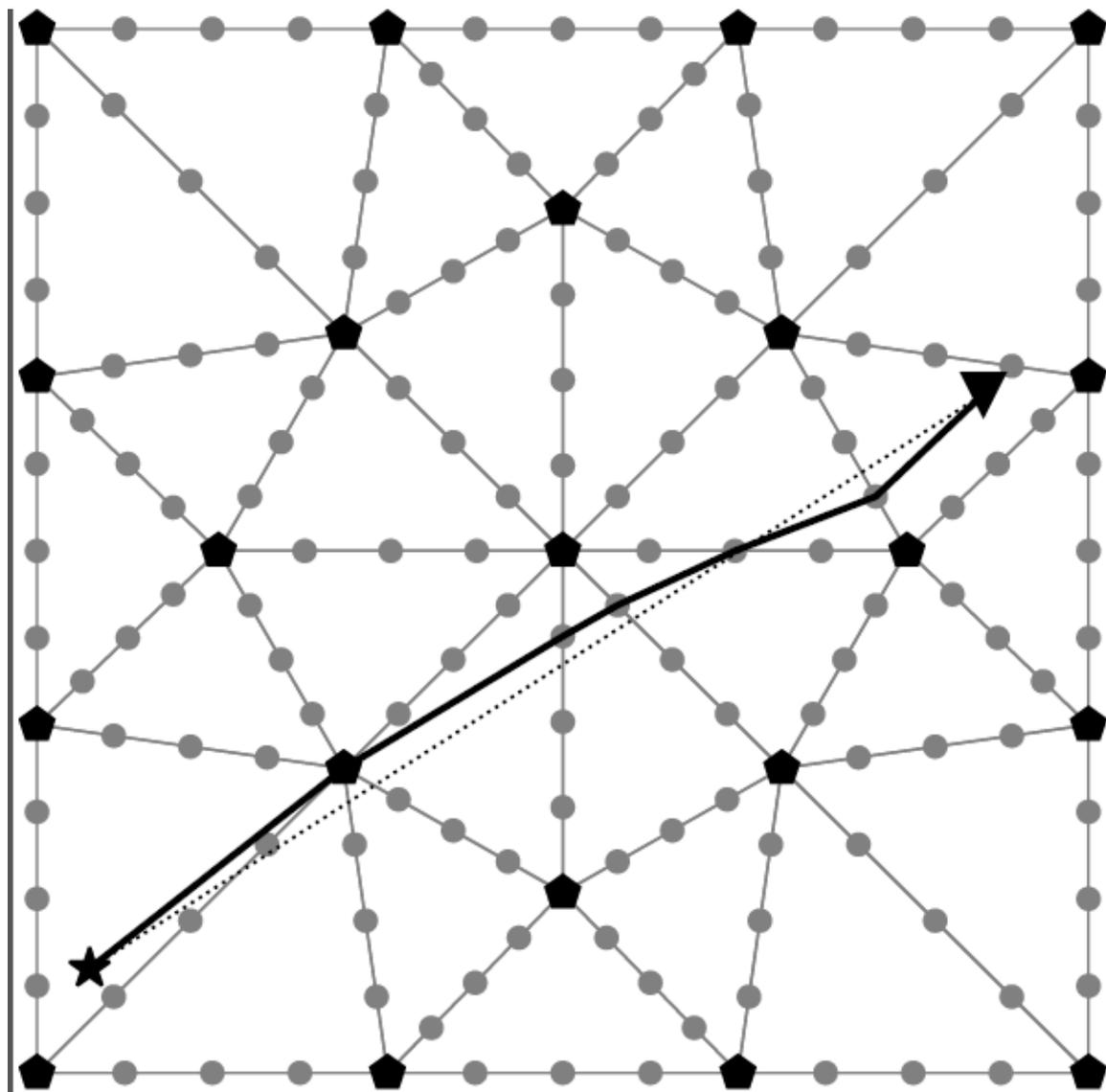
This research was funded by the Fonds de recherche du Québec –Nature et technologie (FRQNT), project 2015-MI-194018.

#### **Data availability statement**

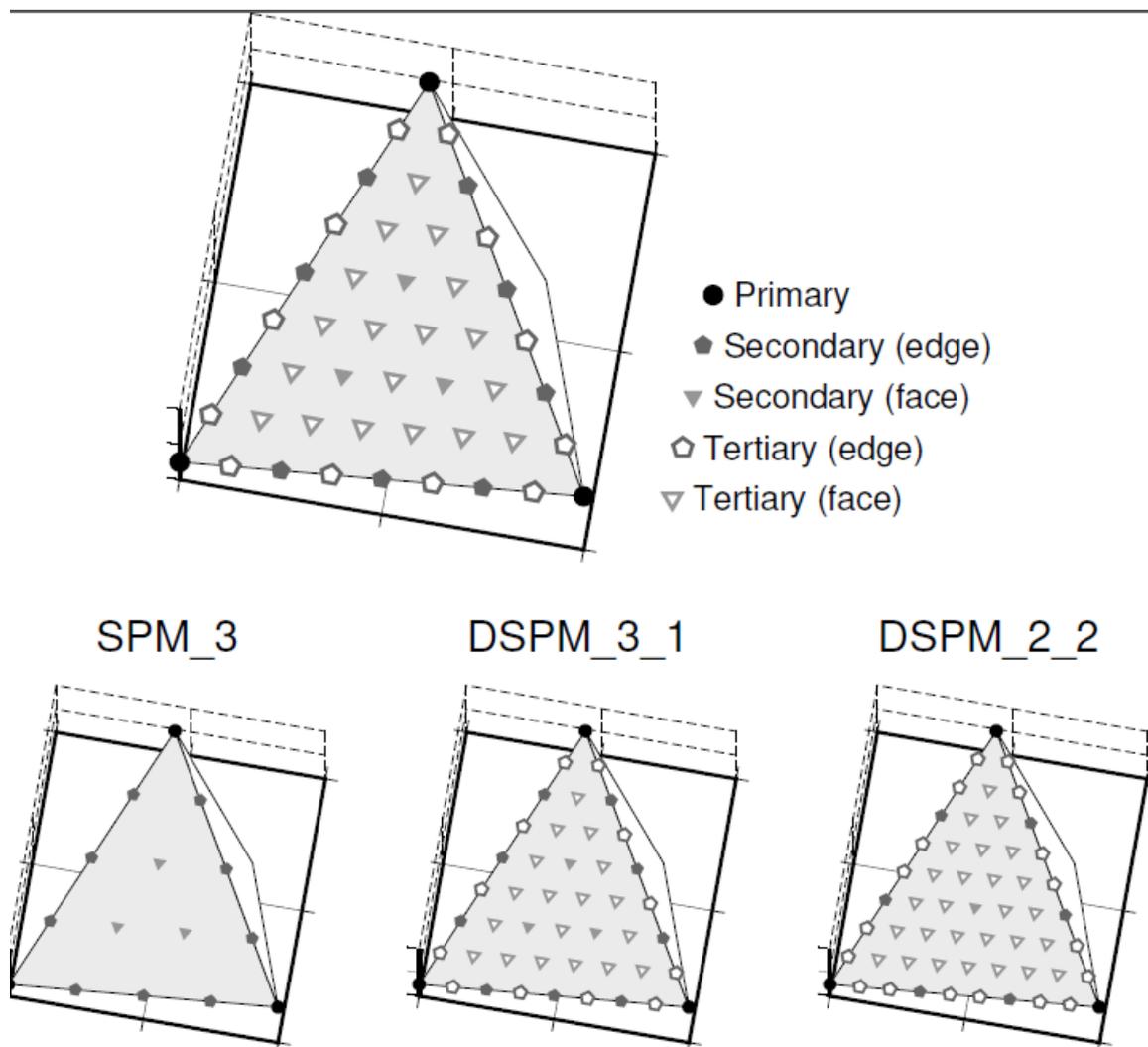
The source code and data that support the findings of this study are openly available on GitHub (Giroux and Nasr, 2018).

**List of the figure legends**

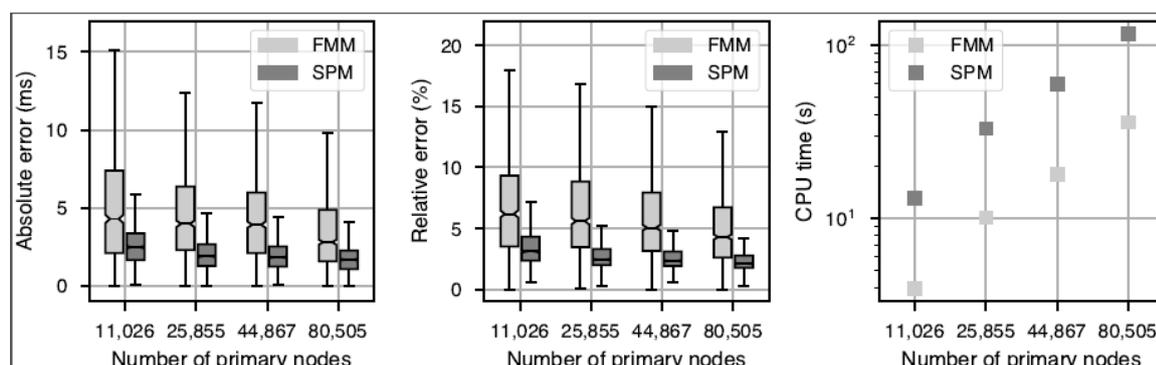
**Figure 1:** Illustration of SPM raytracing in a 2D homogeneous velocity model with paths imposed to secondary nodes. Black pentagons are primary nodes, grey circles are secondary nodes. The solid black line is the discrete path crossing secondary nodes. The path should be linear from the source to the receiver (dashed line).



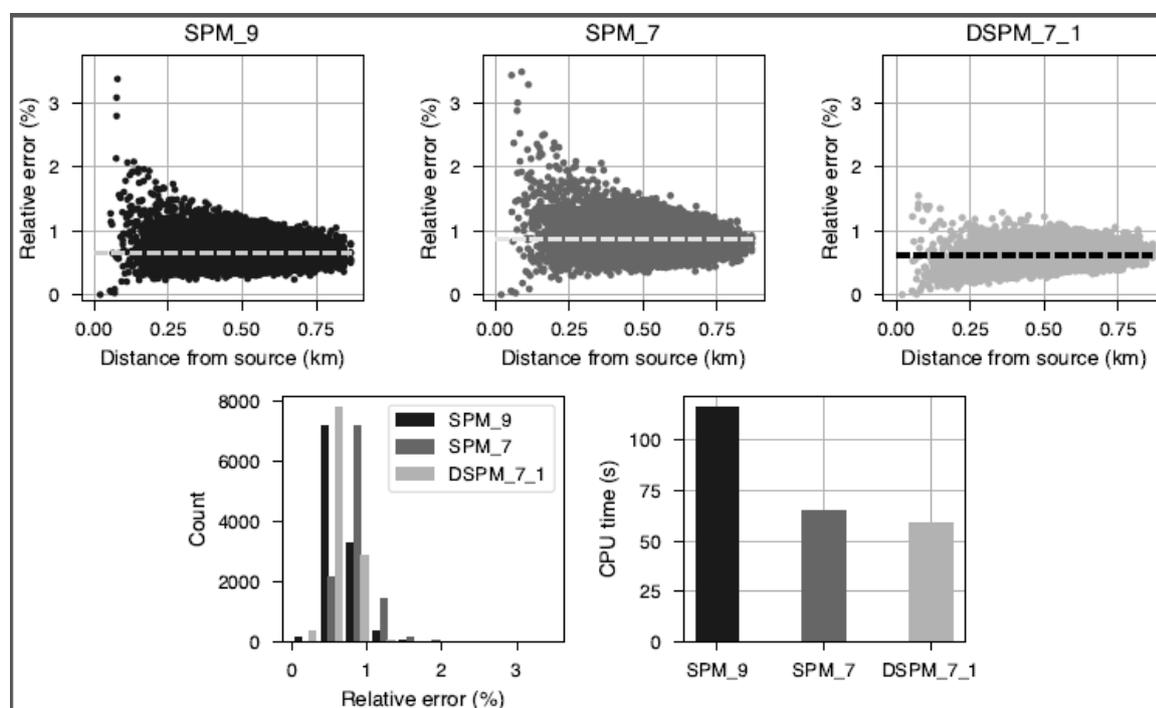
**Figure 2:** Cell face showing different node categories and the nomenclature used to specify discretization. The first number indicates the number of secondary nodes on an edge, and the second number indicates the number of tertiary nodes between secondary nodes, on an edge.



**Figure 3:** Absolute (left) and relative (right) error, as well as CPU time (right) for the FMM and SPM with different mesh resolution. For the SPM, 3 secondary nodes were added in each cell face and edge. The target and the box limits correspond respectively to error median, first and third quartiles, and the whisker refers to the 5% and 95% percentiles (McGill, Tukey and Larsen1978).

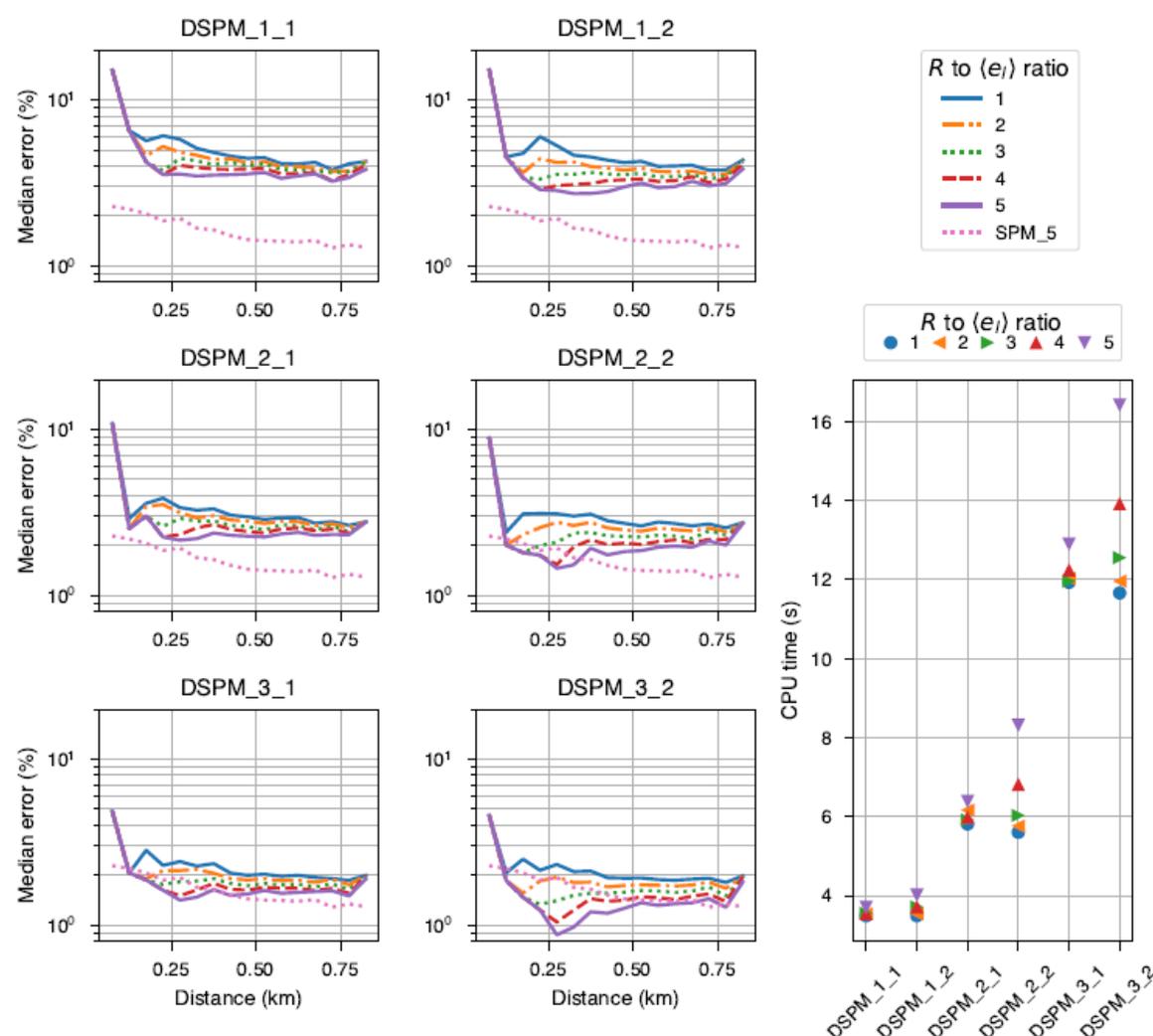


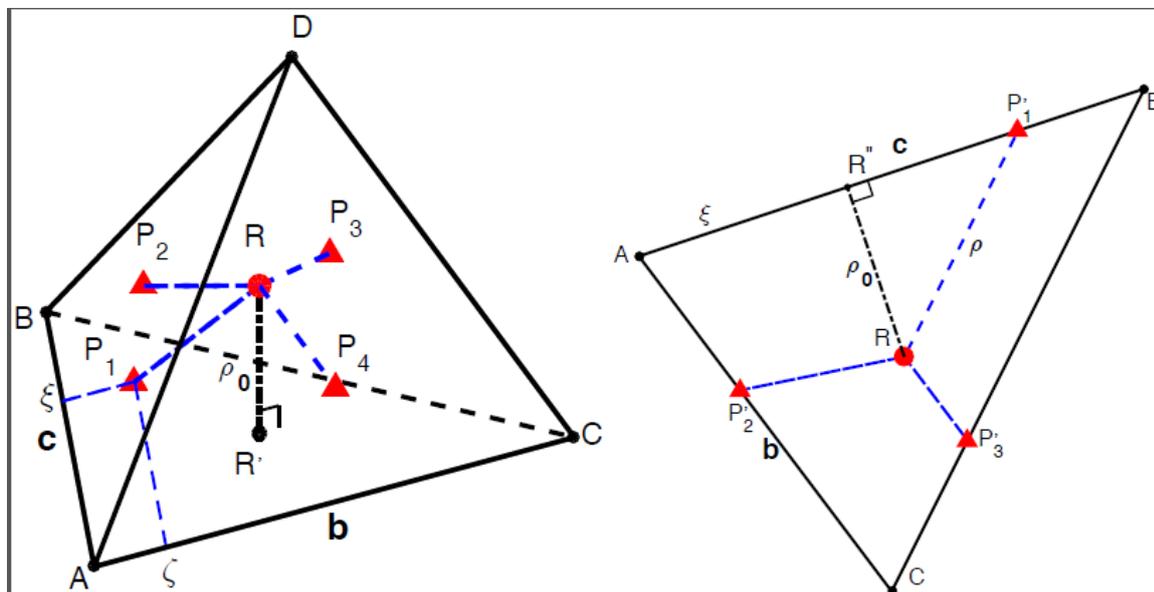
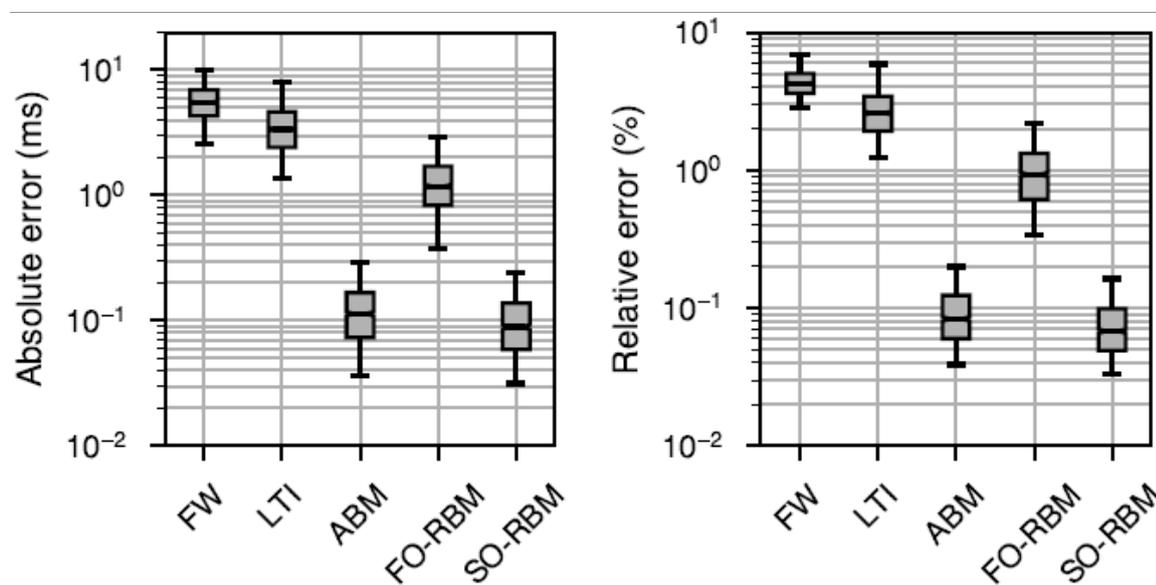
**Figure 4:** Relative error versus distance from source for three discretization schemes. Dashed lines are the average errors. The bottom left figure is the relative error histogram for the tested schemes, and the bottom right is the CPU time for the three runs.



This article is protected by copyright. All rights reserved.

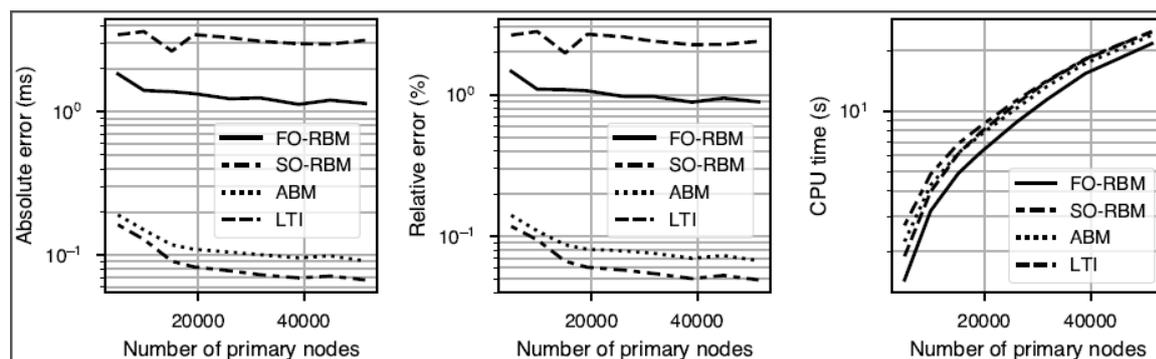
**Figure 5:** Relative travel time error as function of distance from the source for different DSPM schemes and subdomain radii  $R$ . Radii are normalized by the mean cell edge length ( $\langle e_i \rangle$ ). Upper right legend refers to the error plots.



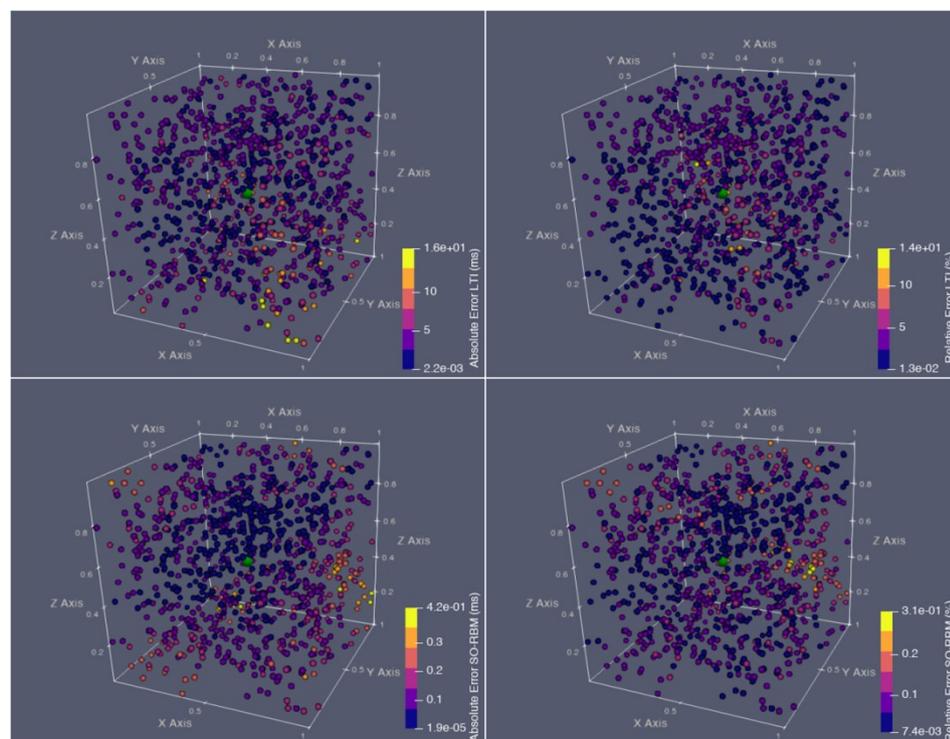
**Figure 6:** Geometric scheme for raytracing LTI solver at tetrahedron cell scale (left) and at triangle face (right)(Lelièvre *et al.* 2011).**Figure 7:** Box and whisker plots showing accuracy comparison between 4 backward process methods as well as travel times computed at the forward step by interpolation.

This article is protected by copyright. All rights reserved.

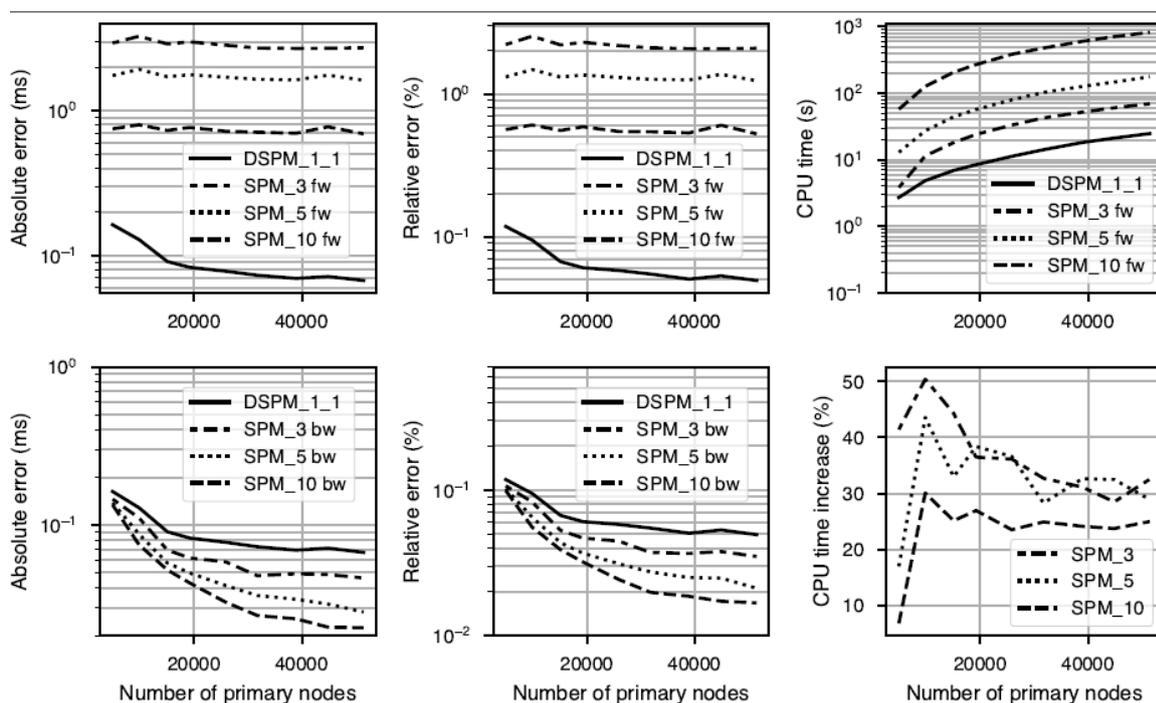
**Figure 8:** Performance comparison as a function of mesh resolution between 4 backward process methods with the DSPM\_1\_1: average of absolute error (left), average of relative error (centre), and CPU time (right). Velocity model, domain dimensions, seismic source and receiver positions remain those described in section 2.1.



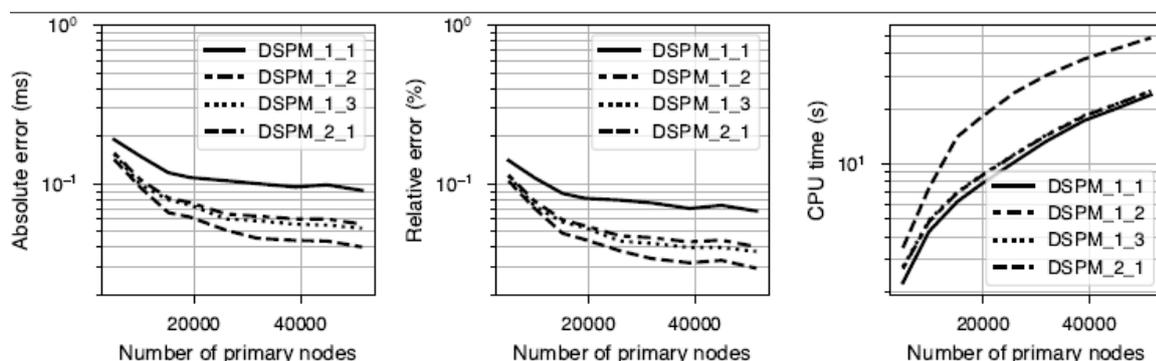
**Figure 9:** Spatial distribution of absolute and relative error for the LTI (top) and SO-RBM (bottom) backward process methods. Source position is indicated by the green sphere.



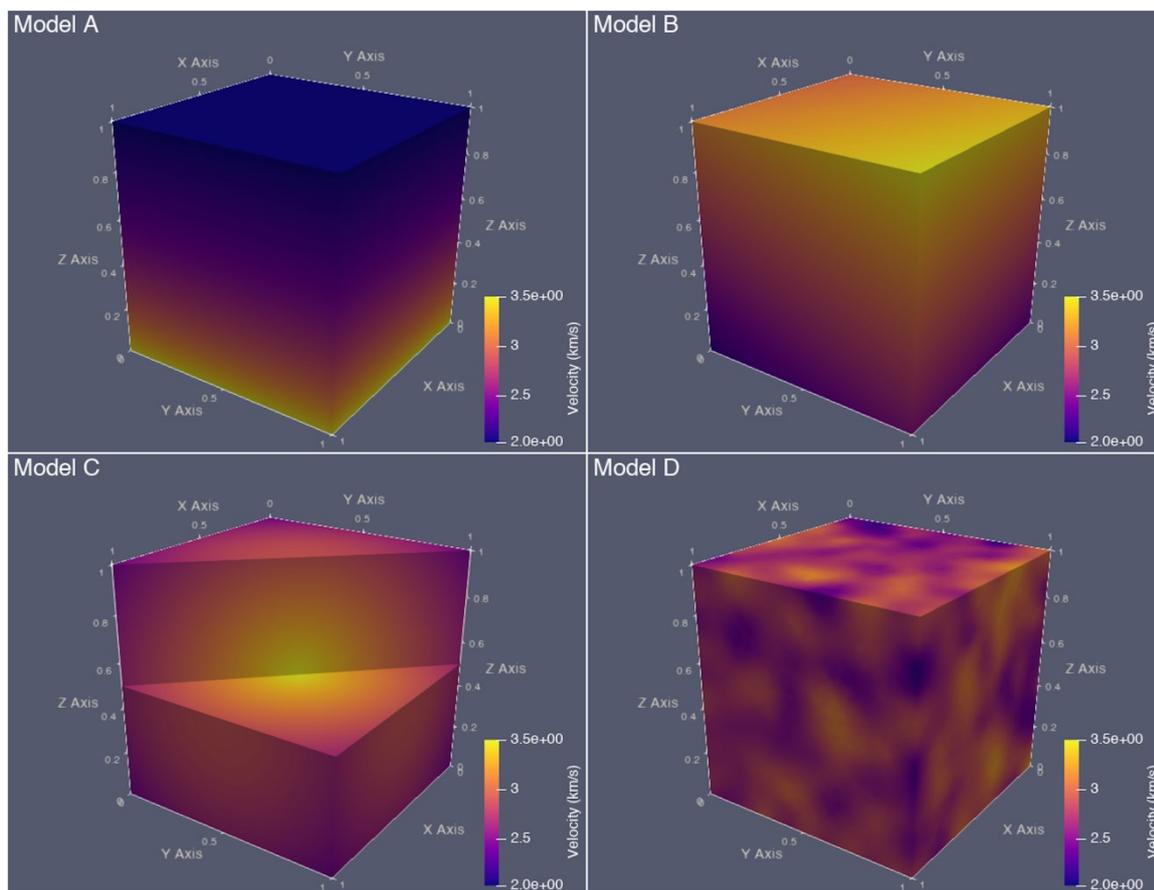
**Figure 10:** Performance comparison between the DSPM with backward travel time calculation (SO-RBM) and the travel times computed with the standard SPM at the forward step (labelled fw in top figures) and at the backward step (marked by bw in the bottom figures). Velocity model, domain dimensions, seismic source and receiver positions remain those described in section 2.1.

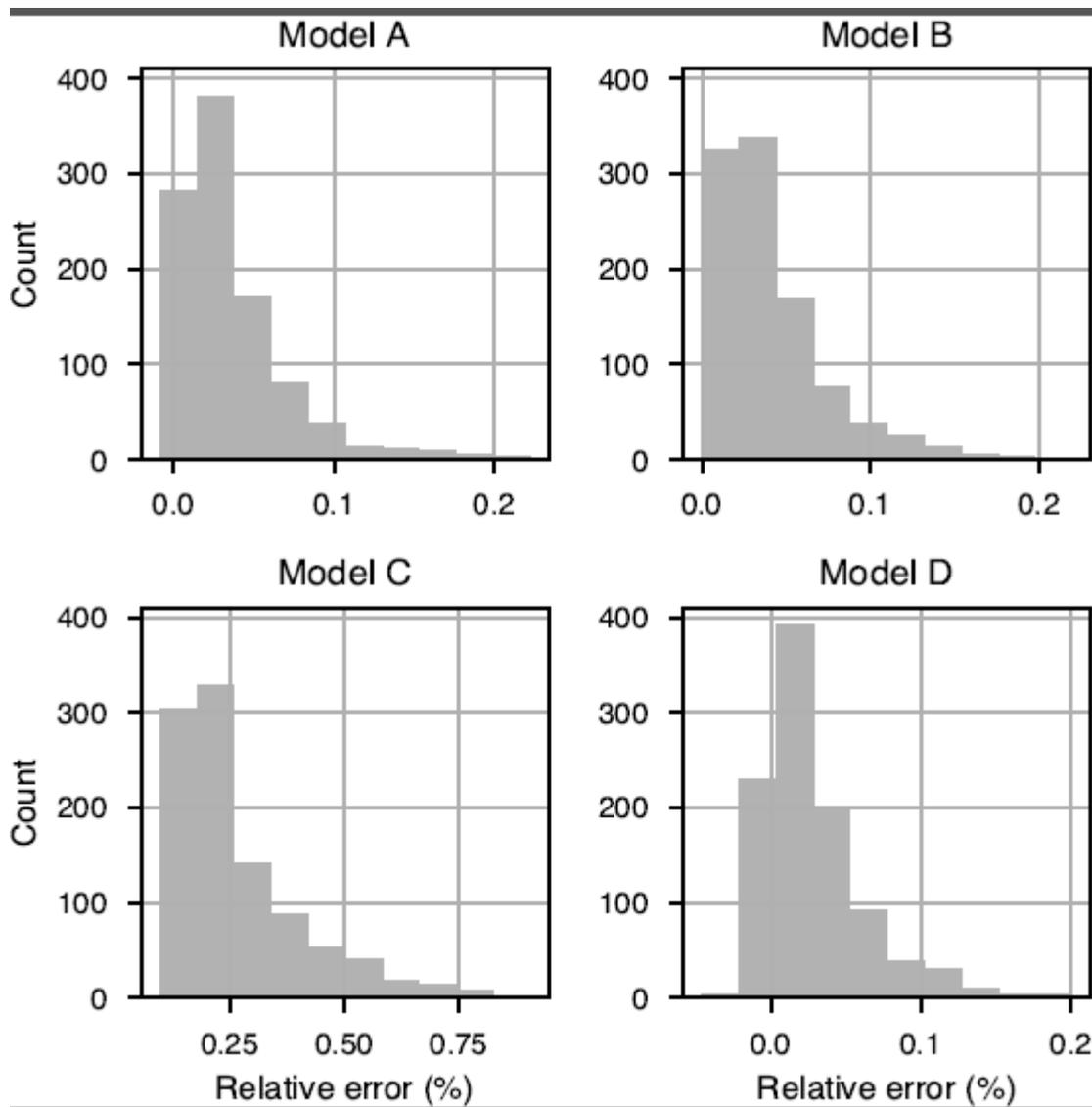


**Figure 11:** Influence of the number of secondary and tertiary nodes on the performance of the DSPM, as function of mesh resolution.

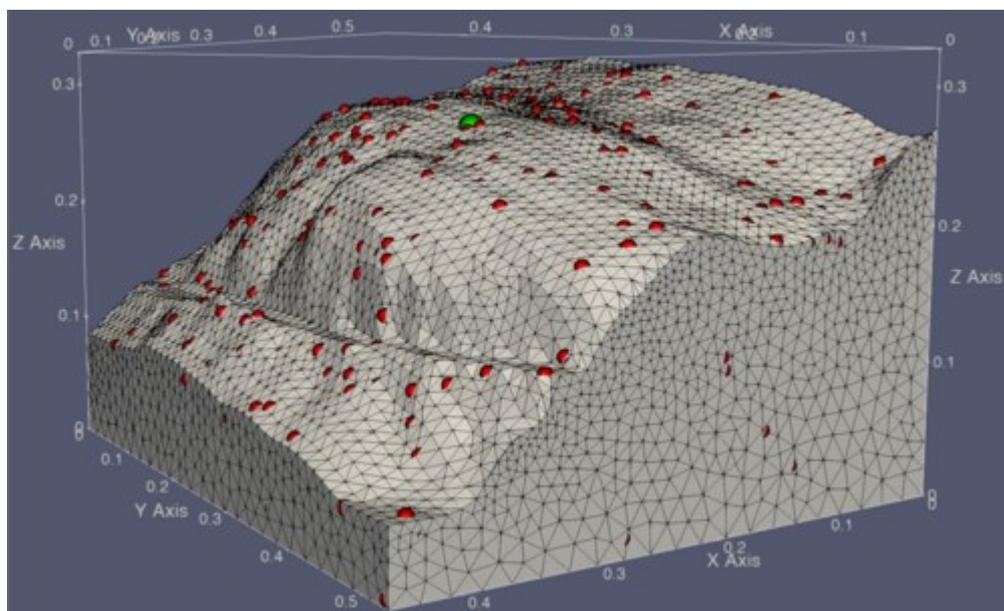


**Figure12:** Velocity models used to test the proposed approach. Model A: constant gradient of slowness squared, Model B: 3D constant gradient of velocity, Model C: radial velocity model, Model D: random velocity model.

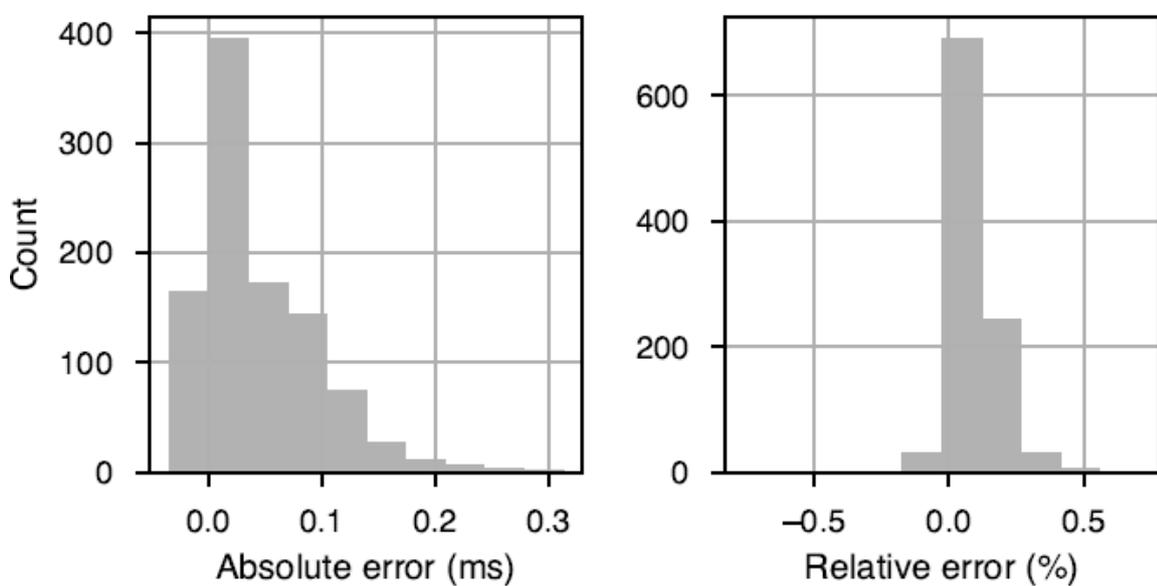


**Figure 13:** Histograms of travel time relative error for the models in Figure 12.

**Figure 14:** The cliff model used to test the approach on an irregular-topography model. Green dot is the receiver while the red dots are the visible seismic sources that appear on the surface.



**Figure 15:** Histograms of relative and absolute errors for the model in Figure14



## References

Asakawa E. and Kawanaka T. 1993. Seismic ray tracing using linear traveltime interpolation.

*Geophysical Prospecting* **41**, 99-111.

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

Bai C.Y., Li X.L. and Tang X.P. 2011. Seismic wavefront evolution of multiply reflected, transmitted, and converted phases in 2D/3D triangular cell model. *Journal of Seismology* **15**, 637-652.

Bai C.Y., He L.Y., Li X.W. and Sun J.Y. 2018. Simultaneous travel time tomography for updating both velocity and reflector geometry in triangular/tetrahedral cell model. *Journal of Seismology* **22**, 569-574.

Capozzoli A., Curcio C., Liseno A. and Savarese S. 2014. A comparison of fast marching, fast sweeping and fast iterative methods for the solution of the eikonal equation. *Telfor Journal* **6**, 142-147.

Cardarelli E. and Cerreto A. 2002. Ray tracing in elliptical anisotropic media using the linear traveltime interpolation (LTI) method applied to traveltime seismic tomography. *Geophysical Prospecting* **50**, 55-72.

Cerveny V. 2005. *Seismic ray theory*. Cambridge University Press.

Correa C.D., Hero R. and Ma K.L. 2011. A comparison of gradient estimation methods for volume rendering on unstructured meshes. *IEEE Transactions on Visualization and Computer Graphics* **17**, 305-319.

- Dahoe A. and Cant R. 2004. On least-squares gradient reconstruction and its application in conjunction with a Rosenbrock method. Technical Report, Ulster University, Antrim.
- Dijkstra E.W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* **1**, 269-271.
- Ding P., Yang G., Li Z., Zhang K. and Liu Q. 2016. Ray tracing based on 3D multi-stencils fast marching algorithm for complex near-surface model. *Geophysical Prospecting for Petroleum* **55**, 483-492, in Chinese.
- Doser D.I. 2006. Relocations of earthquakes (1899-1917) in south-central Alaska. *Pure and Applied Geophysics* **163**, 1461-1476.
- Erem B. and Brooks D.H. 2011. Differential geometric approximation of the gradient and Hessian on a triangulated manifold. Proceedings/IEEE International Symposium on Biomedical Imaging: from nano to macro, Chicago, IL, USA.
- Fomel S. 1997. A variational formulation of the fast marching eikonal solver. *SEP-95: Stanford Exploration Project*, 127-147.
- Fomel S., Luo S.T. and Zhao H.K. 2009. Fast sweeping method for the factored eikonal equation. *Journal of Computational Physics* **228**, 6440-6455.
- Fu Z., Kirby R.M. and Whitaker R.T. 2013. A Fast Iterative Method for Solving the Eikonal Equation on Tetrahedral Domains. *SIAM Journal on Scientific Computing* **35**, c473-c494.
- Gavete L., Gavete M.L. and Benito J.J. 2003. Improvements of generalized finite difference method and comparison with other meshless method. *Applied Mathematical Modelling* **27**, 831-847.

- Geuzaine C. and Remacle J. 2009. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering* **79**, 1309-1331.
- Giroux B. and Larouche B. 2013. Task-parallel implementation of 3D shortest path raytracing for geophysical applications. *Computers and Geosciences* **54**, 130-141.
- Giroux B. 2014. Comparison of grid-based methods for raytracing on unstructured meshes. 84th SEG Annual Meeting, Denver, Colorado, USA, Expanded Abstracts, 3388-3392.
- [Data set] Giroux B. and Nasr M. 2018. ttc with DSPM: groupeLIAMG/ttc, Version (v0.1).767  
doi:10.5281/zenodo.1495231.
- Göktürkler G. and Balkaya Ç. 2010. Traveltime tomography of crosshole radar data without ray tracing. *Journal of Applied Geophysics* **72**, 213-224.
- Gruber T. and Greenhalgh S.A. 1998. Precision analysis of first-break times in grid models. *Geophysics* **63**, 1062-1065.
- Grunberg M., Genaud S. and Mongenet C. 2004. Seismic ray-tracing and Earth mesh modeling on various parallel architectures. *Journal of Supercomputing* **29**, 27-44.
- Huang Y.Q., Zhang J.Z. and Liu Q.H. 2011. Three-Dimensional GPR Ray Tracing Based on Wavefront Expansion With Irregular Cells. *IEEE Transactions on Geoscience and Remote Sensing* **49**, 679-687.
- Hysing S.R. and Turek S. 2005. The Eikonal equation: numerical efficiency vs. algorithmic complexity on quadrilateral grids. *Proceedings of ALGORITMY 2005*, 22-31.

- Jeong W.K. and Whitaker R.T. 2008. A Fast Iterative Method for Eikonal Equations. *SIAM Journal on Scientific Computing* **30**, 2512-2534.
- Kim S. 2001. An  $O(N)$  Level Set Method for Eikonal Equations. *SIAM Journal on Scientific Computing* **22**, 2178-2193.
- Lan H. and Zhang Z. 2013. Topography-dependent eikonal equation and its solver for calculating first-arrival traveltimes with an irregular surface. *Geophysical Journal International* **193**, 1010-1026.
- Ledergerber C., Guennebaud G., Meyer M., Bacher M. and Pfister H. 2008. Volume MLS ray casting. *IEEE Transactions on Visualization and Computer Graphics* **14**(6), 1372-1379.
- Leidenfrost A., Ettrich N., Gajewski D. and Kosloff D. 1999. Comparison of six different methods for calculating traveltimes. *Geophysical Prospecting* **47**, 269-297.
- Lelièvre P.G., Farquharson C.G. and Hurich C.A. 2011. Computing first-arrival seismic traveltimes on unstructured 3-D tetrahedral grids using the Fast Marching Method. *Geophysical Journal International* **184**, 885-896.
- Li T., Liu J. and Zhang J. 2019. A 3D reflection ray-tracing method based on linear traveltime perturbation interpolation. *Geophysics* **84**, T181-T191.
- Li X.G. and Ulrych T.J. 1993. LTI formulations and application to curved wave fronts. *Journal of Seismic Exploration* **2**, 239-246.
- Majdański M., Grad M., Guterch A. and the SUDETES 2003 Working Group 2006. 2-D seismic tomographic and ray tracing modelling of the crustal structure across the Sudetes Mountains basing on SUDETES 2003 experiment data. *Tectonophysics* **413**, 249-269.

- Mavriplis D. 2003. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. 16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, USA, 3986.
- McGill R., Tukey J.W. and Larsen W.A. 1978. Variations of box plots. *The American Statistician* **32**, 12-16.
- Moser T.J. 1991. Shortest-Path Calculation of Seismic Rays. *Geophysics* **56**, 59-67.
- Nakanishi I. and Yamaguchi K. 1986. A numerical experiment on nonlinear image reconstruction from first-arrival times for two-dimensional island arc structure. *Journal of Physics of the Earth* **34**, 195-201.
- Penz S. 2012. Modélisation et inversion de données électriques en courant continu: vers une prise en compte efficace de la topographie. PhD Thesis, Ecole Nationale Supérieure des Mines de Paris, in french.
- Podvin P. and Lecomte I. 1991. Finite-Difference Computation of Traveltimes in Very Contrasted Velocity Models - a Massively Parallel Approach and Its Associated Tools. *Geophysical Journal International* **105**, 271-284.
- Qian J.L., Zhang Y.T. and Zhao H.K. 2007. Fast sweeping methods for eikonal equations on triangular meshes. *SIAM Journal on Numerical Analysis* **45**, 83-107.
- Serretti P. and Morelli A. 2011. Seismic rays and traveltimes tomography of strongly heterogeneous mantle structure: application to the Central Mediterranean. *Geophysical Journal International* **187**, 1708-1724.

Sethian J.A. 1999 a. Advancing interfaces: level set and fast marching methods. Proceedings of the international conference on industrial and applied mathematics, plenary lectures.

Sethian J.A. 1999 b. Fast marching methods. *SIAM Review* **41**, 199-235.

Symes W.W., Versteeg R., Sei A. and Tran Q.H. 1994. Kirchhoff simulation migration and inversion using finite-difference travel-times and amplitudes. Annual Report, The Rice Inversion Project.

ter Braak C.J. and Juggins S. 1993. Weighted averaging partial least squares regression (WA-PLS): an improved method for reconstructing environmental variables from species assemblages. *Hydrobiologia* **269**, 485-502.

Trampert J. and van der Hilst R.D. 2005. Towards a quantitative interpretation of global seismic tomography. In: *Earth's Deep Mantle: Structure, Composition, and Evolution*, Geophysical Monogram series, vol. 160 (Eds. R. van der Hilst, J. D. Bass, J. Matas, J. Trampert), pp. 47-62. American Geophysical Union.

Vanelle C. and Gajewski D. 2002. Second-order interpolation of traveltimes. *Geophysical Prospecting* **50**, 73-83.

Wang F., Qu X., Liu S., Li Y. and Wu J. 2014. A new ray tracing approach based on both multi-stencils fast marching and the steepest descent. *Oil Geophysical Prospecting* **49**, 1106-1114.

Yoon K., Shin C., Suh S., Lines L.R. and Hong S. 2003. 3D reverse-time migration using the acoustic wave equation: An experience with the SEG/EAGE data set. *The Leading Edge* **22**, 38-41.

- Zhang D., Zhang T.T., Zhang X.L., Yang Y., Hu Y. and Qin Q.Q. 2013. A new 3-D ray tracing method based on LTI using successive partitioning of cell interfaces and traveltimes gradients. *Journal of Applied Geophysics* **92**, 20-29.
- Zhang H.J. and Thurber C.H. 2003. Double-difference tomography: The method and its application to the Hayward Fault, California. *Bulletin of the Seismological Society of America* **93**, 1875-1889.
- Zhang J., Shi J., Song L.-P. and Zhou H.-w. 2015. Linear traveltimes perturbation interpolation: a novel method to compute 3-D traveltimes. *Geophysical Journal International* **203**, 548-552.
- Zhang J.Z., Chen S.J. and Xu C.W. 2004. A method of shortest path raytracing with dynamic networks. *Chinese Journal of Geophysics* **47**, 1013-1018.
- Zhang J.Z., Huang Y.Q., Song L.P. and Liu Q.H. 2011. Fast and accurate 3-D ray tracing using bilinear traveltimes interpolation and the wave front group marching. *Geophysical Journal International* **184**, 1327-1340.
- Zhao H.K. 2004. A fast sweeping method for eikonal equations. *Mathematics of Computation* **74**, 603-627.
- Zhou Z.S., Zhang S.M. and Chen L.J. 2004. Seismic ray-tracing calculation based on parabolic traveltimes interpolation. *Journal of Central South University of Technology* **11**, 199-205.

**Table 1: Number of the variables to store and interpolation calls for a double and a simple meshing. The example assumes a model of almost 5,000 nodes for the inversion problem and 35,000 for the ray-tracing step. SPM is used with one secondary node.**

	Double meshing	Simple meshing (with SPM)
Nodes for the inversion process	4,920	4,920
Nodes at ray-tracing step	35,132	34,961
Total stored nodes	40,052	34,961
Cells for the inversion process	22,744	22,744
Cells at ray-tracing step	175,708	22,744
Total stored cells	198,452	22,744
Slowness interpolation operations	40,052	30,041

**Table 2: comparison of the SPM\_9, SPM\_7 and DSPM\_7\_1 scheme**

	SPM_9	SPM_7	DSPM_7_1
Secondary nodes per edge	9	7	7
Secondary nodes per face	36	21	21
Secondary nodes per cell	198	126	126
Tertiary nodes per edge	0	0	8
Tertiary nodes per face	0	0	84
Tertiary nodes per cell	0	0	384
Total nodes (x 10 <sup>6</sup> )	~4.7	~2.8	~3.0

**Table 3: Absolute and relative errors travel time computed at the forward step (FW) and for 4 backward process methods (LTI solver, ABM, FO-RBM, SO-RBM)**

	Forward	LTI solver	ABM	FO-RBM	SO-RBM
Average of absolute error (ms)	5.8	3.8	0.13	1.37	0.11
Absolute RMSE <sup>*</sup> (ms)	6.3	4.4	0.15	1.61	0.13
Average of relative error (%)	4.5	2.9	0.10	1.07	0.08
Relative RMSE (%)	4.8	3.3	0.11	1.25	0.09
Maximum absolute error (ms)	16.1	15.6	0.51	8.42	0.42
Maximum relative Error (%)	30.0	14.4	0.31	5.59	0.31

\* Root-mean-square error

#### Algorithm 1: 2D LTI solver for cell scale raytracing

inputs: Receiver R, face F, Slowness s

1  $t \leftarrow \infty$

2 for all edges that belong to face F

3     calculate  $\tilde{w}^2$

4     if  $\tilde{w}^2 > 0$

5         calculate  $\xi$

6          $P_i \leftarrow A + \xi \mathbf{c}$  // A is the edge node with smallest travel time

7         if  $T(P_i) + s RP_i < t$

8              $P \leftarrow P_i$

9              $t \leftarrow T(P_i) + s RP_i$

10         end if

11 end for

12 for all vertices V that belong to face F

13     if  $T(V) + s RV < t$

14          $P \leftarrow V$

15          $t \leftarrow T(V) + s RV$

16     end if

17 end for

18 return P

---

**Algorithm 2: 3D LTI solver for cell scale raytracing**


---

 Inputs: Receiver R, the  $n$  faces surrounding R, Slowness  $s$ 

```

1  $t \leftarrow \infty$ 
2 for  $i= 1$  to  $n$ 
3   calculate  $w^2$ 
4   if  $w^2 > 0$ 
5     calculate  $\xi$  and  $\zeta$ 
6      $P_i \leftarrow A + \xi \mathbf{c} + \zeta \mathbf{b}$  // A is the face node with smallest travel time
7     if  $T(P_i) + s RP_i < t$  //  $T(P_i)$  is the interpolated travel time at  $P_i$ 
8        $P \leftarrow P_i$ 
9        $t \leftarrow T(P_i) + s RP_i$ 
10    end if
11  end if
12   $P_i \leftarrow$  2D-LTI solver (R, faces(i), s)
13  if  $T(P_i) + s RP_i < t$ 
14     $P \leftarrow P_i$ 
15     $t \leftarrow T(P_i) + s RP_i$ 
16  end if
17end for
18 return P

```

---