A High-Order Fast-Sweeping Scheme for Calculating First-Arrival Travel Times with an Irregular Surface

by Haiqiang Lan and Zhongjie Zhang

Abstract Topography-dependent eikonal equation (TDEE) formulated in a curvilinear coordinate system has been recently established and is effective for calculating first-arrival travel times in an Earth model with an irregular surface. In previous work, the Lax–Friedrichs sweeping scheme used to approximate the TDEE viscosity solutions was only first-order accurate. We present a high-order fast-sweeping scheme to solve the TDEE with the aim of achieving high-order accuracy in the travel-time calculation. The scheme takes advantage of high-order weighted essentially nonoscillatory (WENO) derivative approximations, monotone numerical Hamiltonians, and Gauss Seidel iterations with alternating-direction sweepings. It incorporates high-order approximations of the derivatives into the numerical representation of the Hamiltonian such that the resulting numerical scheme is formally high-order accurate and inherits fast convergence from the alternating sweeping strategy. Extensive numerical examples are presented to verify its efficiency, convergence, and high-order accuracy.

Introduction

In seismic exploration, the problem of irregular topography makes the fine reconstruction of underground structures rather difficult at the crustal and, especially, the basin scales. In seismic exploration for oil and gas, seismologists encounter similar problems of undulating topography along the line of a survey. In processing seismic data, seismic travel time is a very important constraint in static correction (e.g., Lawton, 1989; Coppens, 2006), Kirchhoff migration (e.g., Gray and May, 1994; Symes et al., 1994), location of earthquakes (e.g., Kennett and Engdahl, 1991; Aki and Richards, 2002; Bouchon and Vallée, 2003; Doser, 2006), and seismic tomography (e.g., Hole and Zelt, 1992; Piromallo and Morelli, 1997; Doser et al., 1998; Zhang and Thurber, 2003; Badal et al., 2004; Zhao et al., 2004; Trampert and van der Hilst, 2005; Xu et al., 2006, 2010; Wang, 2011).

A major challenge for seismologists in these areas is the difficulty of obtaining a stable and accurate travel time. To date, methods for dealing with this problem are essentially based on unstructured grids (e.g., Fomel, 1997; Kimmel and Sethian, 1998; Sethian, 1999; Sethian and Vladimirsky, 2000; Alkhalifah and Fomel, 2001; Rawlinson and Sambridge, 2004; Qian et al., 2007a,b; Kao et al., 2008; Lelièvre et al., 2010). However, a number of wavefield modeling methods are based on structured grids (e.g., Fornberg, 1988; Tessmer et al., 1992; Hestholm and Rudd, 1994; Nielsen et al., 1994; Tessmer and Kosloff, 1994; Hestholm and Rudd, 1998; Zhang and Chen, 2006; Appelo and Petersson, 2009; Lan and Zhang, 2011a,b), making it important to calculate seismic travel times (for inversion and migration courses) on these grids. Recently, we developed a method for calculating first-arrival travel times by solving a topography-dependent eikonal equation (TDEE; Lan and Zhang, 2013), which takes advantage of a transformation from Cartesian to curvilinear coordinates. In this method, the Lax–Friedrichs sweeping scheme used to approximate the TDEE viscosity solutions only has first-order accuracy (Lan and Zhang, 2013). High-order accuracies are required to compute other quantities to an acceptable precision, such as the amplitude and the distribution of velocities (e.g., Cerveny et al., 1977; Qian and Symes, 2002b; Luo and Qian, 2011; Luo et al., 2011). We herein advance a high-order fast-sweeping scheme to solve the TDEE, integrating the advantages of high-order weighted essentially nonoscillatory (WENO) derivative approximations, monotone numerical Hamiltonians, and Gauss Seidel iterations with alternating-direction sweepings.

First, we briefly summarize the TDEE in a curvilinear coordinate system. Then, we describe the solutions of the TDEE (in a curvilinear coordinate system) using the high-order Lax–Friedrichs fast-sweeping method and present a number of numerical examples to demonstrate the accuracy and efficiency of our proposed method. Last, we draw some conclusions.

A Topography-Dependent Eikonal Equation in the Curvilinear Coordinate System

In the Cartesian coordinate system and for an isotropic medium, the eikonal equation in 2D can be written as
\[
\left( \frac{\partial T}{\partial x} \right)^2 + \left( \frac{\partial T}{\partial z} \right)^2 = S^2(x, z), \tag{1}
\]

where \(T(x, z)\) is the time of the first arrival of the seismic energy that propagates from a point source through a medium that has a slowness distribution of \(s(x, z)\).

To deal with irregular surfaces, we perform a transformation from the Cartesian coordinate system to the curvilinear coordinate system (see Appendix A for details). Using equation (A3) to replace the metric derivatives \(q_x, q_z\) into account, as demonstrated in a number of studies (e.g., Sethian and Vladimirsky, 2001; Alton and Mitchell, 2007a,b; Qian et al., 2007a,b). This expression gives the travel time \(T(x, z)\) of a ray that passes through a point \((x, z)\) in a medium with slowness \(s(x, z)\) and can be written in a more compact form such that

\[
\left( \frac{\partial q}{\partial \xi} \right)^2 T^2 + \frac{2}{\partial \eta} \left( \frac{\partial q}{\partial \xi} \right) + \left( \frac{\partial q}{\partial \eta} \right)^2 = s^2(q, r). \tag{2}
\]

Equation (3) is known as the TDEE (Lan and Zhang, 2013) and is a member of the Hamilton–Jacobi equations (e.g., Sethian and Vladimirsky, 2001; Qian et al., 2007a,b; Alton and Mitchell, 2008). This expression gives the travel time \(T(x, z)\) of a ray that passes through a point \((x, z)\) in a medium with slowness \(s(x, z)\) and can be written in a more compact form such that

\[
A \left( \frac{\partial T}{\partial q} \right)^2 + B \frac{\partial T}{\partial \eta} + C \left( \frac{\partial T}{\partial \eta} \right)^2 = s^2(q, r). \tag{4}
\]

where \(A = \frac{1}{2}(x_q^2 + z_q^2)\), \(B = -\frac{1}{2}(x_q x_q + z_q z_q)\), and \(C = \frac{1}{2}(x_q^2 + z_q^2)\). The parameters \(A, B, C\) are topography dependent. When the free surface is flat, \(x_q, z_q\) become zero, and \(s(x, z)\) reduces to the classical eikonal equation.

A High-Order Lax–Friedrichs Sweeping Scheme for Solving the Topography-Dependent Eikonal Equation

The TDEE is an elliptically anisotropic eikonal equation (e.g., Sethian and Vladimirsky, 2001; Qian et al., 2007a,b), although the medium is isotropic. Errors will occur in calculating travel times if the fast-marching method designed for isotropic eikonal equations are extended to anisotropic eikonal equations without taking the differences between them into account, as demonstrated in a number of studies (e.g., Sethian and Vladimirsky, 2001; Qian et al., 2007a,b; Alton and Mitchell, 2008).

A paraxial formulation establishing a relationship between the characteristic direction and the travel-time gradient direction has been proposed to tackle such anisotropic eikonal equations in seismic studies (e.g., Qian and Symes, 2001, 2002a,b, 2003). Almost at the same time, Sethian and Vladimirsky (2001, 2003) developed a so-called ordered upwind schemes for the static Hamilton–Jacobi equations. As an iterative method for Hamilton–Jacobi equations, the fast-sweeping method has been recently gaining attention. Tsai et al. (2003) applied the fast-sweeping method to a class of static Hamilton–Jacobi equations and derived explicit updated formulas so that the Gauss-Seidel-based sweeping strategy can be performed easily. Kao et al. (2004) extended the fast-sweeping method to approximate the viscosity solutions of arbitrary static Hamilton–Jacobi equations by use of the Lax–Friedrichs monotone numerical Hamiltonian. Zhang et al. (2006) extended the first-order Lax–Friedrichs fast-sweeping scheme to a higher order, based on WENO Schemes (Jiang and Shu, 1996; Jiang and Peng, 2000). Here, we introduce the high-order Lax–Friedrichs sweeping scheme (Zhang et al., 2006), as a substitute for the first-order method (Kao et al., 2004), to solve the TDEE for calculating the first-arrival travel times (Lan and Zhang, 2013).

In the following, we rearrange the TDEE into the Lax–Friedrichs Hamiltonian form, and then present the discretization formulas and the calculation steps for the solution of the TDEE using the Lax–Friedrichs fast-sweeping scheme.

Lax–Friedrichs Hamiltonian of the Topography-Dependent Eikonal Equation

Consider the rectangular domain \([q_{min}, q_{max}] \times [r_{min}, r_{max}]\) in the curvilinear coordinate system with a uniform discretization \((q_i, r_j), i = 1, 2, ..., m, j = 1, 2, ..., n\), where \(q_i = (i-1)h_q + q_{min}, r_j = (j-1)h_r + r_{min}\). The topography-dependent eikonal equation (3) or (4) is a special case of a static Hamilton–Jacobi equation and can be written as

\[
\begin{align*}
H(T_q, T_r) = s(q, r) & \quad (q, r) \in \Omega, \\
T(q, r) = g(q, r) & \quad (q, r) \in \Gamma \in \Omega, \tag{5}
\end{align*}
\]

where \(\Omega\) denotes the computational domain; \(\Gamma\) denotes source points within the computational domain; \(T_q = \frac{\partial T}{\partial q}\), \(T_r = \frac{\partial T}{\partial r}\). \(H(T_q, T_r) = \sqrt{A \left( \frac{\partial T}{\partial q} \right)^2 + B \frac{\partial T}{\partial \eta} + C \left( \frac{\partial T}{\partial \eta} \right)^2}\) is the Hamiltonian of equation (4); \(g(q, r)\) denotes initial values for the source; the meanings of \(A, B, C\) are the same as defined in equation (4); and \(s(x, z)\) is the slowness of the medium.

To discretize the Hamiltonian \(H(T_q, T_r)\), we use the Lax–Friedrichs numerical Hamiltonian, which is the simplest among all monotone numerical Hamiltonians, such that
\[ H^{LF}(u^-, u^+, v^-, v^+) = H\left(\frac{u^- + u^+}{2}, \frac{v^- + v^+}{2}\right) - \frac{1}{2} \alpha'(v^+ - v^-), \]

where \( u = \partial T / \partial x; v = \partial T / \partial z; u^\pm \) and \( v^\pm \) are the forward and backward difference approximations of \( \partial T / \partial x \) and \( \partial T / \partial z \), respectively; and \( \alpha_q \) and \( \alpha_r \) are artificial viscosities that satisfy

\[
\alpha_q \geq \max_{\frac{\partial Q}{\partial r} \neq 0} |H_1(u, v)|, \quad \alpha_r \geq \max_{\frac{\partial Q}{\partial r} \neq 0} |H_2(u, v)|.
\]

Here, \( H_1(u, v) \) is the partial derivative of \( H \) with respect to the \( i \)th argument, \([D, E]\) is the value range for \( u^\pm \), and \([F, K]\) is the value range for \( v^\pm \).

Kao et al. (2004) constructed a first-order Lax–Friedrichs sweeping scheme for general static Hamilton–Jacobi equations, which we used to solve the topography-dependent eikonal equation (5):

\[
T_{i,j}^{n+1} = \left( \frac{1}{\alpha_q \sqrt{h_q}} + \frac{\alpha_r}{\sqrt{h_r}} \right) \left[ s_{i,j} - H \left( \frac{T_{i+1,j} - T_{i-1,j}}{2h_q}, \frac{T_{i,j+1} - T_{i,j-1}}{2h_r} \right) \right] \]
\[+ \alpha_q \frac{T_{i+1,j} + T_{i-1,j}}{2h_q} + \alpha_r \frac{T_{i,j+1} + T_{i,j-1}}{2h_r} \right]. \tag{7}
\]

The superscripts of \( T \) are not included because they depend on the sweeping directions of the alternating Gauss Seidel type iterations. We always use the most recent value for \( T \) in the interpolation stencils according to the Gauss Seidel type iteration.

To construct a high-order scheme, we need to approximate the derivatives \( T_q \) and \( T_r \) with high-order accuracy. We use the popular WENO approximations, developed by Zhang et al. (2006). To illustrate the feasibility of this approach, we take the third order rather than the fifth order WENO approximations; one certainly can replace the third-order WENO approximations for the derivatives \( T_q \) and \( T_r \), while \( T_q \) can be considered as an approximation to \( T_{i,j} \), \( T_{i+1,j} \), and \( T_{i,j+1} \) can be considered as approximations to \( T_{i,j-1} \) and \( T_{i,j+1} \), respectively. Replacing \( T_{i,j-1} \), \( T_{i+1,j} \), \( T_{i,j+1} \), and \( T_{i,j+1} \) with these approximations in equation (7), we derive the following high-order scheme, such that

\[
T_{i,j}^{n+1} = \left( \frac{1}{\alpha_q \sqrt{h_q}} + \frac{\alpha_r}{\sqrt{h_r}} \right) \left[ s_{i,j} - H \left( \frac{T_q^i_{i,j} + (T_r^i_{i,j})^i_{i,j}}{2}, \frac{(T_r^i_{i,j})^i_{i,j} + (T_r^i_{i,j})^i_{i,j}}{2} \right) \right] \]
\[+ \alpha_q \frac{2T_q^i_{i,j} + h_q(T_q^i_{i,j})^i_{i,j}}{2h_q} \]
\[+ \alpha_r \frac{2T_r^i_{i,j} + h_r(T_r^i_{i,j})^i_{i,j}}{2h_r} \right]. \tag{12}
\]

which may be written more compactly as

\[
T_{i,j}^{n+1} = \left( \frac{1}{\alpha_q \sqrt{h_q}} + \frac{\alpha_r}{\sqrt{h_r}} \right) \left[ s_{i,j} - H \left( \frac{T_q^i_{i,j} + (T_r^i_{i,j})^i_{i,j}}{2}, \frac{(T_r^i_{i,j})^i_{i,j} + (T_r^i_{i,j})^i_{i,j}}{2} \right) \right] \]
\[+ \alpha_q \frac{T_q^i_{i,j} - (T_r^i_{i,j})^i_{i,j}}{2h_q} + \alpha_r \frac{T_r^i_{i,j} - (T_r^i_{i,j})^i_{i,j}}{2h_r} \right] + T_{i,j}^{old}, \tag{13}
\]

where \( T_{i,j}^{old} \) denotes the to-be-updated numerical solution for \( T \) at the grid point \( (i, j) \) and \( T_{i,j}^{old} \) denotes the current (old) value for \( T \) at the same grid point. When the WENO approximations for the derivatives \( T_q^i_{i,j}, T_q^i_{i,j}, T_r^i_{i,j}, T_r^i_{i,j}, \) and \( T_r^i_{i,j} \) in equation (13) are calculated using equations (8)–(11), we always use the newest value for \( T \) in the interpolation stencils, according to the Gauss–Seidel type iteration. Of
course, because we did not update $T_{i,j}$ yet, $T_{i,j}^{\text{old}}$ is used in equations (8)–(11).

Steps for Calculating the First-Arrival Travel Times Using the Lax–Friedrichs Fast-Sweeping Scheme

The Lax–Friedrichs Hamiltonian gives a solution dependent on all its neighbors in all dimensions, so we need to carefully specify the values of points outside the computational domain, otherwise huge errors will be introduced for the points on the computational boundary and then propagate into the computational domain. Here, we impose the following conditions on the four sides of the boundary to combine extrapolation, maximization, and minimization and to calculate the values for points outside the computational domain, as proposed by Kao et al. (2004):

\[
\begin{align*}
T_{i,j}^{\text{new}} &= \min(\max(2T_{i+1,j} - T_{i,j}), T_{i,j}^{\text{old}}), \\
T_{i+1,j}^{\text{new}} &= \min(\max(2T_{i,j+1} - T_{i,j}), T_{i,j}^{\text{old}}), \\
T_{i,j+1}^{\text{new}} &= \min(\max(2T_{i,j+1} - T_{i+1,j}), T_{i,j}^{\text{old}}), \\
T_{i,m+1,j}^{\text{new}} &= \min(\max(2T_{i,j} - T_{i,m+1,j}), T_{i,j}^{\text{old}}),
\end{align*}
\]

where $i = -1, 0$ and $j = -1, 0$.

The Lax–Friedrichs fast-sweeping method for equation (5) consists of four steps that include (1) initialization; (2) alternating sweeps; (3) enforcing computational boundary conditions; and (4) convergence test. These steps can be summarized as follows:

- **Initialization.** Exact or interpolated values $T_{i,j}^0$ are assigned at grid points on or near the source $\Gamma$, and these values are fixed throughout the iterations. Large positive values are assigned at all other grid points; these should be larger than the maximum value of the true solutions, and they are updated in the iterations.

- **Alternating sweepings.** At iteration $n + 1$, we calculate $T_{i,j}^{n+1}$ according to equation (13) at all grid points $(q_i, r_j)$, $1 \leq i \leq m_1$, and $1 \leq j \leq m_2$, except for those that have assigned values, and update $T_{i,j}^{n+1}$ only when its value is smaller than its previous value $T_{i,j}^n$. As stated above, this process needs to be performed in alternating sweeping directions, which means that it needs four different sweeps in the 2D case: (a) from lower left to upper right $i = 1 : m_1$, $j = 1 : m_2$; (b) from lower right to upper left $i = m_1 : 1$, $j = 1 : m_2$; (c) from upper left to lower right $i = 1 : m_1$, $j = m_2 : 1$; and (d) from upper right to lower left $i = m_1 : 1$, $j = m_2 : 1$. In general, in the $d$-dimension case, we need $2^d$ alternating sweeps.

- **Enforcing computational boundary conditions.** After each sweep, we enforce computational boundary conditions according to equation (14), trivially modified depending on the boundary we are using.

Figure 1. A stencil of the third-order WENO scheme, $\varphi = q, r$.

- **Convergence test.** Given the convergence criterion $\delta$, the algorithm converges and stops if $\|T^{n+1} - T^n\|_{L_1} \leq \delta$.

Accuracy and Efficiency Tests

We perform three groups of numerical experiments, with increasing topographical complexity, to test the accuracy and efficiency of our proposed method for the computation of first-arrival travel times in the presence of surface topography. Initially, we consider a group of two models with a flat surface: the first is a homogeneous medium, and the second is the SEG/EAGE Marmousi model, which is used to test the method on media with strong lateral/vertical velocity variations. The second group includes a couple of models with cosine-curved surfaces and a source located at a different depth in each model. The third group comprises a model with a much more complex surface topography. We compute the analytical solutions (see Appendix B for details) and measure the errors under grid refinement in each case to explore the convergence of our numerical solutions. In the following examples, we consider the iteration convergent if the L1 error of the difference between two successive iterations $\|T^{n+1} - T^n\|_{L_1}$ is less than $10^{-6}$. The formula for the L1 error (Qian and Symes, 2002a) is

\[
L1(t) = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \frac{|T_{\text{ana}} - t_n|}{m_1 m_2},
\]

where $m_1$ and $m_2$ are grid point quantities in the horizontal and vertical directions, respectively; $t_{\text{ana}}$ is the travel time from the analytical method; and $t_n$ is the travel time from the finite-difference scheme. A single iteration includes four alternating sweepings for 2D problems. Generally, the algorithm converges within a few hundred iterations. The velocity is 2000 m/s in all of the homogeneous models presented in this paper. Initial values are assigned in a circle, which includes the source point, and the radius of the circle is $2h$ where $h$ is the mesh size (i.e., the number of grid points.
Table 1
L1 Errors and Iteration Numbers for Numerical Results Calculated by First- and High-Order Lax–Friedrichs Sweeping Schemes for the Homogeneous Flat-Surface Model, with a Source Located at (0.8 km, 1.1 km)

<table>
<thead>
<tr>
<th>Mesh</th>
<th>L1 Error (First Order)</th>
<th>L1 Error (High Order)</th>
<th>Iteration Number (First Order)</th>
<th>Iteration Number (High Order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>160 × 120</td>
<td>1.0087 × 10^{-2}</td>
<td>8.6768 × 10^{-3}</td>
<td>38</td>
<td>64</td>
</tr>
<tr>
<td>320 × 240</td>
<td>5.9143 × 10^{-2}</td>
<td>5.4398 × 10^{-3}</td>
<td>64</td>
<td>117</td>
</tr>
<tr>
<td>640 × 480</td>
<td>3.3920 × 10^{-3}</td>
<td>3.1793 × 10^{-3}</td>
<td>113</td>
<td>222</td>
</tr>
</tbody>
</table>

Table 2
L1 Errors and Iteration Numbers for the First- and High-Order Lax–Friedrichs Sweeping Schemes in the Marmousi Model with a Source Located at (0.8 km, 1.1 km)

<table>
<thead>
<tr>
<th>Mesh</th>
<th>L1 Error (First Order)</th>
<th>L1 Error (High Order)</th>
<th>Iteration Number (First Order)</th>
<th>Iteration Number (High Order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>160 × 120</td>
<td>1.8107 × 10^{-2}</td>
<td>4.3745 × 10^{-3}</td>
<td>40</td>
<td>74</td>
</tr>
<tr>
<td>320 × 240</td>
<td>1.0973 × 10^{-2}</td>
<td>1.4187 × 10^{-3}</td>
<td>67</td>
<td>130</td>
</tr>
<tr>
<td>640 × 480</td>
<td>6.2147 × 10^{-3}</td>
<td>5.3735 × 10^{-4}</td>
<td>118</td>
<td>239</td>
</tr>
</tbody>
</table>

The surface of the model used is described by $z(x) = 1.0 + 0.2 \cos(1.5\pi x) \text{km}$, $x \in [-1 \text{ km}, 1 \text{ km}]$, representing topography with combinations of one hill and two depressions. The computational domain extends to a depth of $-1 \text{ km}$. The $100 \times 100$ mesh boundary-conforming grids are shown in Figure 3. We focus on the performance of the proposed method for the first-arrival travel-time calculation in a model with more complicated topography and seismic sources at different depths. We place the seismic source at six different depths ($\eta = 1.1, 0.7, 0.3, -0.1, -0.5$, and $-0.9 \text{ km}$), and the source is located at $(-0.2 \text{ km}, \eta)$. We compute the analytical solutions in each case (see Appendix B for details) and measure the errors between the calculations and the source is located at (0.8 km, 1.1 km) for both models. The computed solution, while the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes, respectively. For Group 1 models, (a) homogeneous medium model (b) Marmousi model. The source is at (0.8 km, 1.1 km) for both models. The computations are performed on the 160 × 120 mesh for (a) and the 640 × 480 mesh for (b). The black solid line represents the analytical solution, while the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes, respectively.

Figure 2. Travel-time contours (in seconds) for the group 1 models, (a) homogeneous medium model (b) Marmousi model. The source is located at (0.8 km, 1.1 km) for both models. The computations are performed on the 160 × 120 mesh for (a) and the 640 × 480 mesh for (b). The black solid line represents the analytical solution, while the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes, respectively.

Group 1

A flat-surface model with the size of 1.6 km × 1.2 km is considered here. We discretize the model using three mesh sizes: 160 × 120, 320 × 240, and 640 × 480. The source is located at coordinate (0.8 km, 1.1 km). We calculate first-arrival travel times using the low- and high-order Lax–Friedrichs fast-sweeping schemes, respectively, and compare the results with the exact solutions. To save space here, only the travel-time contours of the numerical and analytical solutions on the 160 × 120 mesh are shown (Fig. 2a), and comparisons of the L1 errors between the numerical and exact solutions for the different mesh sizes are summarized in Table 1. The L1 errors of the high-order scheme are hundreds of times smaller than those of the low-order method, and only about twice as many iterations are required.

Next, we use the SEG/EAGE Marmousi model to test the method on media with strong lateral/vertical velocity variations. Because the analytical solution is unavailable, we use the results calculated by the fast-sweeping method on a finer (2560 × 1920) mesh to approximate the analytical solution. The L1 errors for the full domain are reported in Table 2. The high-order method clearly achieves much better accuracy than the first-order scheme on the same mesh. The numerical and analytical solution travel-time contours on the 640 × 480 mesh (see Fig. 2b) show good agreement between the analytical and high-order scheme solutions, but the results from the first-order scheme reveal remarkable errors.

Group 2

The surface of the model used is described by $z(x) = 1.0 + 0.2 \cos(1.5\pi x) \text{km}$, $x \in [-1 \text{ km}, 1 \text{ km}]$, representing topography with combinations of one hill and two depressions. The computational domain extends to a depth of $-1 \text{ km}$. The $100 \times 100$ mesh boundary-conforming grids are shown in Figure 3. We focus on the performance of the proposed method for the first-arrival travel-time calculation in a model with more complicated topography and seismic sources at different depths. We place the seismic source at six different depths ($\eta = 1.1, 0.7, 0.3, -0.1, -0.5$, and $-0.9 \text{ km}$), and the source is located at $(-0.2 \text{ km}, \eta)$. We compute the analytical solutions in each case (see Appendix B for details) and measure the errors between the calculations and the source is located at (0.8 km, 1.1 km) for both models. The computed solution, while the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes, respectively. For Group 1 models, (a) homogeneous medium model (b) Marmousi model. The source is at (0.8 km, 1.1 km) for both models. The computations are performed on the 160 × 120 mesh for (a) and the 640 × 480 mesh for (b). The black solid line represents the analytical solution, while the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes, respectively.

Figure 2. Travel-time contours (in seconds) for the group 1 models, (a) homogeneous medium model (b) Marmousi model. The source is located at (0.8 km, 1.1 km) for both models. The computations are performed on the 160 × 120 mesh for (a) and the 640 × 480 mesh for (b). The black solid line represents the analytical solution, while the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes, respectively.

is fixed in the initialization throughout the mesh-refinement study).
using the Lax–Friedrichs fast-sweeping schemes and the exact solutions under grid refinement. The L1 errors of the first- and high-order Lax–Friedrichs sweeping schemes and the respective iteration numbers are shown in Table 3. The high-order scheme gives much smaller errors than does the low-order method on the same grids.

Figure 4a–f shows the travel-time contours for each test. The numerical solutions of the high-order method on the $100 \times 100$ mesh match very well with the analytical solutions for each case and are much better than the results from the first-order method on a finer mesh ($800 \times 800$). The results of the first-order method on the sparse mesh ($100 \times 100$) reveal notable errors, which should not be used directly in light of this consideration. Figure 5a–f displays the distributions obtained by comparing travel times calculated by the high-order Lax–Friedrichs sweeping scheme and the analytical solutions on the $100 \times 100$ mesh for each test. The errors are larger where the curvature of the topography is greatest. Figure 6a–f shows a comparison of the numerical and analytical solutions of travel times recorded at the irregular surface for each test, which confirms our conclusions.

Group 3

To test the proposed scheme further, we consider another model with a much more complex surface topography.

![Figure 3](image-url)

**Figure 3.** The $100 \times 100$ mesh boundary-conforming grids for the group 2 models. The inverted black triangles represent the receivers located at the irregular surface, with horizontal distances between $-1$ and $1$ km.

![Table 3](image-url)

<table>
<thead>
<tr>
<th>Source Location</th>
<th>Mesh</th>
<th>L1 Error (First Order)</th>
<th>L1 Error (High Order)</th>
<th>Iteration Number (First Order)</th>
<th>Iteration Number (High Order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(-0.2, 1.1)$</td>
<td>$100 \times 100$</td>
<td>$2.9857 \times 10^{-2}$</td>
<td>$3.1788 \times 10^{-3}$</td>
<td>56</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>$200 \times 200$</td>
<td>$1.9682 \times 10^{-2}$</td>
<td>$1.4779 \times 10^{-3}$</td>
<td>93</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>$400 \times 400$</td>
<td>$1.1723 \times 10^{-2}$</td>
<td>$5.3604 \times 10^{-4}$</td>
<td>161</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>$800 \times 800$</td>
<td>$6.6531 \times 10^{-3}$</td>
<td>—</td>
<td>290</td>
<td>—</td>
</tr>
<tr>
<td>$(-0.2, 0.7)$</td>
<td>$100 \times 100$</td>
<td>$2.1935 \times 10^{-2}$</td>
<td>$1.6522 \times 10^{-3}$</td>
<td>44</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>$200 \times 200$</td>
<td>$1.3471 \times 10^{-2}$</td>
<td>$6.6823 \times 10^{-4}$</td>
<td>74</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>$400 \times 400$</td>
<td>$8.1473 \times 10^{-3}$</td>
<td>$1.6854 \times 10^{-4}$</td>
<td>130</td>
<td>228</td>
</tr>
<tr>
<td></td>
<td>$800 \times 800$</td>
<td>$4.5590 \times 10^{-3}$</td>
<td>—</td>
<td>238</td>
<td>—</td>
</tr>
<tr>
<td>$(-0.2, 0.3)$</td>
<td>$100 \times 100$</td>
<td>$1.9476 \times 10^{-2}$</td>
<td>$9.2037 \times 10^{-4}$</td>
<td>39</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>$200 \times 200$</td>
<td>$1.2158 \times 10^{-2}$</td>
<td>$4.1163 \times 10^{-4}$</td>
<td>64</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>$400 \times 400$</td>
<td>$7.1047 \times 10^{-3}$</td>
<td>$6.7634 \times 10^{-5}$</td>
<td>112</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>$800 \times 800$</td>
<td>$4.0577 \times 10^{-3}$</td>
<td>—</td>
<td>203</td>
<td>—</td>
</tr>
<tr>
<td>$(-0.2, -0.1)$</td>
<td>$100 \times 100$</td>
<td>$2.0109 \times 10^{-2}$</td>
<td>$1.1333 \times 10^{-4}$</td>
<td>38</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>$200 \times 200$</td>
<td>$1.2047 \times 10^{-2}$</td>
<td>$1.6507 \times 10^{-5}$</td>
<td>61</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>$400 \times 400$</td>
<td>$6.9463 \times 10^{-3}$</td>
<td>$3.9539 \times 10^{-6}$</td>
<td>104</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>$800 \times 800$</td>
<td>$4.0412 \times 10^{-3}$</td>
<td>—</td>
<td>185</td>
<td>—</td>
</tr>
<tr>
<td>$(-0.2, -0.5)$</td>
<td>$100 \times 100$</td>
<td>$2.0161 \times 10^{-2}$</td>
<td>$1.2355 \times 10^{-3}$</td>
<td>43</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>$200 \times 200$</td>
<td>$1.2622 \times 10^{-2}$</td>
<td>$4.1841 \times 10^{-4}$</td>
<td>71</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>$400 \times 400$</td>
<td>$7.2589 \times 10^{-3}$</td>
<td>$4.7859 \times 10^{-5}$</td>
<td>121</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>$800 \times 800$</td>
<td>$4.1329 \times 10^{-3}$</td>
<td>—</td>
<td>218</td>
<td>—</td>
</tr>
<tr>
<td>$(-0.2, -0.9)$</td>
<td>$100 \times 100$</td>
<td>$2.7176 \times 10^{-2}$</td>
<td>$1.0876 \times 10^{-3}$</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>$200 \times 200$</td>
<td>$1.3382 \times 10^{-2}$</td>
<td>$4.3949 \times 10^{-4}$</td>
<td>81</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>$400 \times 400$</td>
<td>$7.6445 \times 10^{-3}$</td>
<td>$4.8919 \times 10^{-5}$</td>
<td>142</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>$800 \times 800$</td>
<td>$4.4423 \times 10^{-3}$</td>
<td>—</td>
<td>256</td>
<td>—</td>
</tr>
</tbody>
</table>
Figure 4. Travel-time contours (in seconds) for the group 2 models with different source depths: (a) (−0.2 km, 1.1 km), (b) (−0.2 km, 0.7 km), (c) (−0.2 km, 0.3 km), (d) (−0.2 km, −0.1 km), (e) (−0.2 km, −0.5 km), (f) (−0.2 km, −0.9 km). The black solid line represents the analytical solution, the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes on the 100 × 100 mesh, and the green dashed line represents the first-order Lax–Friedrichs sweeping scheme numerical solutions on the 800 × 800 mesh.

Figure 5. Absolute differences, or errors, (in seconds) between travel times calculated using the high-order Lax–Friedrichs sweeping scheme and the analytical solutions on the 100 × 100 mesh for the group 2 models with different source depths, (a) (−0.2 km, 1.1 km), (b) (−0.2 km, 0.7 km), (c) (−0.2 km, 0.3 km), (d) (−0.2 km, −0.1 km), (e) (−0.2 km, −0.5 km), and (f) (−0.2 km, −0.9 km).
The size of the model is $1.6 \text{ km} \times 1.32 \text{ km}$. The surface undulation is between $1.1 \text{ km}$ and $1.32 \text{ km}$ and represents topographies with combinations of two hills and two depressions. We discretize the model with meshes of $160 \times 120$, $320 \times 240$, $640 \times 480$, and $1280 \times 960$ (the last is used only to solve the first-order scheme). Figure 7 shows the boundary-conforming grids of the $160 \times 120$ mesh. The source is located at $(0.8 \text{ km}, 1.1 \text{ km})$. We test the accuracy of the numerical methods by comparing them with the analytical solutions (see Appendix B for details) under the mesh refinement, and the results are shown in Table 4. The contour plots of the numerical solutions and the analytical results are shown in Figure 8. Figure 9 shows the error distributions for the high-order scheme on the $160 \times 120$ mesh. Comparing the numerical and analytical solutions of travel times recorded at the irregular surface (Fig. 10) provide further support to our conclusions.

Concluding Remarks

We have presented a TDEE in a curvilinear coordinate system that uses a boundary-conforming grid and maps a rectangular grid onto a curved one. We use a high-order fast-sweeping method to solve the TDEE in the curvilinear coordinate system. A general procedure is given to incorporate the high-order approximations into monotonic numerical Hamiltonians so that the first-order sweeping scheme can be extended to a high-order scheme. Extensive numerical

---

**Figure 6.** Comparison of analytical and numerical solution travel times at the irregular surface for the group 2 models with different source depths, (a) ($-0.2 \text{ km}, 1.1 \text{ km}$), (b) ($-0.2 \text{ km}, 0.7 \text{ km}$), (c) ($-0.2 \text{ km}, 0.3 \text{ km}$), (d) ($-0.2 \text{ km}, -0.1 \text{ km}$), (e) ($-0.2 \text{ km}, -0.5 \text{ km}$), and (f) ($-0.2 \text{ km}, -0.9 \text{ km}$). The black solid line represents the analytical solution, the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes on the $100 \times 100$ mesh, and the green dashed line represents the first-order Lax–Friedrichs sweeping scheme numerical solutions on the $800 \times 800$ mesh.
examples demonstrate that the high-order method yields high-order accuracy in the solution and fast convergence to viscosity solutions of the topography-dependent equation, which will be helpful in the real-world processing of seismic data acquired at irregular surfaces, for static correction, pre-stack migration, and tomography.

Data and Resources

All data used in this study are obtained from published sources listed in the references.

Table 4
L1 Errors and Iteration Numbers for the First- and High-order Lax–Friedrichs Sweeping Schemes in the Group 3 Model with a Source Located at (0.8 km, 1.1 km)

<table>
<thead>
<tr>
<th>Mesh</th>
<th>L1 Error (First Order)</th>
<th>L1 Error (High Order)</th>
<th>Iteration Number (First Order)</th>
<th>Iteration Number (High Order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>160 × 120</td>
<td>1.1560 × 10^{-2}</td>
<td>4.5952 × 10^{-4}</td>
<td>48</td>
<td>65</td>
</tr>
<tr>
<td>320 × 240</td>
<td>6.9548 × 10^{-3}</td>
<td>1.0930 × 10^{-4}</td>
<td>82</td>
<td>119</td>
</tr>
<tr>
<td>640 × 480</td>
<td>3.9093 × 10^{-3}</td>
<td>2.1213 × 10^{-5}</td>
<td>146</td>
<td>224</td>
</tr>
<tr>
<td>1280 × 960</td>
<td>2.2930 × 10^{-3}</td>
<td>—</td>
<td>288</td>
<td>—</td>
</tr>
</tbody>
</table>

Figure 8. Travel-time contours (in seconds) for the group 3 model with the source at (0.8 km, 1.1 km). The solid black line represents the analytical solution, the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes on the 160 × 120 mesh, and the green dashed line represents the numerical solutions of the first-order Lax–Friedrichs sweeping scheme on the 1280 × 960 mesh.

Figure 9. Absolute differences, or errors (in seconds) between travel times calculated using the high-order Lax–Friedrichs sweeping scheme and the analytical solutions on the 160 × 120 mesh for the group 3 model.

Acknowledgments

We would like to thank Editor in Chief Diane I. Doser, Associate Editor Michel Bouchon, and two anonymous referees for their constructive comments and helpful suggestions, which led to a significant improvement of the early manuscript. Financial support from the Chinese Academy of Sciences (KZCX2-YW-132), the National Natural Science Foundation of China (41074033, 40721003, 40830315, and 40874041), the Important National Science and Technology Specific Projects (2008ZX05008-006), and the Ministry of Science and Technology of China (SINOPROBE-02-02) are gratefully acknowledged.
Figure 10. Comparison of analytical and numerical solution travel times at the irregular surface for the group 3 model with the source located at (0.8 km, 1.1 km). The black solid line represents the analytical solution, the blue and red dashed lines represent the numerical solutions obtained using the first- and high-order Lax–Friedrichs sweeping schemes on the 160 × 120 mesh, and the green dashed line represents the numerical solutions of the first-order Lax–Friedrichs sweeping scheme on the 1280 × 960 mesh.

References


**Appendix A**

**Transformation from Cartesian to Curvilinear Coordinates**

For a given topographic surface, the discrete grid should conform to the free surface to suppress artificial errors. Such a grid is termed a boundary-conforming grid (Thompson *et al.*, 1985; Hvid, 1994), and has been used by a number of researchers in the simulation of seismic-wave fields (Fornberg, 1988; Zhang and Chen, 2006; Appelo and Petersson, 2009; Lan and Zhang, 2011a,b). A grid of this type may be formed by carrying out a transformation from a (curvilinear) computational space to a (Cartesian) physical space, as shown in Figure A1. Under such a transformation, the curvilinear coordinates \(q\) and \(r\) are mapped to Cartesian coordinates within the physical space, with both systems including a positive upward direction for the vertical coordinate.

After generating the boundary-conforming grid, the Cartesian coordinates of each grid point may be determined from the curvilinear coordinates using the following equations:

\[
x = x(q, r) \quad \text{and} \quad z = z(q, r).
\]

We then can express the spatial derivatives in the Cartesian coordinate system \((x, z)\) from the curvilinear coordinate system \((q, r)\) following the chain rule, such that

\[
\partial_x = q_x \partial_q + r_x \partial_r, \quad \text{and} \quad \partial_z = q_z \partial_q + r_z \partial_r,
\]

where \(q_x\) denotes \(\partial q(x, z)/\partial x\), and similar relationships of this type apply in other cases. These derivatives are known as metric derivatives or simply metrics. We can also find the derivatives of the metrics

\[
q_x = \frac{z_r}{J}, \quad q_z = -\frac{x_r}{J}, \quad r_x = -\frac{z_q}{J}, \quad \text{and} \quad r_z = \frac{x_q}{J},
\]

where \(J\) is the Jacobian determinant of the transformation, which can be written as \(J = x_q z_r - x_r z_q\).

It is worth noting that even if the mapping equations (A1) are expressed as an analytical function, the derivatives should still be calculated numerically to avoid spurious source terms that may be caused by the derivative coefficients when the conservative forms of the equations are used (Thompson *et al.*, 1985). In all of the examples presented here, the metric derivatives are computed numerically using second-order accurate finite-difference approximations.

**Figure A1.** Mapping between computational and physical space in two dimensions.
Appendix B

Strategies for Computing the Travel-Time Analytical Solutions for the Homogeneous Models with Irregular Surfaces Presented in This Paper

Analytical Solutions for the Group 2 Models

For the models in group 2, we first consider the solutions for the source at (−0.2 km, 1.1 km) (Fig. B1a), solutions for other cases are discussed later. Computing analytical solutions for a cosinusoid surface is not as easy as for a flat surface. The challenge in this context is the difficulty in seeking the shortest paths from the source to other points within the domain. According to the paths of the rays emitted from point $S$, the domain can be divided into three parts: the upper left part (surrounded by the arc $DM$, and the line segments $ME$ and $ED$), the upper right part (surrounded by the arc $GN$, and the line segments $NF$ and $FG$), and the remaining part. The rays reach the former two parts with bent paths but get to the last part with straight paths. The first-arrival travel times for the third part can be calculated easily; that is, the analytical solutions are simply the Euclidean distances divided by the velocity in the medium. However, for the upper left and right parts, the first-arrival travel times cannot be computed so easily. The left part is taken as an example in the following (it is easy to extend this strategy to the right part).

For points beneath the surface, take point $B$ as an example, the shortest path for the ray from $S$ to $B$ is $SD + DC + CB$, where $SD$ and $CB$ are tangents to the cosine curve.
(irregular surface) at points D and C, respectively, and arc DC is part of the cosine curve. This path can be demonstrated to be the shortest path by comparing it with others, but that is not included here. After finding the shortest path, it is easy to compute its length. The lengths of vectors SD and CB can be obtained using the distance formula between the two points, while the length of DC can be calculated by integrating the arc between the points D and C.

Both arcs MD and NG are strictly concave, so, for points on these two arcs, the shortest path consists of an arc and a line segment. Take the point A as an example, the shortest path for the rays emitted from point S arriving at this point is SD + DA, the length of which can be calculated using the method mentioned above.

When the source is located at (−0.2 km, 0.7 km) or (−0.2 km, 0.3 km) (Fig. B1b and c, respectively), the solution is similar to the case analyzed above. For the source located at (−0.2 km, −0.1 km) (Fig. B1d), only a fraction on the upper right of the domain needs to be treated using the methods described above; for other parts, including when the source is at (−0.2 km, −0.5 km) or (−0.2 km, −0.9 km) (Fig. B1e and f, respectively), the analytical solutions are simply the Euclidean distances divided by the velocity.

Analytical Solutions for the Group 3 Model

Generally speaking, and for this model (Fig. B2), we can calculate the first-arrival travel times in a similar way to the method described above. To make the solutions easier, we put the source at a special point, the x coordinate of the point is at the center of the x axis, while the y coordinate is at the same height as the lowest point of the complex surface (D and G). Except for the two shaded regions, travel times for other parts can be obtained directly. Travel times in the two shaded regions can be calculated in a similar way to the method described above, but with some differences. DM and GN are two complex curves that consist of a convex curve (CM or BN) and a concave curve (DC or GB), respectively. For a point on CM or BN, there should be a line through the point tangential to DC or GB. Furthermore, the shortest path from S to the point on CM or BN consists of three parts (two segments and an arc), while for the point on DC or GB, the shortest path comprises only two parts (a segment and an arc, as in group 2). For the other shaded regions, the travel times can be computed as described above.

State Key Laboratory of Lithospheric Evolution
Institute of Geology and Geophysics
Chinese Academy of Sciences
No. 19 Beitucheng Western Road 9825#
Chaoyang District
Beijing 100029, China
lanhq@mail.iggcas.ac.cn

Manuscript received 9 June 2012