Commun. Comput. Phys.
doi: 10.4208/cicp.130312.301012a

Vol. **14**, No. 2, pp. 509-536
August 2013

Numerical Boundary Conditions for Specular Reflection in a Level-Sets-Based Wavefront Propagation Method

Sheri L. Martinelli[1,2,*]

[1] *Torpedo Systems Department, Naval Undersea Warfare Center,
1176 Howell Street, Newport, Rhode Island 02841, USA.*
[2] *Division of Applied Mathematics, Brown University, Providence,
Rhode Island 02912, USA.*

Received 13 March 2012; Accepted (in revised version) 30 October 2012

Communicated by Lianjie Huang

Available online 4 January 2013

Abstract. We study the simulation of specular reflection in a level set method implementation for wavefront propagation in high frequency acoustics using WENO spatial operators. To implement WENO efficiently and maintain convergence rate, a rectangular grid is used over the physical space. When the physical domain does not conform to the rectangular grid, appropriate boundary conditions to represent reflection must be derived to apply at grid locations that are not coincident with the reflecting boundary. A related problem is the extraction of the normal vectors to the boundary, which are required to formulate the reflection condition. A separate level set method is applied to pre-compute the boundary normals which are then stored for use in the wavefront method. Two approaches to handling the reflection boundary condition are proposed and studied: one uses an approximation to the boundary location, and the other uses a local reflection principle. The second method is shown to produce superior results.

AMS subject classifications: 65M06, 65M22, 65M25

PACS: 43.30.Gv, 43.30.Zk

Key words: Boundary conditions, reflection, level set method, wavefront methods, high-frequency acoustic propagation, WENO.

1 Introduction

1.1 Motivation

High frequency wave propagation models are of vital importance to understanding the propagation of sound in shallow water environments. Applications such as acoustic tomography [1] and underwater communications [2] rely on ray tracing for system design

[*]Corresponding author. *Email address:* `sheri_martinelli@alumni.brown.edu` (S. Martinelli)

and performance prediction. The frequencies useful for such applications in shallow water are at least on the order of 1kHz and can range into the MHz regime, rendering full wave models impractical in simulations. On the other hand, ray tracing has many known limitations. In particular, the Lagrangian nature of the ray trace model leads to difficulty resolving wave arrivals at a fixed location in space. This is especially true in range-dependent, shallow water environments. The goal of this work is to devise stable, accurate, numerical boundary conditions for reflection in a level set method implementation that solves the high frequency equation for the acoustic phase, thus tracing entire wavefronts on a fixed grid in space. A practical wavefront model for shallow water acoustics would allow for improved accuracy in such harsh environments, where multipath propagation is evident due to multiple reflections off of the sea surface and bottom. Such a model would also be more robust to perturbations in material properties. For example, Godin [3] studied the effect of small perturbations in the sound speed on rays, showing that the resulting displacement of rays *across* the wavefront are much smaller than displacements *along* the wavefront. This type of relative stability of wavefronts is critical to practical simulations due to the uncertainty present in sound speed profiles derived from at-sea data collection.

This study is based on a method laid out in [4], introducing an implementation of the level set method intended for a two-dimensional shallow water acoustics application. The algorithm is based on the level set method for geometric optics [5]. The level set equations are solved using a fifth order Weighted Essentially Non-Oscillatory (WENO) method for the spatial operator [6] combined with third order Total Variation Diminishing Runge-Kutta (TVDRK) time integration [7] for stability. This choice is motivated by the presence of a discontinuity in the phase space at a reflecting boundary [4]. While the discontinuity is a consequence of the rectangular scatterers presented in that work, this is not the case in general, but even in the benign case of a flat reflecting boundary, a sharp cusp forms at the boundary in the reduced phase space. The examples presented in [4] are limited by the types of domain geometry that could be handled under the reflection boundary condition to either grid-conforming rectangular domains, or a domain with a linear boundary. For the linear boundary case, the boundary did not conform to the grid and an approximation was applied in order to provide the required boundary conditions for the numerical solver. This approach is discussed in greater detail in Section 2.3.1. The goal of the present work is to extend the method to accommodate more realistic domain geometry, and improve the accuracy of the reflection boundary condition. To achieve this, one could modify the rectangular grid to conform to the geometry, as in [8], however in the general case, this affects the convergence of WENO. Refining the grid to match the geometry can result in strong restrictions on the CFL condition. Another approach would be to use finite elements or finite volume methods on an unstructured grid, but this complicates the implementation of WENO, especially in high dimensions (for a dimension d physical space, the reduced phase space has dimension $2d-1$). Alternately, the approach taken in this work is to embed the reflector in the uniform, rectangular grid and derive appropriate boundary conditions to apply at grid points adjacent to the reflector.

An additional consideration is the knowledge of the boundary normals which is essential to computing the reflection condition. For general domain geometry, the normals need to be computed. We accomplish this via a co-dimension one level set method that computes the signed distance function to the boundary [8]. Given the signed distance function, the normal vector can be extracted by taking its gradient, using central differencing, for example. The procedure only needs to be run once before the main time loop, and the resulting normal angles may be stored for use within the main program.

1.2 Related work

In [4], a method was described for implementing the level set method introduced in [5] for the purpose of modeling high frequency underwater acoustic propagation. The present work is focused on further analysis into the implementation of reflection boundary conditions. The implementation of the boundary condition that was applied in [4] is discussed further, and alternate higher order approach is presented. Cheng et al. [8] undertook a detailed study of reflection in the level set method to improve the efficiency of the implementation proposed in [5]. In that work, the grid is augmented with the boundary condition, and the CFL condition is adjusted accordingly to maintain stability. In the event that the boundary location lies within Δx of a grid point, the point closest to the boundary is removed. However, the accuracy analysis of the finite difference WENO algorithm relies on a uniform underlying grid (or at least a smooth mapping to a uniform grid) [9]. Given the errors that will inevitably be introduced into the model by uncertainties in the ocean environment, it is desirable to work with a uniform grid in order to maintain the convergence properties of WENO to the extent that is possible. Another related approach to reflection was proposed in [10], where a DG expansion was applied to the phase space variable and reflection boundary conditions were derived in the coefficient domain. The test problems presented in this work did not expose the phase space cusp however, so it is unclear how the method would perform under more general initial conditions. We find that WENO methods perform very well for such problems. Forrer and Jeltsch [11] and also Forrer and Berger [12] present a boundary condition for computing reflections that is very similar to the higher order approach studied in this work, however they use finite volume methods, whereas the context of the present study is WENO finite differences. Finally, the work by [13] offers a very thorough description of the reinitialization problem that is applied here and in [8] to obtain the signed distance function to the boundary interface and thus the boundary normals.

1.3 Layout

The level set equations and reflection boundary conditions are reviewed in Section 2, and the proposed methods are introduced with an example in one spatial dimension. A discussion of stability and accuracy properties follows in Section 3. Results for simple test cases are presented in Section 4, and Section 5 concludes this paper.

2 Background

2.1 Level set method for shallow water acoustics

One begins with the Eikonal equation for the acoustic phase function $S(t,\mathbf{x})$ as derived from the geometric acoustics (high frequency) approximation to the linear wave equation,

$$S(t,\mathbf{x})+c(\mathbf{x})|\nabla S(t,\mathbf{x})|=0, \tag{2.1}$$

with $\mathbf{x}\in\Omega\subset\mathbb{R}^d$ and $t\in\mathbb{R}^+$. The speed of sound in the water column is given by $c(\mathbf{x})>0$, and it is assumed to be a known, smooth function. Consider the problem in two-dimensional physical space, with $\mathbf{x}=(x,z)$, and z increases with water depth. The acoustic source is located at $\mathbf{x}_s=(x_s,z_s)$. The physical model is of an infinite line source parallel to the y axis. The same level set model will result from the more commonly applied assumption of a point source under azimuthal symmetry under the restriction $x\in\mathbb{R}^+$, where in this case x is identified with the range in a cylindrical coordinate system. Both geometry types are illustrated in Fig. 1. In this work, we concentrate on the $d=2$ case as illustrated, but the techniques can be extended to higher dimensional propagation problems.

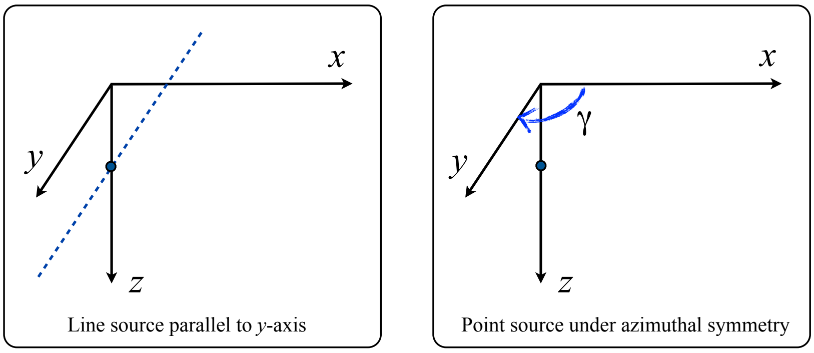

x
y
z
Line source parallel to y-axis
x
γ
y
z
Point source under azimuthal symmetry

Figure 1: Source geometry. The infinite line source symmetry is depicted on the left, and on the right the point source with azimuthal symmetry.

In the level set method with $d=2$, we embed the source location as the zero level set of a vector-valued function $\mathbf{\Phi}(t,x,z,\theta):\mathbb{R}^+\times\Omega\times[-\pi,\pi]\to\mathbb{R}^2$. Here θ is the angle of the propagation direction in the reduced phase space; the magnitude of the generalized momentum vector is determined by the sound speed c. It can be shown that the components of the level set function, ϕ_l, $l=1,2$, satisfy the following:

$$\frac{\partial\phi_l}{\partial t}=\mathcal{L}\phi_l, \qquad \mathcal{L}=-\mathbf{V}\cdot\nabla. \tag{2.2}$$

This is the transport equation that evolves the wavefront in the direction of its normal vector which is the local ray direction given by the velocity field

$$\mathbf{V}=\begin{pmatrix} c\cos(\theta) \\ c\sin(\theta) \\ \frac{\partial c}{\partial x}\sin(\theta)-\frac{\partial c}{\partial z}\cos(\theta) \end{pmatrix}. \tag{2.3}$$

The objective is to solve (2.2) subject to initial conditions, $\phi_l(0,x,z,\theta)=\phi_l^0(x,z,\theta)$, $l=1,2$. For a point source (equivalently the projection of the line source on to the $x-z$ plane) this is simply

$$\phi_1^0=x-x_s, \qquad \phi_2^0=z-z_s. \tag{2.4}$$

This equation is to be solved twice for two component level set functions for which the zero level sets are orthogonal to one another at the wavefront. This is consistent with Osher's formulation for geometric optics [5].

To compute numerical solutions, apply the semi-discrete approximation $\mathcal{L}\approx\tilde{\mathcal{L}}_x+\tilde{\mathcal{L}}_z+\tilde{\mathcal{L}}_\theta$, where $\tilde{\mathcal{L}}_{(.)}$ is the product of the relevant component of $\mathbf{V}$ evaluated on the grid, and a WENO differential operator. The WENO operator may thus be computed dimension-by-dimension. The problem statement in (2.2) and (2.4) is incomplete without the addition of boundary conditions. Periodic boundary conditions are imposed in the θ dimension, while in x and z the boundary conditions are determined by the upwind direction, discerned from the sign of $\mathbf{V}$, which is fixed given a fixed propagation direction θ. With the upwind methods, only two types of boundary condition need to be specified. An inflow condition is needed to specify function values flowing into the domain. For this, it is sufficient to apply Neumann conditions, for instance. Since the wavefronts propagate away from the source (and not into the domain), small perturbations at inflow due to assumptions made about function values flowing into the domain will not affect the location of the zero level sets. Of greater interest is the boundary condition applied along a reflecting surface which occurs for example, at a pressure release boundary (air-water interface) or from a hard rock surface.

The reflection boundary condition at a point along an interface given by (x,z_b) is derived from Snell's law and is stated in [8] as

$$\phi(t,x,z_b,\theta_{refl})=\phi(t,x,z_b,\theta_{inc}), \tag{2.5}$$

where θ_{refl} and θ_{inc} are the reflected and incident angles respectively. The subscripts indicating the level set function components have been dropped to simplify the notation. The reflected angle satisfies

$$\theta_{refl}=2\theta_B-\theta_{inc}-\pi. \tag{2.6}$$

Here θ_B is the angle of the outward normal to the reflecting surface. When the point (x,z_b) is in the grid over the physical domain, applying the boundary condition is a matter of pre-computing the incident angles and interpolating ϕ as a function of θ at a fixed point on the boundary to approximate the value $\phi(t,x,z_b,\theta_{inc})$. Note that one must be careful in implementing the interpolation strategy as the level set function generally has discontinuous derivatives at a reflection boundary. Then at each time step we can directly apply (2.5) when computing the spatial operator. For this purpose, [4] presents a WENO procedure adapted to reconstruction at an arbitrary point.

2.2 Computing boundary normals

The method proceeds based on the exposition in [8,13,14]. A level set function represents the boundary implicitly. Let Ω be the rectangle containing the portion of the water column being modeled. For convenience, take $\Omega=\{(x,z)\in[-1,1]\times[0,2]\}$. Overlay a grid $\{(x_i,z_j,\theta_k)\}_{i,j,k=0}^{N-1}$, with

$$x_i=-1+\Big(i+\frac{1}{2}\Big)h, \qquad z_j=\Big(j+\frac{1}{2}\Big)h,$$
$$\theta_k=-\pi+\pi\Big(k+\frac{1}{2}\Big)h, \qquad h=\frac{2}{N}.$$

Consider these grid points to be the centers of the cells $I_{ijk}=[x_{i-1/2},x_{i+1/2}]\times[z_{j-1/2},z_{j+1/2}]\times[\theta_{k-1/2},\theta_{k+1/2}]$.

Assume the ocean surface is given by $z=0$, and the bottom is described by $z=z_b(x)$, where $z_b(x)$ is some smooth function. Establish the following notation: $\Gamma=\{(x,z):z=z_b(x)\}$, $\Omega^+=\{(x,z):z>z_b(x)\}$, and $\Omega^-=\Omega-\Gamma-\Omega^+$. This is illustrated in Fig. 2.

We seek to compute $\theta_B(x_i)$, the angle of the outward normal to the interface at the center of each grid cell along the boundary I_{ij_bk}, $i=0,1,\cdots,N-1$, j_b is the index of the grid cell containing $z_B(x_i)$ (Fig. 3). This computation is restricted to the physical domain (x-z), so for now, ignore the k index.

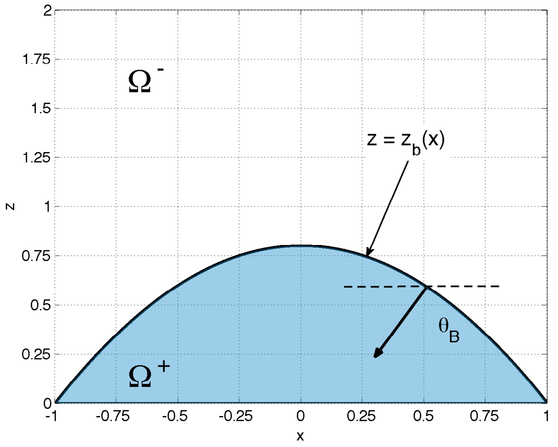


Figure 2: Domain geometry.

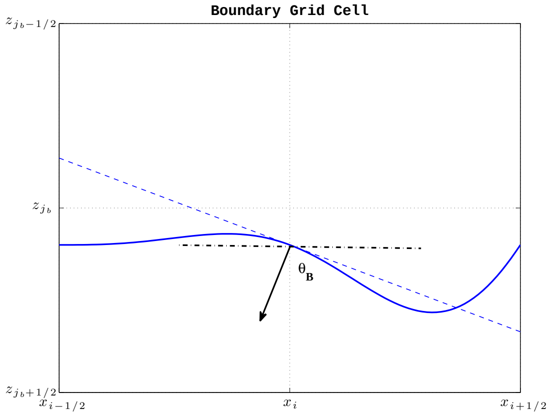


Figure 3: A grid cell intersecting the boundary.

A distance function, $D(\mathbf{x})$ is defined by

$$D(\mathbf{x}) = \min_{\mathbf{x}_I \in \Gamma}(|\mathbf{x}-\mathbf{x}_I|). \tag{2.7}$$

Thus $D(\mathbf{x})$ has the property that $D(x,z_b(x))=0$. A signed distance function, $d(\mathbf{x})$ satisfies the following additional properties:

1. $|d(\mathbf{x})| = D(\mathbf{x})$
2. $d(\mathbf{x}) = 0$ when $\mathbf{x}\in\Gamma$
3. $d(\mathbf{x}) = D(\mathbf{x})$ when $\mathbf{x}\in\Omega^+$
4. $d(\mathbf{x}) = -D(\mathbf{x})$ when $\mathbf{x}\in\Omega^-$
5. $|\nabla d(\mathbf{x})| = 1$
6. $\mathbf{N} = \nabla d$ is the unit normal to the interface Γ.

In order to obtain $\theta_B(x_i)$, we seek to construct the signed distance function, $d(x,z)$ for the interface $z=z_b(x)$. Then, using Property 6 above, we can compute $\mathbf{N}$ and use the result to compute its angle.

To construct $d(x,z)$, Sussman, Smereka, and Osher [15] suggested solving the following reinitialization equation

$$d_\tau + S(d_0)(|\nabla d| - 1) = 0 \tag{2.8}$$

to steady state, where $S(\cdot)$ is a sign function and d_0 is an initial value of d. The term *reinitialization equation* comes from the use of this equation to reset the level set function to a signed distance function in the traditional level set method. Since the characteristics of this equation propagate outward in the normal direction from the interface, this method converges quickly to a signed distance function near the interface (cf. [13]). Peng et al. [13] refined the method by replacing $S(d_0)$ in (2.8) with the smoothed out function

$$s(d)=\frac{d}{\sqrt{d^2+d_x^2\Delta x^4+d_z^2\Delta z^4}} \tag{2.9}$$

to improve convergence. We initialize with the estimate

$$d_0(x,z)=z_b(x)-z. \tag{2.10}$$

To implement the method, we use a fifth order WENO scheme to compute the spatial derivatives; the same TVDRK method as is used to solve the level set equations to discretize time. The partial differential equation (2.8) is of Hamilton-Jacobi form, with Hamiltonian

$$H(\nabla d,d)=s(d)(|\nabla d|-1). \tag{2.11}$$

For an upwind scheme, we need to define a monotone numerical Hamiltonian. We choose Godunov's method for this problem which results in the following discretization

$$\begin{aligned}\hat{H}(d_{ij})=&s_{ij}^{+}\left(\sqrt{\max((A^+)^2,(B^-)^2)+\max((C^+)^2,(D^-)^2)}-1\right)\\&+s_{ij}^{-}\left(\sqrt{\max((A^-)^2,(B^+)^2)+\max((C^-)^2,(D^+)^2)}-1\right),\end{aligned} \tag{2.12}$$

where

$$A=\frac{1}{\Delta x}\tilde{\nabla}_x^- d_{ij}, \qquad B=\frac{1}{\Delta x}\tilde{\nabla}_x^+ d_{ij},$$
$$C=\frac{1}{\Delta z}\tilde{\nabla}_z^- d_{ij}, \qquad D=\frac{1}{\Delta z}\tilde{\nabla}_z^+ d_{ij},$$

$\tilde{\nabla}^-$ and $\tilde{\nabla}^+$ represent the one-sided difference operators that look to the right and to the left respectively, and

$$\alpha^+=\max(\alpha,0), \qquad \alpha^-=\min(\alpha,0).$$

For an implementation with first order time integration, the scheme then can be written down as

$$d_{ij}^{n+1}=d_{ij}^{n}-\Delta\tau\hat{H}(d_{ij}^{n}). \tag{2.13}$$

A tolerance may be defined to determine when steady state has been reached within some band of grid cells surrounding the boundary, or in practice one could choose a fixed number of iterations. Central differencing may then be used to compute the value of the normal vector and extract θ_B for each of the x_i, $i=0,\cdots,N-1$.

2.3 Reflection

Fix (x_i,θ_k) and consider the operator $\tilde{\mathcal{L}}_z$ introduced in Section 2.1. The WENO procedure yields reconstructions $\tilde{\phi}^+_{i,j-1/2,k} \approx \phi^+(x_i,z_{j-1/2},\theta_k)$ and $\tilde{\phi}^-_{i,j+1/2,k} \approx \phi^-(x_i,z_{j+1/2},\theta_k)$ for $j = 0,\cdots,N-1$. The superscripts $(\cdot)^+$ and $(\cdot)^-$ indicate the limits from the right and left, respectively. This allows us to write

$$\tilde{\mathcal{L}}_z = c_{ij}\sin\theta_k \frac{\hat{\phi}_{i,j+1/2,k} - \hat{\phi}_{i,j-1/2,k}}{h},$$

where $\hat{\phi}$ is the numerical upwind flux given by

$$\hat{\phi}_{i,j+1/2,k} = \begin{cases} \tilde{\phi}^+_{i,j+1/2,k}, & \text{if } \sin(\theta_k) < 0, \\ \tilde{\phi}^-_{i,j-1/2,k}, & \text{otherwise.} \end{cases} \tag{2.14}$$

If $z_b(x_i) \equiv 2$, then the grid point $z_{N-1/2}$ coincides with the reflecting boundary so (2.5) is implemented by setting $\tilde{\phi}^+_{i,N-1/2,k} \doteq \tilde{\phi}^-_{i,N-1/2,N-k-1}$. In the following, we consider the situation where

$$z_b(x_i) = z_{j_b-1/2} + \gamma h, \tag{2.15}$$

with $0 < \gamma < 1$ for some index $j_b \in \{0,1,\cdots,N-1\}$. The one dimensional restriction is shown in Fig. 4.

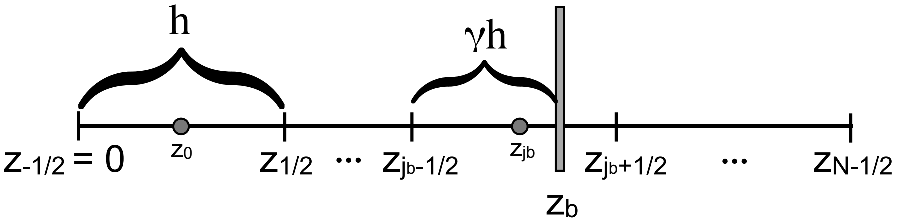


Figure 4: 1D representation of the grid with boundary located at $z_b = z_{j_b-1/2} + \gamma h$.

Two approaches are suggested. In the first, described in Section 2.3.1, the location of the boundary is approximated by the nearest grid point. In Section 2.3.2, a different approach is described which assumes that the sound speed is constant in a neighborhood of the boundary in an effort to derive a higher order approach. Each is described with respect to computing $\tilde{\mathcal{L}}_z$. The details are similar for computing $\tilde{\mathcal{L}}_z$.

2.3.1 Method 1: Boundary location approximation

This approach is straightforward and easy to implement. Given j_b for each x_i, apply (2.5) at the location $z_{j_b-1/2}$. Most of the computation can be performed prior to the main wavefront solver loop and stored in appropriate data structures. The following steps are applied in the pre-compute stage:

1. Solve the Hamilton-Jacobi equation for the signed distance function to the boundary as described in Section 2.2 and store the values. Also store, for each x_i, the angle of the outward normal to the boundary, $\theta_b(x_i)$.

2. Loop over each physical space dimension individually to locate the indices i_b for the left and right boundaries, and j_b for surface and bottom boundaries. Store the indices i_b associated with each $j=0,\cdots,N-1$, and also the indices j_b associated with each $i=0,\cdots,N-1$.

3. For each $\theta_b(x_i)$, loop over all angles θ_k, $k=0,\cdots,N-1$, to identify reflected angles (this may be accomplished by testing the sign of $\sin(\theta_k)\sin(\theta_B)+\cos(\theta_k)\cos(\theta_B)$). Store the reflected angles.

4. For each reflected angle identified, compute and store the associated incident angle using (2.6).

2.3.2 Method 2: Local ray approximation

In this approach, we combine the law of reflection with the fact that (2.2) transports function values along rays, which are straight lines when the wave speed c is constant. Assume that $c(x,z)$ is a smooth function that may be reasonably approximated as constant in a neighborhood of $z_b(x_i)$, and that $z_b(x)$ is smooth and reasonably approximated as piecewise linear. The time domain ray (characteristic) equations for the Eikonal equation are

$$\dot{x}(t)=c(x(t),z(t))\cos\theta(t), \tag{2.16a}$$

$$\dot{z}(t)=c(x(t),z(t))\sin\theta(t), \tag{2.16b}$$

$$\dot{\theta}(t)=\frac{\partial c}{\partial x}\sin\theta(t)-\frac{\partial c}{\partial x}\cos\theta(t). \tag{2.16c}$$

The dot indicates differentiation with respect to time t. For small t, we can expand $\theta(t)\approx\theta(0)\equiv\theta_0$, and likewise, using (2.16),

$$x(t)\approx x_0+tc(x_0,z_0)\cos\theta_0, \tag{2.17a}$$

$$z(t)\approx z_0+tc(x_0,z_0)\sin\theta_0. \tag{2.17b}$$

The assumption is that the travel time along a ray for which $||(x,z)-(x,z_b(x))||<h$ is small, so that the rays can be treated as straight lines in this region. Changing notation for the moment, let $R=(x_i,z_{j_b-1/2})$, and let $P=(x_b,z_b)$ where $(x_i,z_{j_b-1/2})$ are as above, and (x_b,z_b) is the point of boundary interaction for the ray that reaches point R at time t. Let τ be the travel time along that same ray from the boundary interaction until it reaches R. Then by property of the transport equation for the level set functions and (2.5),

$$\phi(t,R,\theta_{refl})=\phi(t-\tau,P,\theta_{refl})=\phi(t-\tau,P,\theta_{inc}). \tag{2.18}$$

In the absence of the reflecting boundary, the ray would continue to travel in a straight line to a location $R'\in\Omega^+$, and by (2.18), the level set function would satisfy

$$\phi(t,R,\theta_{refl})=\phi(t,R',\theta_{inc}). \tag{2.19}$$

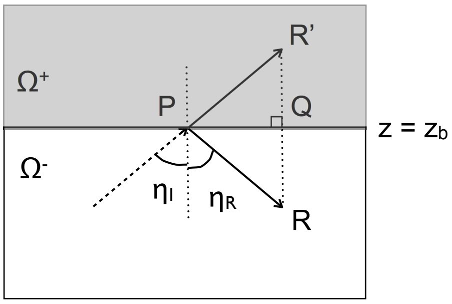


Figure 5: Law of reflection.

That is, if we continue to propagate the level set function into Ω^+, an exact boundary condition is available when c is (locally) constant and $z_b(x)$ is (locally) linear (although typically R' will not lie in the grid so interpolation is needed). So to implement the reflection condition, extend the domain of computation into Ω^+ for directions θ_k incident on the boundary, compute the locations R' for each pair (x_i,z_{j_b}), and apply (2.19).

To compute R', consider the geometry illustrated in Fig. 5. According to the law of (specular) reflection, $\eta_I=\eta_R$, where η_I is the angle between the incident ray and the boundary normal, and η_R is the angle between the boundary normal and the reflected ray. Also, under the constant wave speed assumption, the length of the ray from P to R' is equal to the length of the ray from P to R, which is $c\tau$. This implies that the line segment RR' is normal to the boundary, i.e., R' is just the normal reflection through the boundary at the point $Q=(x_i,z_b(x_i))$. Thus, given R, $\theta_B(x_i)$, and assuming the boundary is linear in the cell $(x_{i-1/2},x_{i+1/2})$, $(R')^T=TR^T$, where

$$T=\begin{pmatrix}\sin^2\theta_B-\cos^2\theta_B & -2\cos\theta_B\sin\theta_B\\ -2\cos\theta_B\sin\theta_B & -\sin^2\theta_B+\cos^2\theta_B\end{pmatrix}. \tag{2.20}$$

To implement this, the following steps additional to those listed in Section 2.3.1 are required:

5. Extend the domain into Ω^+ to allow for solutions surrounding the image location. The distance beyond the boundary of the image location is bounded above by γh, so the number of grid points to insert beyond the boundary can be determined based on the stencil size of the numerical solver and interpolation used, but a minimum of 2 additional cells is recommended.

6. Extend the wave speed $c(x,z)$ into Ω^+. If c is constant, this is trivial. More generally, it is not clear what the best approach is. We choose to set c_{ij} to be an average of the values obtained from neighboring points in Ω^-, fixing the value as constant for (x_i,z_j) such that the neighboring points are all in Ω^+, and setting $\frac{\partial c}{\partial x}\big|_{ij}=\frac{\partial c}{\partial z}\big|_{ij}=0$. It is less clear how to extend $c(x,z)$ into Ω^+ if one wishes to use a higher order approximation of the ray in the image space.

7. For each $\theta_b(x_i)$, compute and store the associated image mapping (2.20).

8. For each $\theta_b(x_i)$, compute and store the image location.

2.4 Remark on implementation

The prototype code is written in C using the GNU Scientific Library [16]. In order to store all of the precomputed information, we define the following C structures in Listings 1 and 2.

Listing 1: Boundary gridcell structure

```
typedef struct {

        size_t indx;  //Index of this cell
        int boundary_indx; //Index of the boundary
        double th_normal; //Outward normal to the boundary
        size_t num_angles; //# of inc/refl angle pairs
        size_t * refl_indx; //Array of indices into refl. angles
        size_t * non_refl_indx; //Complement of refl_indx
        gsl_vector * inc_angles; //Associated incident angles
        gsl_vector * x_im; //Store computed image locations
        gsl_vector * z_im;
        Box * neighbors;  //Store adjacent points for interpolation
        gsl_matrix * R; //Reflection map

} gridcell;
```

Listing 2: Boundary Slice structure

```
// This is a higher level structure to contain data when
// there are multiple segments in a vertical or horizontal slice.
typedef struct {

        size_t Nseg;  // Index into grid of this cell
        gridcell * Bleft; // An Nseg-length array of gridcells
        gridcell * Bright;
        double * zstar1_left; // Boundary limits
        double * zstar1_right;
        double * zstar2;

} Slice;
```

A `Slice` contains the boundary information for a single dimension in the physical space. Its members include a `gridcell` array for each boundary in that dimension. If the domain is convex, the field `Slice.Nseg`, which specifies the length of the arrays, equals 1. Otherwise, if there are multiple segments (e.g., for an object of interest inside the domain), we store a `gridcell` for each segment, tied to the relevant boundary (left or right).

3 Analysis

In this section, we study the proposed boundary conditions more closely to help quantify the error that will be imposed by the approximations. First we consider the accuracy of the boundary conditions in terms of arrival time errors. Second, we look at the stability and show that it is worthwhile to implement a higher order method such as the one described in Section 2.3.2. It is also worth noting that the piecewise linear assumption on $z_b(x)$ (or equivalently, piecewise constant for $\theta_B(x)$) results in additional $\mathcal{O}(h^2)$ error.

3.1 Arrival time accuracy

To study the error produced by this boundary condition, it is not sufficient to apply Taylor expansions since the CFL condition guarantees that ϕ is not differentiable in the interval (z_b-h,z_b). Instead, we investigate specific errors that occur. In this subsection, we focus on how the wavefront travel time is affected, which is more physically relevant than global error in ϕ since it provides the phase information for the Eikonal equation (2.1).

3.1.1 Method 1

Let t^* be the arrival time of the wavefront at the boundary z_b and consider the case where c is constant, and $\theta_B(x_i)=\frac{\pi}{2}$. Then the incident angles lie in $(0,\pi)$ and $t^*=\frac{z_s-z_b}{c\sin\theta}$. Using (2.15), the numerical wavefront arrival would occur at time $\tilde{t^*}=t^*+\frac{\gamma h}{c\sin\theta}$ so that the arrival time error, e_{t^*} is given by

$$e_{t^*}=\frac{\gamma h}{c\sin\theta}. \tag{3.1}$$

To evaluate further, consider $\theta_B=\frac{\pi}{2}$. Then the incident angles satisfy $\theta_{inc}=-\theta_{refl}$. For angles close to normal incidence, $e_{t^*}\approx\frac{\gamma h}{c}=\mathcal{O}(h)$ for a first order approximation. The worst case error will occur for incident angles that are close to parallel with the boundary. For a fixed h, this occurs when $\theta=\pi(1-\frac{h}{2})$. Then, $e_{t^*}\approx\frac{2\gamma}{\pi c}$, which appears to result in an order one error in the arrival time at these angles. In fact, we cannot guarantee for a given refinement $\{h_l\}_{l=0}^{\infty}$, with $h_l \underset{l\to\infty}{\to} 0$ that the corresponding sequence $\{\gamma_l\}$ satisfies $\lim_{l\to\infty}\gamma_l=0$. This can be seen by expressing γ_l in terms of the first value γ_0 as $\gamma_l=\frac{\gamma_0}{h_l}-\lfloor\frac{\gamma_0}{h_l}\rfloor$. The sequence $\{\gamma_l\}$ need not converge: for example, take $\gamma_0=\frac{1}{7}$ with $h_l=2^{-l}$. On the other hand, if the h_l are chosen such that there is an index L for which $\frac{\gamma_0}{h_l}$ is an integer for $l>=L$, then $e_{t^*}\to 0$ as $h_l\to 0$. This is essentially modifying the grid to include the point z_b, and that modification could be different for each fixed x. But in general, $h\to 0$ does not imply $e_{t^*}\to 0$. This error in arrival time that is proportional to γ and dependent on the arrival angle is actually a numerical consequence of the fact that as $h\to 0$, the reflected angle approaches an angle that is parallel to the boundary and is not actually a reflected angle. Since this is a consequence of the physics (i.e., the cusp in $\phi(\cdot,\theta)$) we would expect any approximation to exhibit this degradation at reflected angles that are nearly parallel to $z_b(x_i)$.

3.1.2 Method 2

As mentioned in Section 2.3.2, this boundary condition is *exact* when c is constant and $z_b(x)$ is linear in each cell $(x_{i-1/2},x_{i+1/2})$, so comparable analysis to that in Section 2.3.1 yields $e_{t^*}=0$. Instead, consider the error made in the approximation $\theta(t)\approx\theta_0$. The key assumption is that τ, the travel time along the ray that is reflected at the boundary at location P and lands at location R inside the domain (c.f. Fig. 5), is small. To investigate the validity of that assumption, we solve for τ using the ray equations (2.17) combined with the locally linear boundary assumption

$$z_b(x)\approx z_b(x_i)+\cot(\theta_B(x_i))(x-x_i). \tag{3.2}$$

Proceed by taking $P=(x_b,z_b)$, where the pair (x_b,z_b) satisfy (3.2), and $R=(x_R,z_R)$, where $|z_R-z_b|=\gamma h$ and $0<\gamma<1$. First compute the travel time, τ_1, under the approximation with c locally constant, and compare to the true travel time τ_2 for linear c. For τ_1, (2.17) becomes

$$x_R=x_b+c\tau_1\cos\theta_{refl}, \tag{3.3a}$$

$$z_R=z_b+c\tau_1\sin\theta_{refl}, \tag{3.3b}$$

$$\theta(\tau_1)=\theta_{refl}. \tag{3.3c}$$

Upon substitution of (3.2) into (3.3), we see that

$$\tau_1=\frac{\gamma h}{c\left(\cos\theta_{refl}\cot\theta_B-\sin\theta_{refl}\right)}, \tag{3.4}$$

which for $\theta_B=\frac{\pi}{2}$ is exactly (3.1) in magnitude. Taking $\theta_B=\frac{\pi}{2}$ for comparison, the remainder term for the small time Taylor expansion of $\theta(t)$ about θ_{refl} will be proportional to

$$|R_0(\tau_1)|=\frac{\gamma h}{|\sin(\theta_{refl})|}\left|\frac{\partial c}{\partial x}\sin\theta(\xi)-\frac{\partial c}{\partial z}\cos\theta(\xi)\right| \tag{3.5}$$

for some $\xi\in(0,\tau_1)$. Again, this is order h for reflection angles close to normal, and order 1 for angles near parallel with the boundary, with the constant dependent on the sound speed profile, as $h\to 0$. Typical sound speed profiles vary more rapidly in depth than in range, so we would expect the $\frac{\partial c}{\partial z}\cos\theta$ term to dominate.

We can now consider how the approximation affects the travel time error by looking at the effect when c is linear and independent of x, since in that case an analytical solution is available so that we may compute the exact travel time τ_2 and compare to τ_1 to obtain an error estimate. Letting $c(z)=gz+c_0$ where g and c_0 are known constants, the travel time along the ray that leaves the boundary z_b and terminates at z_R is given by $\tau=\int_0^s\frac{1}{c(z(u))}du$, where s is the arc length of the ray. For the given c, the travel time is [17, p. 210]

$$\tau_2=\left|\frac{1}{g}\log\left[\frac{1+\sqrt{1-a^2c^2}}{ac}\right]_{c(z_b)}^{c(R)}\right|. \tag{3.6}$$

The constant a is found by applying Snell's law to the ray at its turning point, yielding $a=\frac{\cos\theta_{refl}}{c(z_b)}$. Substituting back into (3.6) and using the fact that $c(z_b)=c(R)+g\gamma h$,

$$\tau_2=\left|\frac{1}{g}\log\left(\frac{c(R)+g\gamma h+\sqrt{c(R)^2\sin^2\theta_{refl}+g\gamma h(2c(R)+g\gamma h)}}{c(R)\left(1+|\sin\theta_{refl}|\right)}\right)\right|. \tag{3.7}$$

For reflections roughly normal to the boundary, $\theta_{refl}\approx-\frac{\pi}{2}$, (3.7) simplifies to

$$\begin{aligned}\tau_2&=\left|\frac{1}{g}\log\left(1+\frac{g\gamma h}{c(R)}\right)\right|\approx\left|\frac{\gamma h}{c(R)}-\frac{g\gamma^2h^2}{c(R)^2}\right|\\&=\tau_1+C(\gamma h)^2.\end{aligned} \tag{3.8}$$

Then, $e_{t^*}=\mathcal{O}(h^2)$ as $h\to 0$ when the approximation is applied for a linear sound speed profile with normal incidence. Recall that in Section 3.1.1 e_{t^*} was derived assuming that the sound speed was constant; the error would be larger had it been assumed that c was not constant due to an error in the angles. This result suggests this method would provide improved performance over the location approximation method even for the variable coefficient case.

Again, we would expect to observe degraded performance when the angle the ray makes with the boundary is near parallel. Given a fixed mesh spacing h, the worst case would be $\theta_{refl}=-\pi\frac{h}{2}$, for which we have $\tau_1\approx\frac{2\gamma}{\pi c(R)}-\mathcal{O}(h^2)$. Expanding (3.7) about small γh leads to

$$\begin{aligned}\tau_2&\approx\left|\frac{2\gamma}{\pi c(R)}\left(\pi\sqrt{\frac{c(R)}{2\gamma g}}\sqrt{h}\right)-\frac{\gamma h}{c(R)}\left(1+\frac{\pi c(R)}{2g\gamma}\right)\right|\\&=\tau_1\sqrt{\frac{c(R)\pi^2}{2\gamma g}}\sqrt{h}+\mathcal{O}(\gamma h).\end{aligned} \tag{3.9}$$

Then e_{t^*} is proportional to

$$\tau_2-\tau_1\approx\sqrt{\frac{2\gamma}{\pi c(R)}}\left(\sqrt{\frac{\pi h}{g}}-\sqrt{\frac{2\gamma}{\pi c(R)}}\right).$$

As $h\to 0$, $e_{t^*}\to-\tau_1$, which is representative of the fact that as θ_{refl} approaches parallel to the boundary, under the linear c assumption, the ray has a turning point at z_b so that it grazes the boundary but does not reflect. So the arrival time error will have the same magnitude as that in Section 3.1.1 when a linear c is approximated by constant c.

3.2 Stability

To simplify the analysis, we study the 1D problem for x. First we set up the analysis when the exact boundary condition is applied (i.e., the boundary location is in the grid).

In Section 3.3, we consider the effect of errors incurred in approximating the boundary condition. In one dimension, there are only two directions given by $\theta \in \{-\pi,0\}$. Equivalently, we solve the system

$$\frac{\partial \mathbf{U}}{\partial t}+\mathbf{A}(x)\frac{\partial \mathbf{U}}{\partial x},$$

where

$$\mathbf{U}=\begin{pmatrix} u \\ v \end{pmatrix}(t,x), \qquad \mathbf{A}(x)=\begin{pmatrix} c(x) & 0 \\ 0 & -c(x) \end{pmatrix}.$$

Initial and boundary conditions are required to pose the problem completely. In this case, we have

$$u(0,x)=v(0,x)=x-x^*,$$

where x^* is a fixed, pre-determined point in $(0,x_B)$. It is assumed that $c(x)>\epsilon>0$, so that $u(t,x)$ represents a right-traveling wave and $v(t,x)$ is a left-traveling wave. Thus, boundary conditions need to be imposed at $x=0$ for u and at $x=x_B$ for v. Assume for now that reflection only occurs at $x=x_B$, and apply an inflow condition at $x=0$:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\frac{\partial \mathbf{U}}{\partial t}(t,0)=0,$$
$$\begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix}\mathbf{U}(t,x_B)=0.$$

The two equations are now coupled by the reflection boundary condition.

To further simplify the analysis, consider the constant coefficient problem in the half plane $x\in(0,\infty)$. We will use standard notation for $k=\Delta t$, $\lambda=\frac{k}{h}$, and E^j is the shift operator: $E^j(U(x))=U(x+jh)$. Apply the first order scheme with reflection at $x=0$:

$$\begin{aligned}\mathbf{U}_j^{n+1}&=\left[\begin{pmatrix} c\lambda & 0 \\ 0 & 0 \end{pmatrix}E^{-1}+(1-c\lambda)\mathbf{I}+\begin{pmatrix} 0 & 0 \\ 0 & c\lambda \end{pmatrix}E^1\right]\mathbf{U}_j^n \\ &\equiv Q\mathbf{U}_j^n\equiv\left(\mathbf{A}_{-1}E^{-1}+\mathbf{A}_0E^0+\mathbf{A}_1E^1\right)\mathbf{U}_j^n, \qquad j=1,2,\cdots, \qquad (3.10)\end{aligned}$$

$$\mathbf{U}_0^{n+1}=\left[\begin{pmatrix} 0 & 1-c\lambda \\ 0 & 1-c\lambda \end{pmatrix}+\begin{pmatrix} 0 & c\lambda \\ 0 & c\lambda \end{pmatrix}E^1\right]\mathbf{U}_0^n. \qquad (3.11)$$

We first study the effect of only the boundary condition on the left, using the techniques presented in [18] and [19].

Under a few assumptions on Q (which are satisfied for this scheme; refer to [18]), Q has two linearly independent right-going and two linearly independent left-going solutions when $|\zeta|\geq 1$. It is established in [18] that the number of independent right-going solutions is determined by the size of the system multiplied by the number of stencil points in Q to the left of center, and likewise for the left-going solutions with respect to the number of stencil points to the right of center. We now consider solutions in the form

$$U_j^n=\hat{U}\kappa^j\zeta^n, \qquad \kappa,\zeta\in\mathbb{C}-\{0\}, \qquad (3.12)$$

with $\kappa = e^{i\xi h}$ and $\zeta = e^{i\omega k}$. The dispersion relation relates ω to ξ for waves. Analogously, we follow [19] and consider the numerical dispersion relation for ζ in terms of κ. This is given by substituting (3.12) into Q and solving. We find that the resulting system satisfies

$$\begin{pmatrix} \zeta - \frac{c\lambda}{\kappa} - 1 + c\lambda & 0 \\ 0 & \zeta - [c\lambda(\kappa-1)+1] \end{pmatrix} \hat{U} = 0.$$

The matrix on the left hand side is singular when

$$\zeta = c\lambda \frac{1-\kappa}{\kappa} + 1,\ c\lambda(\kappa-1)+1. \tag{3.13}$$

The first solution is for the right-going solution component $u(t,x)$ and the second is for the left-going $v(t,x)$. In the subsequent, we will use μ instead of κ for the left-going solutions and retain κ only for the right-going solutions, so the second dispersion relation in (3.13) becomes

$$\zeta = c\lambda(\mu-1)+1.$$

We now seek solutions to the scheme (3.10) that also satisfy (3.11) and have the form

$$U_j^n = \zeta^n \sum_{m=1}^{2} \alpha_m \hat{P}_m \kappa_m^j + \zeta^n \sum_{m=1}^{2} \beta_m \hat{R}_m \mu_m^j. \tag{3.14}$$

Continuing to follow Trefethen [19], apply the boundary condition (3.11)-(3.14) and collect terms in α and β to arrive at an expression of the form

$$E(\zeta) \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = D(\zeta) \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}.$$

Our concern is solutions for which the coefficients α cannot be fully specified in terms of β when $|\zeta| \geq 1$, i.e., $|\zeta| \geq 1$ for which $E(\zeta)$ is singular. We see that

$$\begin{aligned} E(\zeta) &= \begin{pmatrix} \zeta \hat{P}_1^1 + [c\lambda(1-\kappa_1)-1]\hat{P}_1^2 & \zeta \hat{P}_2^1 + [c\lambda(1-\kappa_2)-1]\hat{P}_2^2 \\ (\zeta + c\lambda(1-\kappa_1)-1)\hat{P}_1^2 & (\zeta + c\lambda(1-\kappa_2)-1)\hat{P}_2^2 \end{pmatrix}, \\ D(\zeta) &= \begin{pmatrix} -\zeta \hat{R}_1^1 + [c\lambda(\mu_1-1)+1]\hat{R}_1^2 & -\zeta \hat{R}_2^1 + [c\lambda(\mu_2-1)+1]\hat{R}_2^2 \\ (1-\zeta + c\lambda(\mu_1-1))\hat{R}_1^2 & (1-\zeta + c\lambda(\mu_2-1))\hat{R}_2^2 \end{pmatrix}, \end{aligned}$$

so $E(\zeta)$ is singular when

$$\zeta \det(\hat{\mathbf{P}})(\zeta + c\lambda(1-\kappa)-1) = 0. \tag{3.15}$$

Here, $\hat{\mathbf{P}} = [\hat{P}_1 \hat{P}_2]$ and we have used the fact that the dispersion relation for the scheme produces only one solution for right going waves so that $\kappa_1 = \kappa_2 \equiv \kappa$. In other words, instability is present (for $\zeta \in \mathfrak{C} - \{0\}$) when $\hat{\mathbf{P}}$ is singular, implying that there are not two linearly independent right going waves, or $\zeta = 1 - c\lambda(1-\kappa)$. In the case that $\hat{\mathbf{P}}$ is singular, the problem is one dimensional and the condition is trivially satisfied using the consequence of the dispersion relation that $\kappa = \frac{1}{\mu}$, with a reflection coefficient $E(\zeta)^{-1}D(\zeta) = 1$.

On the other hand, if $\hat{\mathbf{P}}$ has full rank, then $E(\zeta)$ is singular only when $\zeta=1-c\lambda(1-\kappa)$. Applying once again the dispersion relation, there are two possibilities, $\zeta=1$ or $\zeta=1-2c\lambda$. If $\zeta=1$, then $\kappa=\frac{1}{\mu}=1$ yielding the solution $\mathbf{U}(t,x)\equiv$ constant. So the approximation is stable as long as $c\lambda<1$ since then $|1-2c\lambda|<1$.

Again as in [19], we can define the reflection coefficient matrix $A_R(\zeta)=E^{-1}(\zeta)D(\zeta)$ when E is nonsingular. It is straightforward to compute

$$A_R(\zeta)=\frac{1}{\det\hat{\mathbf{P}}}\begin{pmatrix}\hat{P}_2^2(\hat{R}_1^2-\hat{R}_1^1) & \hat{P}_2^2(\hat{R}_2^2-\hat{R}_2^1)\\ -\hat{P}_1^2(\hat{R}_1^2-\hat{R}_1^1) & -\hat{P}_1^2(\hat{R}_2^2-\hat{R}_2^1)\end{pmatrix},$$

which is independent of ζ. This matrix is actually singular and yields solutions $\alpha_2\propto\alpha_1$ which suggests that the analysis for the case $\hat{\mathbf{P}}$ is singular is the correct analysis, with reduced reflection coefficient of one.

In a similar fashion, it can be shown that the problem for a reflection boundary at $x=1$ with boundary condition

$$\mathbf{U}_N^{n+1}=\left[\begin{pmatrix}c\lambda & 0\\ c\lambda & 0\end{pmatrix}E^{-1}+\begin{pmatrix}1-c\lambda & 0\\ 1-c\lambda & 0\end{pmatrix}\right]\mathbf{U}_N^n$$

is also stable with one linearly independent solution in each direction, and reflection coefficient

$$A_R(\zeta)=\kappa^{2N}.$$

This is stable since for $|\zeta|\geq 1$, $|\kappa|=\left|\frac{c\lambda}{\zeta-1+c\lambda}\right|\leq 1$. By Proposition 6 in [19], the model is stable in the sense defined in [18] because the interfaces at $x=0$ and at $x=1$ are both individually stable.

3.3 Effect of perturbations

The proposed boundary conditions rely on an approximation thus an error is imposed at each reflecting boundary at each time step. In this section, we study how the error made at the boundary propagates in the one dimensional case above. Consider (3.10) with initial condition $\mathbf{U}_j^0=0$ and perturbed boundary condition $u_0^n=v_0^n+\epsilon$, where ϵ is the $\mathcal{O}(h)$ or $\mathcal{O}(h^2)$ approximation error for the boundary condition. We also have the reflection boundary condition at $x=1$ for v, $v_N^n=u_N^n$. We consider the u_j^n and v_j^n components separately in the scheme (3.10) to construct the discrete error equation.

We seek to solve the following difference equation:

$$u_j^n=(1-c\lambda)u_j^{n-1}+c\lambda u_{j-1}^{n-1} \tag{3.16}$$

subject to the initial condition $u_j^0=0$ and boundary condition $u_0^n=\epsilon$. The transport property further implies that

$$u_j^n=0 \quad \text{for } j\geq n. \tag{3.17}$$

We use generating functions to solve the difference equation. First, let

$$u_n(x)=\sum_{j=0}^{\infty}u_j^n x^j=\sum_{j=0}^{n-1}u_j^n x^j, \tag{3.18}$$

where the second equality is due to (3.17). Now multiply (3.16) by x^j and sum over j to find

$$\begin{aligned}u_{n+1}(x)&=(1-c\lambda)u_n(x)+\sum_{k=-1}^{n-2}c\lambda u_k^n x^{k+1}\\&=(1-c\lambda)u_n(x)+c\lambda x u_n(x)+c\lambda u_{-1}^n\\&=(1-c\lambda+c\lambda x)u_n(x).\end{aligned} \tag{3.19}$$

Now let $u(x,y)=\sum_{n=0}^{\infty}u_n(x)\frac{y^n}{n!}$, multiply (3.19) by $\frac{y^n}{n!}$, and sum over n:

$$\sum_{n=1}^{\infty}u_n(x)\frac{y^{n-1}}{(n-1)!}=\sum_{n=0}^{\infty}(1-c\lambda+c\lambda x)u_n(x)\frac{y^n}{n!}.$$

Or,

$$\frac{\partial u(x,y)}{\partial y}=(1-c\lambda+c\lambda x)u(x,y), \tag{3.20}$$

which has the solution $u(x,y)=C(x)e^{(1-c\lambda+c\lambda x)y}$. To find $C(x)$,

$$\begin{aligned}u(x,1)=C(x)e^{1-c\lambda+c\lambda x}&=\sum_{n=0}^{\infty}u_n(x)\frac{1}{n!}\\ \Longrightarrow u_n(x)&=C(x)(1-c\lambda+c\lambda x)^n.\end{aligned}$$

Taking $n=1$,

$$C(x)(1-c\lambda+c\lambda x)=u_0^1+u_1^1x+u_2^1x^2+\cdots=\epsilon.$$

Thus,

$$u(x,y)=\frac{\epsilon}{1-c\lambda+c\lambda x}e^{(1-c\lambda+c\lambda x)y},$$

and

$$u_n(x)=\epsilon(1-c\lambda+c\lambda x)^{n-1}. \tag{3.21}$$

Comparing the above expression to (3.18) and applying the binomial theorem, we see that

$$u_j^n=\begin{cases}\epsilon\binom{n-1}{j}(1-c\lambda)^{n-j-1}(c\lambda)^j, & j=0,1,\cdots,n-1,\\ 0, & \text{otherwise.}\end{cases} \tag{3.22}$$

Next, solve the difference equation produced by (3.10) for v_j^n subject to a perturbation at the boundary using the same techniques. The difference equation is

$$v_{N-j}^n=(1-c\lambda)v_{N-j}^{n-1}+c\lambda v_{N-j+1}^{n-1}, \tag{3.23}$$

subject to initial condition $v^0_{N-j}=0$, and boundary condition $v^n_N=\epsilon$, $n=1,2,\cdots$ It is a consequence of the transport equation that $v^n_{N-j}=0$ for $j\geq n$, and we can use this property in addition to the boundary condition to derive the useful identity

$$v^n_{N-n+1}=\epsilon(c\lambda)^{n-1}. \tag{3.24}$$

We use the expansion

$$v_n(x)=\sum_{j=0}^{n-1} v^n_{N-j}x^{-j}, \tag{3.25}$$

to find

$$v_{n+1}(x)=(1-c\lambda)v_n(x)+\frac{c\lambda}{x}\sum_{j=0}^{n-1}v^n_{N-j}x^{-j}+c\lambda v^n_{N+1}-\frac{c\lambda}{x}v^n_{N-n+1}x^{-(n-1)}.$$

Using (3.24) and that $v^n_{N+1}=0$ yields the difference equation in n:

$$v_{n+1}(x)=\left(1-c\lambda+\frac{c\lambda}{x}\right)v_n(x)-\epsilon\left(\frac{c\lambda}{x}\right).$$

Again, let $v(x,y)=\sum_{n=0}^{\infty}v_n(x)\frac{y^n}{n!}$. Then the resulting differential equation is

$$\frac{\partial v(x,y)}{\partial y}=\left(1-c\lambda+\frac{c\lambda}{x}\right)v(x,y)-\epsilon\exp\left(\frac{c\lambda}{x}y\right). \tag{3.26}$$

Eq. (3.26) produces the homogeneous solution

$$v_1(x,y)=C(x)\exp\left[\left(1-c\lambda+\frac{c\lambda}{x}\right)y\right],$$

and particular solution $v_2(x,y)=D(x)\exp\left[\frac{c\lambda}{x}y\right]$. Upon substitution into (3.26) we find that $D(x)=\frac{\epsilon}{1-c\lambda}$, so that the generating function for the coefficients v^n_{N-j} is

$$v(x,y)=C(x)\exp\left[\left(1-c\lambda+\frac{c\lambda}{x}\right)y\right]+\frac{\epsilon}{1-c\lambda}\exp\left[\frac{c\lambda}{x}y\right].$$

Now

$$\begin{aligned}v(x,1)&=C(x)\exp\left[\left(1-c\lambda+\frac{c\lambda}{x}\right)y\right]+\frac{\epsilon}{1-c\lambda}\exp\left[\frac{c\lambda}{x}y\right]\\&=\sum_{n=0}^{\infty}\frac{1}{n!}\left(C(x)\left(1-c\lambda+\frac{c\lambda}{x}\right)^n+\frac{\epsilon}{1-c\lambda}\left(\frac{c\lambda}{x}\right)^n\right)\\&=\sum_{n=0}^{\infty}v_n(x)\frac{1}{n!},\end{aligned}$$

so

$$v_n(x)=C(x)\left(1-c\lambda+\frac{c\lambda}{x}\right)^n+\frac{\epsilon}{1-c\lambda}\left(\frac{c\lambda}{x}\right)^n.$$

Considering $n=1$,

$$\begin{aligned}v_1(x)&=C(x)\left(1-c\lambda+\frac{c\lambda}{x}\right)+\frac{\epsilon}{1-c\lambda}\frac{c\lambda}{x}\\&=\sum_{j=0}^{n-1}v^1_{N-j}x^{-j}=v^1_N=\epsilon,\end{aligned}$$

which leads to the result

$$C(x)=\epsilon\frac{1-c\lambda-\frac{c\lambda}{x}}{(1-c\lambda)\left(1-c\lambda+\frac{c\lambda}{x}\right)}.$$

Substituting this back into the expression for $v_n(x)$, we have

$$v_n(x)=\frac{\epsilon}{1-c\lambda}\left[\left(1-c\lambda-\frac{c\lambda}{x}\right)\left(1-c\lambda+\frac{c\lambda}{x}\right)^{n-1}+\left(\frac{c\lambda}{x}\right)^n\right]. \tag{3.27}$$

Finally, in order to compare (3.27) to the coefficient terms in (3.25), expand using the binomial theorem:

$$\begin{aligned}v_n(x)&=\epsilon\left[\left(1-c\lambda+\frac{c\lambda}{x}\right)^{n-1}-\frac{c\lambda}{x(1-c\lambda)}\sum_{j=0}^{n-1}\binom{n-1}{j}(1-c\lambda)^{n-1-j}\left(\frac{c\lambda}{x}\right)^j+\frac{1}{1-c\lambda}\left(\frac{c\lambda}{x}\right)^n\right]\\&=\epsilon\left[\left(1-c\lambda+\frac{c\lambda}{x}\right)^{n-1}-\sum_{j=1}^{n-1}\binom{n-1}{j-1}(1-c\lambda)^{n-1-j}\left(\frac{c\lambda}{x}\right)^j\right]\\&=\epsilon\left[\binom{n-1}{0}(1-c\lambda)^{n-1-0}\left(\frac{c\lambda}{x}\right)^0+\sum_{j=1}^{n-1}\left[\binom{n-1}{j}-\binom{n-1}{j-1}\right](1-c\lambda)^{n-1-j}\left(\frac{c\lambda}{x}\right)^j\right]\\&=\sum_{j=0}^{n-1}\epsilon(1-c\lambda)^{n-1}\left(\frac{n-2j}{n}\right)\binom{n}{j}\left(\frac{c\lambda}{1-c\lambda}\right)^j x^{-j}.\end{aligned}$$

That is,

$$v^n_{N-j}=\begin{cases}\epsilon(1-c\lambda)^{n-1}\left(\frac{n-2j}{n}\right)\binom{n}{j}\left(\frac{c\lambda}{1-c\lambda}\right)^j, & j=0,1,\cdots,n-1,\\0, & \text{otherwise.}\end{cases} \tag{3.28}$$

We can now apply (3.22) and (3.28) to explore how the error produced at the left boundary propagates. The perturbation at x=0 reaches the right boundary, x=1, at time t=$(N+1)k$, with value $u^{N+1}_N=(c\lambda)^N\epsilon$. The reflection boundary condition at $x=1$ is $v^{N+1}_N=u^{N+1}_N=(c\lambda)^N\epsilon$. Assuming the boundary condition on the right is exact, then the perturbation returns to $x=0$ as

$$v^{2N+1}_0=\epsilon(c\lambda)^{2N}\binom{2N}{N}\left(1-\frac{N}{N+1}\right)(1-c\lambda)^N. \tag{3.29}$$

Upon the next reflection at $x=0$, this means there is an additive error, $u_0^{2N+1}=\epsilon+v_0^{2N+1}$. It is evident that the error continues to propagate through the domain. We can use Stirling's approximation

$$N! \approx \sqrt{2\pi N}\left(\frac{N}{e}\right)^N \tag{3.30}$$

to get a better sense of the size of the error for large n, with n expressed as multiples of N, so this is also a mesh refinement study. Substituting (3.30) in (3.29),

$$v_0^{2N+1} \approx \epsilon\left[4(c\lambda)^2(1-c\lambda)\right]^N \frac{2}{(N+1)\sqrt{\pi N}}. \tag{3.31}$$

The limiting behavior of (3.31) is overwhelmed by the exponential term in the numerator. Write $g(c\lambda)=(c\lambda)^2(1-c\lambda)$. The base of the exponential term is then $4g(c\lambda)$. Since $c\lambda<1$ for stability, we have $g(c\lambda)\le\frac{4}{27}<\frac{1}{4}$. Thus $4g(c\lambda)<1$, and although the error persists, it decreases exponentially as $h\to 0$ so it does not have a significant effect on either method's convergence rate. To be thorough, we computed the error multiplier with $c\lambda=\frac{2}{3}$ (this value of $c\lambda$ maximizes $g(c\lambda)$ over $(0,1)$) for $N=10$, $N=20$, and $N=40$. The added errors were, respectively, $\epsilon 8.554\times10^{-5}$, $\epsilon 1.7026\times10^{-7}$, and $\epsilon 1.76\times10^{-12}$, where the result for $N=40$ is based on an approximation for large N.

3.4 Remark

In Sections 3.2 and 3.3, the analysis is limited to the one-dimensional case. The fact that in this case the reduced phase space is discrete, consisting of only two directions, permits this type of study. Local stability in two dimensions follows for any pair of incident and reflected angles on the surface by coordinate transform. An extension to fully two-dimensional propagation is left to future work, although the experimental results for the two-dimensional propagation problem suggest the method is stable.

4 Results

4.1 Validation of arrival time analysis

Computational results for the arrival time errors are shown in Figs. 6 and 7 below. In both cases, the scenario is for an upward sloping bottom with $\theta_B=\frac{\pi}{3}$. Fig. 6 is for a constant wave speed $c=c_0=1.5$ km/sec, and for Fig. 7, the wave speed is given as $c(z)=0.5z+c_0$. In each figure, the subplots on the left display the error computed at time $T=0.3$ seconds along the wavefront as parameterized by local (reflected) ray direction θ. The subplots on the right hand side show the error as a function of grid size N at the angle $\theta=-\frac{2\pi}{3}$, which is normal to the sloping bottom.

From Fig. 6, it appears that the convergence rate for Method 2 is greater than for Method 1. Although we claim that Method 2 is exact for this scenario, we would not

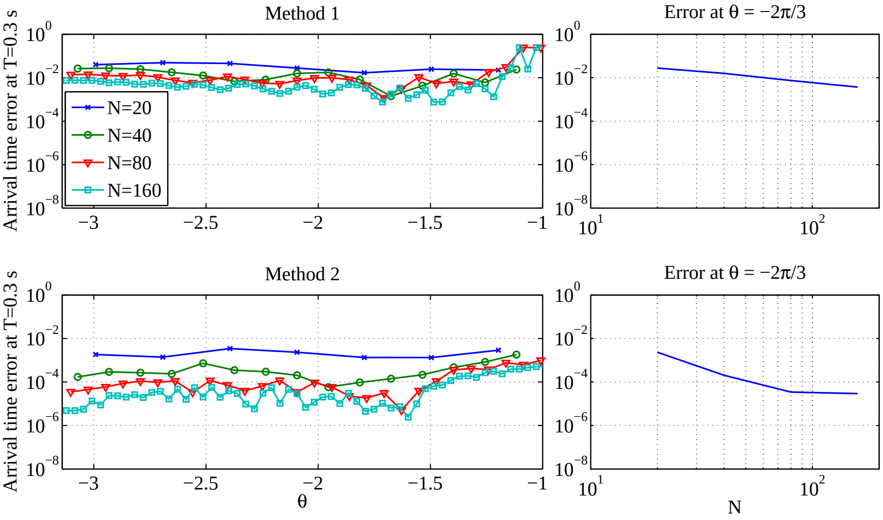


Figure 6: Arrival time errors with $c = 1.5$ km/sec.

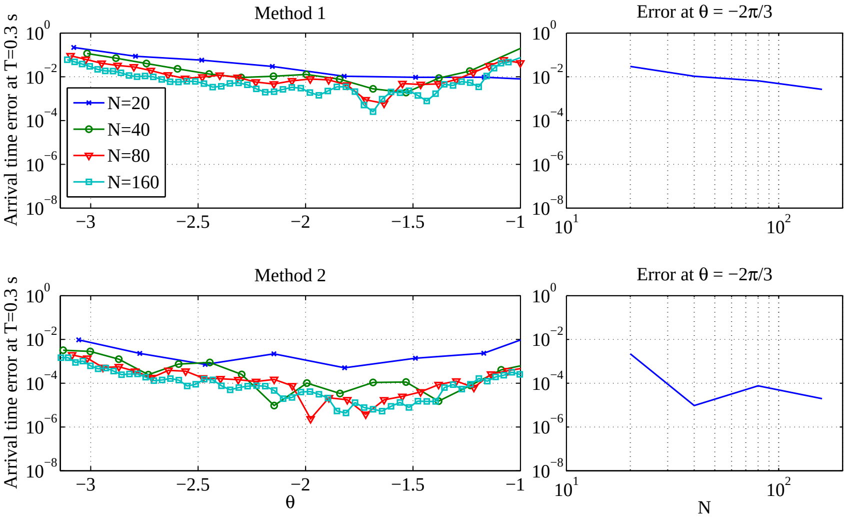


Figure 7: Arrival time errors with $c = 0.5z + 1.5$ km/sec.

expect to observe convergence that is any faster than $\mathcal{O}(h^2)$ due to coupling with the errors in the PDE solver and wavefront extraction routine. We see that the convergence rate for Method 1 is roughly $\mathcal{O}(h)$ and for Method 2, about $\mathcal{O}(h^2)$. The error levels off for Method 2 at about 10^{-5}, this is likely due to time discretization error. In fact, for $\Delta t = 0.0087$ with $N = 160$, there is an error on the order $(\Delta t)^3$ due to the third-order Runge-Kutta time integration. This leads to an error $(\Delta t)^3 \times N_t = 2.2 \times 10^{-5}$ where $N_t = 34$ is the number of times steps processed to compute the arrival time error.

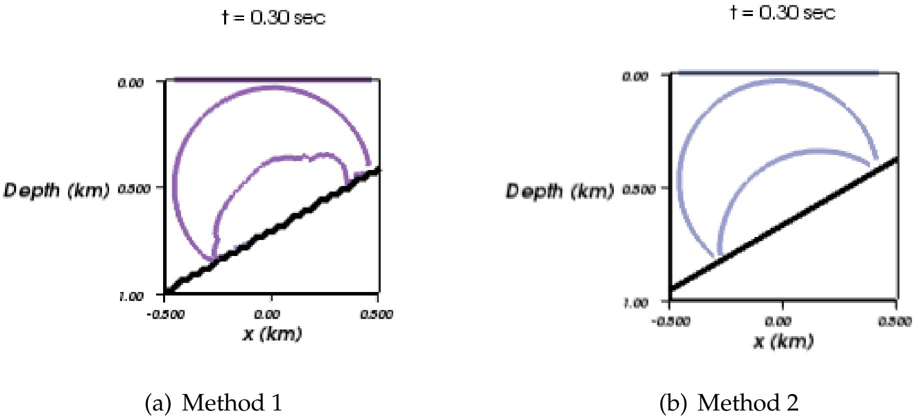
Fig. 7 confirms that for the linear sound speed, the convergence rate slows as the reflected angle becomes closer to parallel with the bottom in both Method 1 and Method 2. The rate close to $\theta=-\frac{2\pi}{3}$ is roughly $\mathcal{O}(h)$ for Method 1. It is less clear for Method 2, but the results appear to be converging faster than $\mathcal{O}(h)$ at the same angle. For both plots the errors noticeably dip near $\theta=-\frac{\pi}{2}$. At this angle, the velocity component in the x-direction is zero, reducing the dimensionality of the problem.

4.2 Wavefront examples

To provide some visual results, we present a few examples of wavefronts extracted from the level set functions in the presence of non-conforming grid geometry with reflections. The visualizations in this section were produced using the Mayavi program [20].

4.2.1 Comparison between Method 1 and Method 2

Fig. 8 shows the results from a $40\times40\times40$ grid over $\Omega=\{(x,z)\in[-0.5,0.5]\times[0,1]\}$ and a sloping bottom having $\theta_B=\frac{\pi}{3}$ radians. A constant sound speed $c=1.5$ km/sec is used here. At this resolution, it is clear that Method 2 is preferred, as the artifacts of the boundary representation used in Method 1 are apparent. Note that although the representation of the bottom location is piecewise constant, the proper normals are applied. When the grid size is increased to $160\times160\times160$, the wavefront produced using the Method 1 boundary condition is converging, though very slowly at angles close to parallel with the boundary.



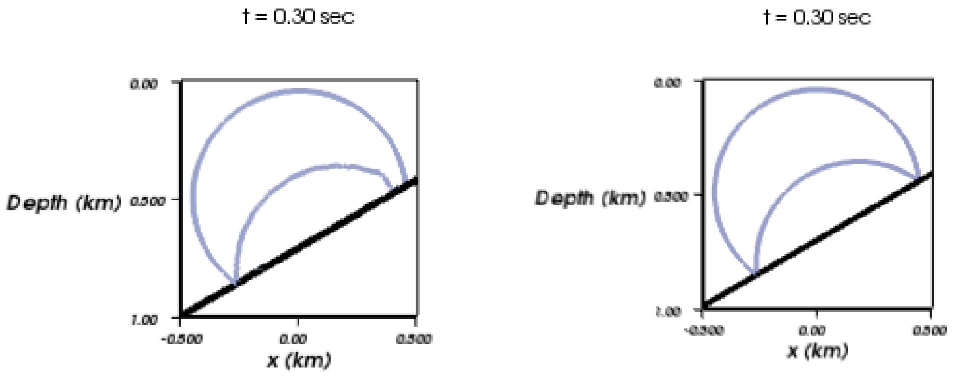
(a) Method 1

(b) Method 2

Figure 8: Wavefront comparison with $N=40$ at 0.3 seconds, $c=1.5$ km/s, $\theta_B=\frac{\pi}{3}$.

4.2.2 Examples with non-linear geometry

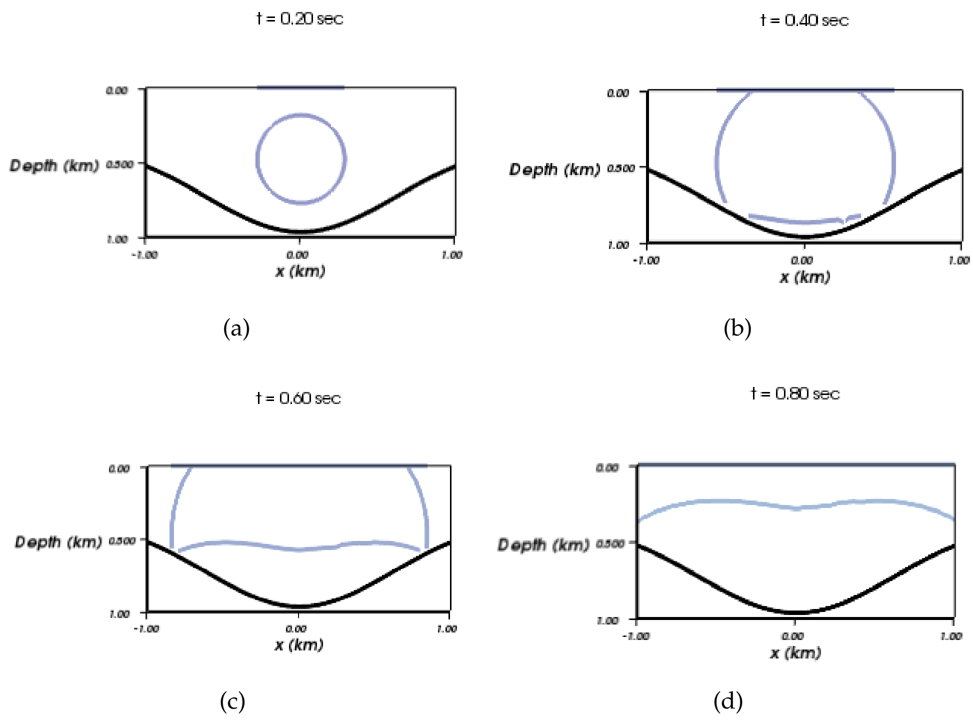
Figs. 10 and 11 are examples of the types of geometry for which Method 2 provides nice results. Again, a constant sound speed $c=1.5$ km/sec is used for simplicity. The domain



(a) Method 1 (b) Method 2

Figure 9: Wavefront comparison with $N=160$ at 0.3 seconds, $c=1.5$ km/s, $\theta_B=\frac{\pi}{3}$.

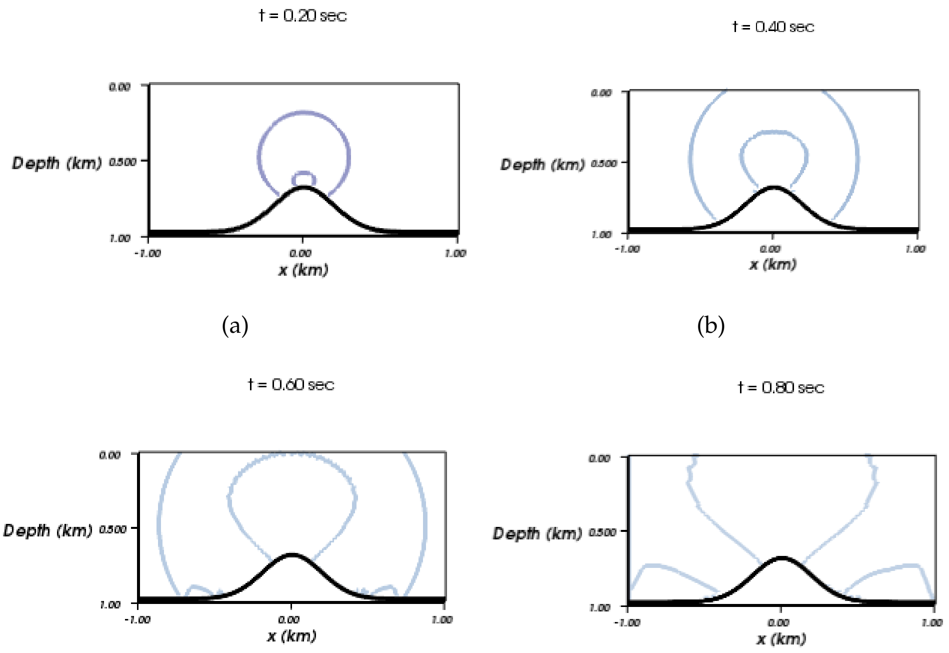
for each is $\Omega=\{(x,z)\in[-1,1]\times[0,1]\}$. Each subfigure is a snapshot of the wavefront at a few times, $t=0.2,0.4,0.6,0.8$. In Fig. 10, a modified Gaussian function is applied to represent an undersea canyon, and in Fig. 11, a seamount is represented by a Gaussian function. Note that the Gaussian function is not a limitation, just a convenient choice.



(a) (b)

(c) (d)

Figure 10: Wavefront snapshots with $z_b(x)=1-\frac{1}{0.6\sqrt{2\pi}}\left(1-\exp[-\frac{x^2}{2(0.6)^2}]\right)$.



(a)

(b)

(c)

(d)

Figure 11: Wavefront snapshots with $z_b(x) = 1 - 0.3\exp\left[-12.5x^2\right]$.

Due to the multiple reflections produced by the canyon example, we were not able to produce reasonable results using Method 1.

5 Conclusion

Level set methods provide an alternative framework for numerical solutions to the high frequency approximation to the wave equation. A fixed-grid point of view is advantageous for medium range (about 1 kilometer), high frequency modeling. In particular, it is very difficult to compute accurate eigenray solutions for active sonar in shallow water environments with range-dependent bathymetric features. Reflections are an important consideration for shallow water acoustics, and a reasonable approximation at high frequencies where the amount of energy transmitted into the sea bottom is negligible.

We have presented a modified level set method directed toward computational underwater acoustics modeling. Two options are proposed for boundary conditions, the first (Method 1) using approximate boundary locations to maintain a uniform grid, and the second (Method 2) taking advantage of the transport property of the level set equation and the law of reflection. Although both yield correct arrival times in the large N limit at most angles, the method based on the law of reflection is shown to converge about one order faster than the method using an approximate location. An analysis of the effect of the boundary condition on arrival times is supported by the results in Section 4.1. The computational examples in Figs. 10 and 11 above show that we are able to compute very nice wavefronts on less restrictive geometry types using Method 2. It is straightforward using the grid structures presented to extend the approach to any smooth boundary representation, including bathymetry data that have been smoothed and interpolated to yield a functional representation. Thus, the tools are available to study more practical propagation problems using the level set method.

Acknowledgments

The author would like to thank Dr. Drumheller and Dr. Soukup of ONR for continued support of this program. This work is also partially funded by the Science, Mathematics, And Research for Transformation (SMART) program. Additionally, I wish to thank Dr. Jan Hesthaven of Brown University for his technical oversight and advice.

References

[1] P. Roux, B. Cornuelle, W. Kuperman, and W. Hodgkiss, The structure of raylike arrivals in a shallow-water waveguide, J. Acoust. Soc. Amer., vol. 124, pp. 3430–3439, 2008.

[2] A. Singer, J. Nelson, and S. Kozat, Signal processing for underwater acoustic communications, Communications Magazine, IEEE, vol. 47, pp. 90–96, january 2009.

[3] O. Godin, Restless rays, steady wave fronts, J. Acoust. Soc. Amer., vol. 122, pp. 3353–3363, 2007.

[4] S. Martinelli, An application of the level set method to underwater acoustic propagation, Commun. Comput. Phys., vol. 12, no. 5, pp. 1359–1391, 2012.

[5] S. Osher, L.-T. Cheng, M. Kang, H. Shim, and Y.-H. Tsai, Geometric optics in a phase-space-based level set and Eulerian framework, J. Comput. Phys., vol. 179, pp. 622–648, 2002.

[6] X.-D. Liu, S. Osher, and T. Chan, Weighted essentially nonoscillatory schemes, J. Comput. Phys., vol. 115, pp. 200–212, 1994.

[7] S. Gottlieb and C. Shu, Total variation diminishing Runge-Kutta schemes, Mathematics of Computation, vol. 67, no. 221, pp. 73–85, 1998.

[8] L.-T. Cheng, S. Osher, M. Kang, H. Shim, and Y.-H. Tsai, Reflection in a level set framework for geometric optics, Computer Modeling in Engineering and Sciences, vol. 5, pp. 347–360, 2004.

[9] C. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, ICASE Report 97-65, NASA/CR-97-206253, November 1997.

[10] B. Cockburn, J. Qian, F. Reitich, and J. Wang, An accurate spectral/discontinuous finite-element formulation of a phase-space-based level set approach to geometrical optics, J. Comput. Phys., vol. 208, pp. 175–195, 2005.

[11] H. Forrer and R. Jeltsch, A high-order boundary treatment for Cartesian-grid methods, J. Comput. Phys., vol. 140, pp. 259–277, 1998.

[12] H. Forrer and M. Berger, Flow simulations on Cartesian grids involving complex moving geometries, in Hyperbolic problems: theory, numerics, applications, Vol. I (Zürich, 1998) (R. Jeltsch, ed.), vol. 129 of Internat. Ser. Numer. Math., pp. 315–324, Birkhäuser Verlag, 1999.

[13] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, A PDE-based fast local level set method, J. Comput. Phys., vol. 155, pp. 410–438, 1999.

[14] S. Osher and R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, vol. 153 of Applied Mathematical Sciences. Springer, 2003.

[15] M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys., vol. 114, pp. 146–159, 1994.

[16] M. Galassi *et al.*, GNU Scientific Library Reference Manual. `http://www.gnu.org/software/gsl`, 2011.

[17] F. Jensen, W. Kuperman, M. Porter, and H. Schmidt, Computational Ocean Acoustics. Modern Acoustics and Signal Processing, Springer, 2nd ed., 2011.

[18] B. Gustafsson, H.-O. Kreiss, and A. Sundström, Stability theory of difference approximations for mixed initial boundary value problems. ii, Mathematics of Computation, vol. 26, no. 119, pp. 649–686, 1972.

[19] L. Trefethen, Stability of finite-difference models containing two boundaries or interfaces, Mathematics of Computation, vol. 45, no. 172, pp. 279–300, 1985.

[20] P. Ramachandran and G. Varoquaux, Mayavi: 3D Visualization of Scientific Data, Computing in Science & Engineering, vol. 13, no. 2, pp. 40–51, 2011.