A Level-Sets-Based Wavefront Propagation Method for Underwater Acoustics

by

Sheri Lynn Martinelli

A.B., Colgate University; Hamilton, NY, 1997

M.S., Rensselaer Polytechnic Institute; Troy, NY, 1999

A dissertation submitted in partial fulfillment of the

requirements for the degree of Doctor of Philosophy

in The Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2012

This dissertation by Sheri Lynn Martinelli is accepted in its present form by The Division of Applied Mathematics as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date_____ _____

Jan S. Hesthaven, Ph.D., Advisor

Recommended to the Graduate Council

Date_____ _____

Chi-Wang Shu, Ph.D., Reader

Date_____ _____

William L. Siegmann, Ph.D., Reader

Approved by the Graduate Council

Date_____ _____

Peter M. Weber, Dean of the Graduate School

Vitae

Sheri Lynn Martinelli was born on August 25, 1975 in Norwich, Connecticut.

Education

1999 M.S., Rensselaer Polytechnic Institute, Troy, NY
Applied mathematics

1997 A.B., Colgate University, Hamilton, NY
Mathematics and French

Professional Experience

2007– Computer Scientist
Naval Undersea Warfare Center - Newport, RI
Undersea Systems and Simulation Branch

- Wavefront propagation modeling
- Reverberation modeling for high frequency active SONAR

2001–2007 Computer Scientist
Naval Undersea Warfare Center - Newport, RI
Sensors and Systems Department

- SONAR system performance evaluation
- Probabilistic multi-hypothesis tracking for active SONAR

1999–2001 Systems Engineer II, Raytheon - Portsmouth, RI

- Image processing for mine warfare system
- Software specifications, mine warfare system

Teaching

2010 *Teaching Assistant*, Brown University Statistical Inference I

1999 *Teaching Assistant*, Rensselaer Polytechnic Institute Calculus I

1997 *Teaching Assistant*, Rensselaer Polytechnic Institute Calculus II

Publications

Refereed Publications

S. Martinelli. An application of the level set method to underwater acoustic propagation. *Commun. Comput. Phys.*, 2012. To appear.

Submitted Publications

S. Martinelli. Numerical boundary conditions for specular reflection in a level-sets-based wavefront propagation method. Submitted (2012).

Conference Presentations

S. Martinelli. Range dependence in the level set method for underwater acoustics. In *Proceedings of the 159th meeting of the Acoustical Society of America*, 2010.

S. Martinelli. A method for numerical representation of arbitrary boundaries in acoustic wavefront propagation. In *Proceedings of the 161st meeting of the Acoustical Society of America*, 2011.

S. Martinelli and A. Fredricks. A level set method for high-frequency ocean acoustic propagation. In *Proceedings of the 158th meeting of the Acoustical Society of America*, 2009.

Professional Memberships

Student Member Acoustical Society of America (ASA)

Student Member American Mathematical Society (AMS)

Student Member Society for Applied and Industrial Mathematics (SIAM)

Awards

2009-2012 Science, Mathematics & Research for Transformation (SMART) Scholarship Award

Acknowledgements

I would first like to thank my advisor, Jan Hesthaven, and also Doug Abraham; their words of encouragement helped keep me on track to complete this program. I thank Chi-Wang Shu and Bill Siegmann for agreeing to serve as readers. Dave Drumheller and Ray Soukup of ONR have enthusiastically provided financial support for my research. I have also received support for the last three years from the Science, Mathematics & Research for Transformation (SMART) scholarship program. I am grateful for the support of several colleagues at Naval Undersea Warfare Center (NUWC). Tom Wettergren introduced me to a number of people that helped me gain a broader view of research projects and expertise at NUWC. Kevin Bongiovanni (now at Raytheon BBN Technologies) initially suggested the idea of using level set methods for acoustic propagation. Andrew Fredricks helped guide some of my early work on this project and has been a valuable source of contacts, having arranged opportunities for me to give talks to Professor Siegmann's group at Rensselaer Polytechnic Institute, the Applied Ocean Physics & Engineering (AOPE) department at Woods Hole Oceanographic Institution (WHOI), and to meet with acoustic propagation researchers at the Naval Research Laboratory. Carlos Godoy took me into his group, offering interesting and challenging project work. Dick Phillips helped me navigate through the system, as I am the first NUWC employee to utilize the SMART program as a retention participant. I have also benefitted as a recipient of NUWC's Long-Term Training Fellowship.

Fellow applied math graduate student Xueyu Zhu provided the full wave equation software that I have used to validate some of the wavefront results. The folks at AOPE-WHOI, in particular Art Newhall, have also helped out by providing the RAY software program for ray tracing to validate some results and the sound speed data from the SW06 experiment.

Finally, I wish to thank all the strong women in my family who paved the way for my accomplishments. Especially my grandmother Dorothy, who quit school at sixteen to work. She didn't make it to see her granddaughter become a "doctor", but if there is an afterlife I am certain that she will be bragging about it to her friends for all eternity (along with the accomplishments of her other grandchildren of course). And most of all, I want to thank my mother Linda, whose college aspirations were discouraged; yet that experience inspired her determination to ensure that I would have a college education.

Abstract of " A Level-Sets-Based Wavefront Propagation Method for Underwater Acoustics " by Sheri Lynn Martinelli, Ph.D., Brown University, May 2012

Computer models for underwater sound propagation often rely on ray tracing to obtain approximate solutions to the high frequency wave equation. While adequate for large scale results in deep water environments, ray tracing has drawbacks that limit its reliability for simulations involving multiple boundary interactions and in studies concerning the effects of unmodeled physical processes on propagation. This work presents the development and application of the level set method to the high frequency wave equation with the intent of providing a framework for wavefront propagation in underwater acoustics as a valuable alternative to ray tracing.

Level set methods are generic, computational models for solving dynamic interface problems. The model regularizes problems which exhibit behavior that is difficult to capture numerically in physical space by posing the problem in the higher-dimensional phase space. A relevant example is the behavior of self-intersecting wavefronts; in physical space, such solutions are multiple-valued. The level set method represents the interface implicitly as the zero level set of some function in the phase space, then propagates the surface according to a problem-dependent velocity field. Although the problem is posed in a higher dimensional space, the quantities of interest are local to the interface. Thus the computational burden may be reduced by solving the level set equations only in a neighborhood of the zero level set.

The behavior of solutions at a reflecting boundary motivates the application of WENO methods, which are known to perform well on problems involving shocks. Since WENO methods are complicated to implement on irregular grids, two methods for applying reflection boundary conditions at an interface embedded within a Cartesian grid are proposed. In addition, this work develops a method for computing spreading loss via a semi-Lagrangian approach using the wavefronts provided by the level set solutions. A final topic is the application of a stochastic collocation method to study the effect of random perturbations in the wave speed on the wavefront propagation algorithm. Verification against a full wave equation solver and ray tracing software shows that the resulting method provides accurate wavefront information in the presence of multi-path propagation.

Contents

List of Tables

List of Figures

List of Algorithms

Listings

Chapter One

Introduction

1.1 Context

High frequency wave propagation models are of vital importance to understanding the propagation of sound in shallow water environments. Applications such as acoustic tomography [56] and underwater communications [62] rely on ray tracing for system design and performance prediction. In acoustic tomography, accurate travel time data between a source and receiver over long ranges are used to infer information about ocean currents and temperatures. Both applications have a need for sorting out multi-path time arrivals that occur as a result of ray bending, and surface and bottom scattering. The frequencies useful for such applications in shallow water are at least on the order of kHz and can range into the MHz regime, rendering full wave models impractical in simulations. For very long range propagation, the same applies due to scaling; the high frequency approximation is appropriate whenever the wavelength is much smaller than any length scale. However, 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.

As rays travel from source to receiver, their trajectories spread out in some areas and converge in others. Ray methods, therefore, must include special processing in order to identify the desired paths. Typically in underwater acoustics, one seeks acoustic path(s) between a source and receiver. The solution to this initial-boundary value problem is the set of *eigenrays*, which include all one-way trajectories between

source and receiver. For example, in a shallow water channel, the paths may be organized by the number of surface and bottom bounces. Figure 1.1 depicts such a scenario. In the case of boundary scattering, the acoustic pressure amplitude decreases rapidly with each boundary interaction, so it is common to set a threshold on the amplitude, neglecting paths that fall below threshold before reaching the receiver. One computes each path via a shooting method using a number of *test rays* in order to seek the departure angle from the source that yields a trajectory that ends at the receiver. The problem is that unless the environment is fairly benign, i.e., predictable reflections, and sound speed knowledge, small perturbations in the departure angle may result in large variations in the resulting trajectories. This phenomenon is commonly referred to in underwater acoustics literature as *ray chaos* [63]. When chaotic rays are present, in order to locate eigenrays, one must specify the initial shooting angles to a high degree of precision. In the presence of uncertainty of the input data (source and receiver locations, neglect of currents, imperfect knowledge of the speed of sound), the eigenray solutions may be inconsistent with observed arrival data in experiments. Compounding the problem, when the source and receiver are co-located as in some applications, it is necessary to construct two-way paths by combining all possible pairs of eigenrays. The effects of source and receiver beam patterns combined with the divergence of rays from the source often lead to inadequate reconstruction of the pressure field, especially in the presence of variable bathymetry or surface waves.

Collins and Kuperman [12] suggested an alternative method of computing eigen-

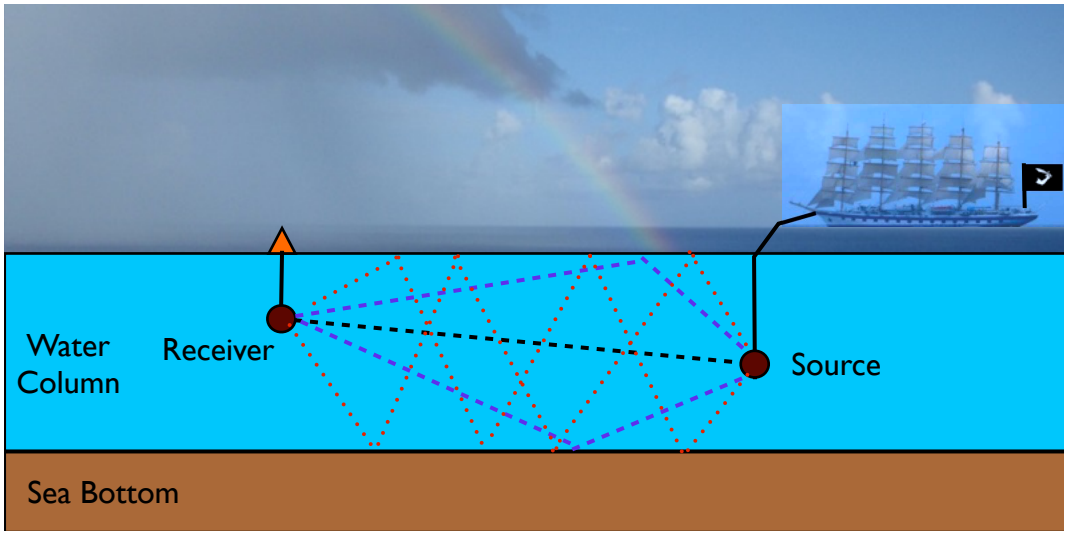


Figure 1.1: Selected ray paths in a sound channel, color-coded by number of surface/bottom interactions [a]

[a] On an unrelated note, the displayed image is composed from photographs taken by the author and a companion in Saint Lucia. The pirate flag depicted is supposedly that of Thomas Tew (died 1695). Tew allegedly had family in Rhode Island and lived in Newport at one time. He later moved to Bermuda and took up piracy (source: Wikipedia). He most likely did not have access to SONAR hydrophones.

rays in the presence of ray chaos by posing a boundary value problem, but their method relied on direct path optimization and did not allow for bottom or surface reflections. More recently, Godin [25] examined the behavior of rays versus that of wavefronts under weak sound speed fluctuations and showed that wavefronts are much more stable than rays, in the sense that the significant ray perturbations tend to occur *along* the wavefronts rather than *across* them. Godin's results suggest that a propagation model based on acoustic wavefronts would be a useful tool for the underwater acoustics community. This type of relative stability of wavefronts is critical for practical simulations.

Other common computational approaches exist which can accurately solve the equations of acoustic propagation. These include normal mode models that solve the full wave equation, and parabolic equation models. Although these methods perform well at low frequencies, they are not appropriate for high frequency propagation where required grid sizes become large enough to overwhelm computational resources. Mode models require the grid to contain at least twice as many points as the highest frequency represented, while parabolic models have a strict restriction on time step for stability ($\Delta t \propto \Delta x^2$ for explicit routines). Furthermore, the derivation of the parabolic wave equation is a result of the paraxial approximation, which does not model scattering back toward the source. Ray tracing is therefore the current standard for high frequency or long range propagation modeling in underwater acoustics.

The focus of this thesis is to develop a solid foundation for acoustic wavefront

propagation using a variation of the level set method, originally presented in [49]. The proposed method builds upon the foundation established by Osher, Cheng, Kang, Shim, and Tsai [47] which introduces a level set method for geometric optics. The resulting equations are linear and hyperbolic, so explicit solvers are stable with $\Delta t \propto \Delta x$. Solving the relevant equations on a fixed grid in the phase space and evolving entire wavefronts in time eliminates the eigenray problem. Thus in complex domains, even though the level set problem is posed in high dimensional phase space, improvement in computational speed may be observed because it is no longer necessary to use a large number of test rays to determine a solution at a given point in space; Figure 1.2 illustrates this idea.

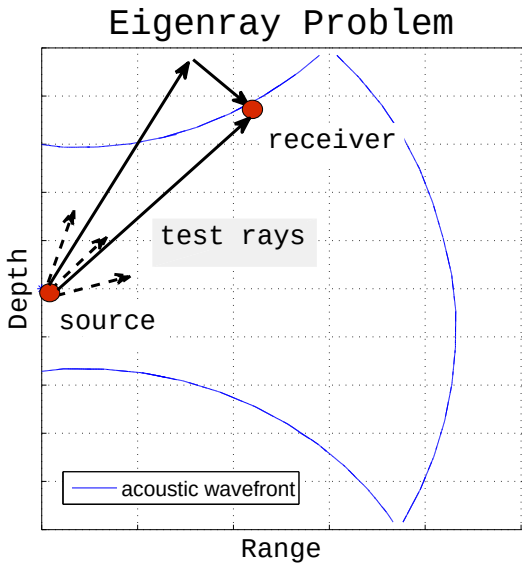


Figure 1.2: The level set method determines arrivals at a given point in space by seeking the wavefront crossing rather than by shooting test rays

1.2 Related Work

1.2.1 Level set methods

Osher and Sethian [49] introduced level set methods as generic, computational techniques for tracking the evolution of moving curves and surfaces. The advantage of this approach is that one may employ standard partial differential equation (PDE) solvers to solve the problem on uniform grids. Refinement of the grid then reduces the global error. Level set methods achieve this by representing the propagating surface implicitly as the zero level set of a function in a higher dimensional space. This zero level set is then transported via the underlying velocity field. In the case of acoustic propagation in isotropic media, the propagation direction is normal to the propagating surface (wavefront). The extension to phase space is critical to the computation of multi-valued solutions in the physical space. Ray tracing handles multi-path propagation naturally, but early fixed grid methods only produced first arrivals. In physical space, the multiply-valued solutions violate well-posedness of the problem. The bicharacteristic curves, an extension of the characteristics to the phase space, are single-valued however and hence working in the phase space, one can capture multi-path arrivals.

Eulerian (fixed grid) geometric optics has been an active research area in the scientific computing community for quite some time, see [2] for a review. The most

similar approach to the level set method is the segment projection method [17] which tracks wavefronts in phase space as projections onto each two dimensional subspace of the three dimensional phase space. This method is effective, but involves complicated bookkeeping in order to reconstruct wavefronts. The approach of Osher, et al. presented in [47] propagates the entire wavefront in the phase space where the bicharacteristics of the eikonal equation are well-behaved. In [52], Qian, et al. build upon [47] by extending the method to propagation in anisotropic materials.

Qian and Leung [53, 54] developed a level set method for the paraxial approximation. Ray trace implementations commonly apply this approximation as it reduces the number of independent equations to solve by propagating in a direction that increases monotonically with time (e.g., range). The approach of Qian and Leung reduces the dimensionality of the problem, but introduces an additional equation to be solved in the phase space to compute arrival times. This method addresses long range propagation problems, but the present work is primarily concerned with reflections and scattering back toward the source, which the paraxial assumption precludes.

1.2.2 Reflection boundary conditions

Cheng, et al. [8] undertook a detailed study of reflection in the level set method to improve the efficiency of the method proposed in [47]. In that work, the authors propose augmenting the grid with the boundary condition, and adjust the CFL

condition 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 to prevent the CFL condition from becoming too restrictive. The development that this thesis presents uses a spatial discretization for which the convergence rate relies on uniformity of the underlying mesh, Chapter 3 provides further motivation for this choice. Given that environmental uncertainty in the ocean will inevitably introduce errors into the model, it is preferable to work with a uniform grid in order to maintain the convergence properties of the method to the extent that is possible.

Cockburn, et al. [9] proposed another related approach to numerical reflection. The authors of this work applied a discontinuous Galerkin expansion to the phase space variable and derived reflection boundary conditions in the coefficient domain. The test problems presented in this work did not expose the cusp that appears in the phase space at a reflected surface however, so it is unclear how the method would perform under more general initial conditions. Chapter 3 of this thesis explicitly addresses the singularity, which motivates the choice of solver.

Forrer and Jeltsch [21] and also Forrer and Berger [20] present a boundary condition for computing reflections that is very similar to the higher order approach studied in the present work, however they used finite volume methods in a fluid dynamics problem, whereas the context of the present study is finite differences. Also, [50] offers a very thorough description of the reinitialization problem that is applied in Chapter 4 of this thesis and in [8] to obtain the signed distance function to the boundary interface and thus the boundary normals.

1.2.3 Underwater acoustics

Most underwater acoustics research in recent years for long range or high frequency problems was conducted based on existing approaches and awareness of the limitations of ray tracing. A couple of older works suggest alternative methods to circumvent ray trace limitations, but have limited applicability. Collins and Kuperman [12] take a variational approach, but do not handle multi-path. Another related work is [19], in which Foreman proposes a method to reconstruct wavefronts given a numerical solution to the Helmholtz equation, thus this approach is only useful for post-processing visualization purposes. Godin's 2007 paper [25] offers perturbation analysis that shows wavefront methods offer stability that ray traces cannot attain. Finally, Finette [18] presents an application of stochastic collocation to underwater acoustic propagation models. Finette's work applies the parabolic solver due to Collins [11] and incorporates a linear regression approach to estimating the statistics. This approach is similar to the approach taken in Chapter 5 of this thesis to the solutions of the level set equations.

1.3 Outline of thesis

This work covers three projects, each relegated to a chapter. Chapter 2 contains relevant background material. The first project, presented in Chapter 3, develops and applies the fundamental method presented in [47] to basic problems of inter-

est in underwater acoustics, namely variable propagation speed, and scattering from boundaries. The contents of Chapter 3 form the basis of [42], which has been accepted for publication in *Communications in Computational Physics*. Chapter 4 develops reflection boundary conditions from objects that do not conform to a Cartesian grid; a manuscript based on this work is currently under review for publication [41]. Chapter 5 presents an application of uncertainty quantification techniques to propagate wavefronts when the wave speed is a function of a random parameter, or a random field. This thesis concludes with a summary of results and suggestions for follow-on work in Chapter 6.

Chapter Two

Preliminaries

The following sections present key concepts and tools used to build a wavefront propagation method for underwater acoustics.

2.1 From fluid dynamics to the Eikonal equation

2.1.1 Linear acoustics

The goal of this thesis is to find an approximate method for solving the linear wave equation of acoustics. There are a number of assumptions required in order to arrive at linear acoustics. The following presents a brief derivation, based on the material in Brekhovskikh and Lysanov [5].

Begin with the hydrodynamic equations of an ideal fluid for the particle velocity $\mathbf{v}$, sound pressure p, and fluid density ρ, all functions of space and time t. These consist of the Euler equation,

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} = -\frac{1}{\rho}\nabla p \tag{2.1}$$

and the continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0. \tag{2.2}$$

A sound wave is a disturbance in the pressure and density, modeled as

$$\begin{aligned} p &= p_0 + p' \\ \rho &= \rho_0 + \rho' \end{aligned} \tag{2.3}$$

where the null subscript indicates an ambient (time-independent) function and the prime indicates a small disturbance in the sense that $p' << p_0$ and $\rho' << \rho_0$. The adiabatic state equation,

$$p = \rho_0 + \rho' \left[\frac{\partial p}{\partial \rho}\right]_{\mathcal{S}} + \frac{1}{2}\left(\rho'\right)^2 \left[\frac{\partial^2 p}{\partial \rho^2}\right]_{\mathcal{S}} + \ldots, \tag{2.4}$$

results from taking $p = p(\rho, \mathcal{S})$ and expanding. The entropy, $\mathcal{S}$, satisfies $\frac{d\mathcal{S}}{dt} = 0$ where $\frac{d}{dt}$ represents the total derivative. The subscript $\mathcal{S}$ in (2.4) indicates that the derivatives are taken under the condition $\frac{d\mathcal{S}}{dt} = 0$.

Substituting these expressions into (2.1) through (2.4), expanding, and retaining only high order terms (note that $\mathbf{v}$ is also a small quantity) reduces to the following:

$$\frac{\partial \mathbf{v}}{\partial t} + \left(\mathbf{v} \cdot \nabla\right)\mathbf{v} = -\frac{1}{\rho_0}\nabla p' \tag{2.5}$$

$$\frac{\partial \rho'}{\partial t} = -\nabla \cdot \left(\rho_0 \mathbf{v}\right) \tag{2.6}$$

$$\frac{1}{c^2}\frac{\partial p'}{\partial t} = \frac{\partial \rho'}{\partial t} + \left(\mathbf{v} \cdot \nabla\right)\rho_0. \tag{2.7}$$

In a homogeneous medium $\nabla \rho = 0$, so the last term in (2.7) is not present.

More generally, take $\frac{\partial}{\partial t}$ of (2.6), multiply (2.5) by ρ_0 and take its divergence, and combine via substitution. Apply (2.7) to obtain

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \rho\nabla\left(\frac{1}{\rho}\nabla p\right) = 0. \tag{2.8}$$

Assume the variation in the density is small and neglect the term $\frac{1}{\rho}\nabla\rho\cdot\nabla p$ to arrive at the linear wave equation for the acoustic pressure:

$$\frac{\partial^2 p}{\partial t^2} - c^2\triangle p = 0. \tag{2.9}$$

2.1.2 High frequency acoustics and the eikonal equation for the phase

The high frequency wave equation results from application of a classical asymptotic approximation to the standard wave equation (2.9). Following [16], start with the d-dimensional linear wave equation for the acoustic pressure in a medium with constant density and variable sound speed function given by $c(\mathbf{x})$ where $(t,\mathbf{x}) \in \mathbb{R}^+ \times \Omega, \Omega \subset \mathbb{R}^d$, and assume that appropriate initial and boundary conditions are available. At very high frequencies, the solutions to this equation exhibit rapid oscillations. In order to be able to reasonably compute these solutions, apply the

geometric optics approximation by assuming a solution of the form

$$p(t, \mathbf{x}) = e^{i\omega S(t,\mathbf{x})} \sum_{k=0}^{\infty} A_k(t, \mathbf{x})(i\omega)^{-k}. \tag{2.10}$$

Upon substitution into the wave equation (2.9), this expression yields the eikonal equation for the phase function S from the highest order terms in ω

$$S_t(t, \mathbf{x}) \pm c|\nabla S(t, \mathbf{x})| = 0 \tag{2.11}$$

and a transport type equation for the first amplitude term

$$(A_0)_t + c(\mathbf{x})\frac{\nabla S \cdot \nabla A_0}{|\nabla S|} + \frac{c(\mathbf{x})^2 \triangle S - S_{tt}}{2c(\mathbf{x})|\nabla S|} A_0 = 0. \tag{2.12}$$

The derivation of transport equations for the remaining amplitude terms follows in a similar manner, however for $\omega >> 1$, only the two leading terms in the expansion are significant; this is the geometric optics approximation.

The weakly coupled system consisting of (2.11) along with (2.12) form the equations of high frequency acoustics.

2.2 The level set method for geometric optics

The eikonal equation (2.11) is a Hamilton-Jacobi type equation with Hamiltonian $H(\mathbf{x}, \mathbf{k}) = c(\mathbf{x})\,|\mathbf{k}|$. The vector-valued variable $\mathbf{k}$ is associated with ∇S. First order partial differential equations of this type may be solved locally via the method of characteristics by expressing (2.11), considering without loss of generality only the equation with the plus sign (propagation outward from the initial wavefront), in terms of the phase space variables $\mathbf{x}$ and $\mathbf{k}$. The Hamilton-Jacobi formulation is the full phase space description of the system, in which the variable $\mathbf{k}$ is the generalized momentum. The characteristic equations are

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \nabla_k H(\mathbf{x}, \mathbf{k}) = c(\mathbf{x}(t))\frac{\mathbf{k}(t)}{|\mathbf{k}(t)|} \\ \dot{\mathbf{k}}(t) &= -\nabla_x H(\mathbf{x}, \mathbf{k}) = -|\mathbf{k}(t)|\nabla_x c(\mathbf{x}(t)) \end{aligned} \tag{2.13}$$

with given, consistent, initial conditions $\mathbf{x}(0) = \mathbf{x}_0, \mathbf{k}(0) = \mathbf{k}_0 = \nabla_x S(0, \mathbf{x}_0)$. Considered as a pair, the bicharacteristic curves $(\mathbf{x}(t), \mathbf{k}(t))$ occupy the $2d$-dimensional phase space, and differentiating with respect to time shows that $H(\mathbf{x}(t), \mathbf{k}(t))$ is constant along them. Conservation of the Hamiltonian ensures that, for $H \equiv 1$,

$$|\mathbf{k}| = |\nabla S| = \frac{1}{c(\mathbf{x})} \tag{2.14}$$

holds.

The ray tracing approach involves computing $\mathbf{x}(t)$ from (2.13), typically using arc length parameterization rather than unscaled time. The difficulty with solving (2.11) on a fixed grid is that, in general, rays may cross, generating multi-valued solutions for the phase function S at some points. Thus standard PDE solvers fail in this case as convergence requires uniqueness of the solution. One of the earlier methods for dealing with this situation was to compute the viscosity solution [14] which forces uniqueness by computing only the first arrival time. However, in applications, multi-valued arrival times are often important. The level set method handles multi-valued arrival times by operating in the phase space, where the bicharacteristics do not suffer from this problem [47].

Ray tracing solves the ordinary differential equations (2.13) along the the trajectory that emanates from a starting point, $\mathbf{x_0}$, on the initial wavefront. Often these equations are parameterized with respect to arc length in cylindrical coordinates in the presence of azimuthal symmetry, (r, z), and the initial conditions are specified by a take-off angle [32]. The level set method instead employs an implicit representation of the wavefront. That is, the level set method embeds the wavefront in the zero level set of a function defined in a higher dimensional space (the phase space), rather than expressing it explicitly in the physical space. Thus, a function that may be multi-valued or otherwise poorly behaved in the physical space is represented by a smoothly varying and well-defined quantity in the phase space. In fact, given the restriction (2.14), it is not necessary to utilize the full phase space in order to find S, since the magnitude $|\mathbf{k}|$ is fixed, the dimension can be reduced by one. For an

example with two dimensional physical space, let

$$\mathbf{k} = \begin{pmatrix} |\mathbf{k}| \cos\theta \\ |\mathbf{k}| \sin\theta \end{pmatrix}, \tag{2.15}$$

then only θ need be considered an independent variable. In this representation, θ is the propagation direction of the wavefront, and $\mathbf{k} = \left(\frac{\cos(\theta)}{c}, \frac{\sin(\theta)}{c}\right)$. So for two-dimensional physical space, the reduced phase space has three dimensions, and for fully three-dimensional acoustics, the reduced phase space is five-dimensional. It is preferable to work in the reduced phase space rather than full phase space for computational efficiency.

In two-dimensional propagation, since the wavefront is a curve in three-dimensional reduced phase space, define a vector-valued function

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}. \tag{2.16}$$

The function $\mathbf{\Phi}$ is the *level set function* [47]. In the phase space for propagation in two-dimensional physical space, $\mathbf{\Phi}$ has two components, $\phi_1(t, \mathbf{x}, \mathbf{k})$ and $\phi_2(t, \mathbf{x}, \mathbf{k})$. One initializes the components such that the initial wavefront is embedded in the intersection of the zero level sets of ϕ_1 and ϕ_2, i.e., the projection onto the physical space of the set $\{(\mathbf{x}, \mathbf{k}) | \phi_1(0, \mathbf{x}, \mathbf{k}) = \phi_2(0, \mathbf{x}, \mathbf{k}) = 0\}$. For full three-dimensional propagation, the reduced phase space has five dimensions and the wavefront is a surface, so the level set function would have three components with the wavefront

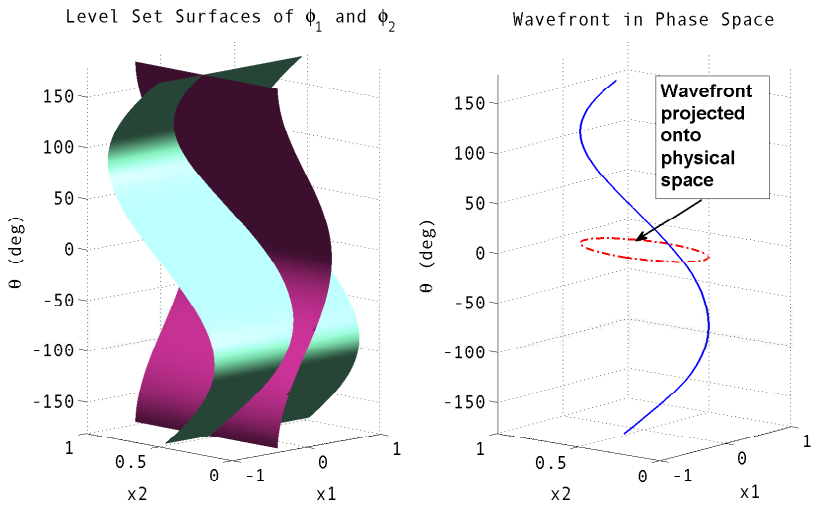


Figure 2.1: Implicit representation of 2D wavefront in the phase space

corresponding to the intersection of the zero level sets of all three component functions [47]. Figure 2.1 provides a simple example in two dimensions of the implicit wavefront representation as the intersection of level set surfaces for two level set functions. Two dimensional propagation is the focus of this thesis; extension to fully three-dimensional propagation is reserved for future work, but follows a similar approach.

To evolve the level set functions, note that the wavefront propagates in the direction of its normal, which is the local ray direction, given by $\frac{\mathbf{k}}{|\mathbf{k}|}$. Consider the zero level set of a component, ϕ_i, evaluated on the ray in phase space, i.e., $\phi_i(t, \mathbf{x}(t), \mathbf{k}(t)) = 0$. Differentiating in time and substituting Equations (2.13) in the result shows that ϕ_i must satisfy the Liouville equation,

$$\frac{\partial \phi_i}{\partial t} + c(\mathbf{x})\frac{\mathbf{k}}{|\mathbf{k}|} \cdot \nabla_x \phi_i - |\mathbf{k}| \nabla_x c(\mathbf{x}) \cdot \nabla_k \phi_i = 0. \tag{2.17}$$

In the reduced phase space for the two dimensional problem, (2.17) reduces to the following transport equation

$$\frac{\partial \phi_i}{\partial t} + \mathbf{V} \cdot \nabla \phi_i = 0, i = 1, 2 \tag{2.18}$$

with the velocity field $\mathbf{V}$

$$\mathbf{V} = \begin{pmatrix} c(\mathbf{x}) \cos\theta \\ c(\mathbf{x}) \sin\theta \\ \frac{\partial c}{\partial x_1} \sin\theta - \frac{\partial c}{\partial x_2} \cos\theta \end{pmatrix} \tag{2.19}$$

where $\mathbf{x} = (x_1, x_2)$ and θ is the direction of propagation in the $x_1 - x_2$ coordinate plane. This velocity field follows directly from the ray equations (2.13). The functions ϕ_1 and ϕ_2 are defined at $t = 0$ so that the known initial wavefront is embedded as the intersection of the zero level surfaces of the two functions. It is convenient to assume that the source has a parameterization in θ so it can be described as

$$x_1 = f_1(\theta)$$
$$x_2 = f_2(\theta).$$

Then ϕ_1 and ϕ_2 can be initialized as

$$\phi_1(0, x_1, x_2, \theta) = x_1 - f_1(\theta)$$
$$\phi_2(0, x_1, x_2, \theta) = x_2 - f_2(\theta). \tag{2.20}$$

For instance, if the wavefront is a circle in the x_1-x_2 plane with radius α centered at some point (x_1^0, x_2^0),

$$\phi_1(0, x_1, x_2, \theta) = x_1 - x_1^0 - \alpha \cos\theta$$
$$\phi_2(0, x_1, x_2, \theta) = x_2 - x_2^0 - \alpha \sin\theta$$

defines such a choice. Letting $\alpha \to 0$ gives an appropriate initial condition for a point source at (x_1^0, x_2^0). In this case, the two level set surfaces are orthogonal, an important property since recovering the wavefront involves seeking the intersection of the level set surfaces. Equation (2.18) represents a decoupled hyperbolic system and can be evolved in time using appropriate numerical techniques. One may recover the wavefront, $W(x_1, x_2; t)$ at any time t as

$$W(x_1, x_2; t) = \{(x_1, x_2) | \phi_1(t, x_1, x_2, \theta) = \phi_2(t, x_1, x_2, \theta) = 0\}. \tag{2.21}$$

Although it appears to be very inefficient to solve a higher dimensional problem, it is not necessary to solve the level set equations over the full $2d-1$ dimensional space. All of the information about the wavefront is concentrated about the intersection of zero level sets of the component functions of $\mathbf{\Phi}$. Thus, following the approach in [50], one need only solve the equations within a tube of grid cells for which $|\phi_i| < \epsilon$ for some threshold ϵ based on the sizes of the mesh and of the stencils for the numerical scheme.

2.3 Ray-based amplitude computation

In underwater acoustics, it is important to be able to compute not only the arrival times of the wavefronts, but the amplitudes as well to properly simulate reverberation or scattering for simulation and testing purposes. The amplitude (alternately expressed as a transmission loss) consists primarily of three components: geometric spreading loss, scattering loss, and attenuation loss. Attenuation causes a plane wave to decay according to an exponential law $A = A(0)e^{-\alpha r}$ due to the conversion of acoustic energy into heat. Given information about an environment, this would be applied as a factor multiplying the other losses. To handle scattering loss (e.g., from surface, bottom, and objects in the environment), one could apply existing models which specify the loss as a function of incidence angle [46]. To incorporate the scattering losses, one can use the angles of incidence and reflection at the boundary which are readily available from the level set data to compute the amplitude boundary condition.

The spreading loss computation poses a greater challenge. An important consideration here is computational speed. In situations where the loss is only required at a few distinct locations in space at a given time, loss data need not be computed everywhere on the wavefront. One might exploit this attribute in order to speed up the algorithm. For other applications, accuracy may be more important than speed, in that full wavefront information might be desired. Methods for computing the spreading loss along a ray have long been established, cf. [16], [32]. In [35], [34], and [38],

the authors show that the related quantity $\rho = \frac{A_0^2}{c^2}$ also satisfies (2.18). However this is a full phase space approach, which significantly increases the computational burden. Instead, this work applies an approach which combines the simplicity of existing algorithms with the wavefront information that the level set method provides.

Recalling (2.12), if the phase function S takes the form $S(t, \mathbf{x}) = \tau(\mathbf{x}) - t$, substituting (2.10) into (2.9) results in the Helmholtz equation and associated frequency domain equation for the phase,

$$|\nabla\tau|^2 = \frac{1}{c^2(\mathbf{x})}, \tag{2.22}$$

and the frequency domain equation for the amplitude:

$$2\nabla\tau \cdot \nabla A_0 + (\nabla^2\tau)A_0 = 0. \tag{2.23}$$

Parameterizing in terms of arc length s defines the ray trajectory $\mathbf{x}(s)$ as $\frac{d\mathbf{x}}{ds} = c\nabla\tau$. Rewriting the transport equation in terms of s gives [32]

$$2\frac{dA_0}{ds} + \left[\frac{c}{\hat{J}}\frac{d}{ds}\left(\frac{\hat{J}}{c}\right)\right]A_0 = 0, \tag{2.24}$$

which has the solution

$$A_0(s) = A_0(0)\left|\frac{c(s)\hat{J}(0)}{c(0)\hat{J}(s)}\right|^{1/2} \tag{2.25}$$

where s is the arc length parameter, and $\hat{J}$ is the Jacobian determinant describing

the spreading of a *ray tube* in terms of the ray arc length, s. This Jacobian has the form $|\frac{\partial \mathbf{x}}{\partial(s,\upsilon,\phi_1)}|$ with υ and ϕ_1 as the initial declination[1] and azimuthal angles of the ray, respectively [32]. Converting to cylindrical coordinates and assuming azimuthal symmetry gives $\hat{J} = r\sqrt{(\frac{\partial z}{\partial \upsilon})^2 + (\frac{\partial r}{\partial \upsilon})^2}$. The key step in computing the amplitude using this approach is to compute the value of the Jacobian. Chapter 3 proposes a simple method to accomplish this.

2.4 Essentially Non-Oscillatory (ENO) and Weighted ENO (WENO) methods

The ENO and related WENO schemes are numerical techniques for approximating function values and by extension, derivatives. This project applies these methods to solve the level set equations (2.18). Finite difference approximations to a function's derivative are based on the use of interpolating polynomials over a subset of the grid (stencil) containing the point at which the derivative is to be evaluated. The derivative of the polynomial approximates the derivative of the underlying function. For smooth functions, the number of points in the stencil determines the order of accuracy of the approximation, e.g., a stencil with $k+1$ points can produce an approximation that converges as $O(\Delta x^k)$. However when a function develops discontinuities, one has to be careful in choosing the appropriate interpolating polynomial.

[1] Typically the declination angle at the source is denoted θ, but υ is used here to avoid confusion with the independent variable θ from the level set method.

In ENO interpolation, introduced by Harten et al. [30] and later refined by Shu and Osher [60, 61], one seeks an appropriate stencil on which to approximate the function with Lagrange polynomials by choosing a stencil that is not likely to contain a point of discontinuity, based on a measure of the function's variation. That is, the stencil is adaptive, versus using a fixed stencil as in a standard interpolation routine. WENO interpolation extends the ENO procedure by combining the results from all candidate stencils to achieve $O(\Delta x^{2k-1})$ accuracy. In Figure 2.2, a simple example indicates the benefit of using the WENO method rather than standard interpolation on a fixed stencil. The plots in Figure 2.2 show the function

$$f(x) = \begin{cases} \frac{1}{2}, & \text{if } x \leq \frac{1}{2} \\ -\frac{1}{2}, & \text{otherwise} \end{cases}$$

and its interpolant, with a grid size of $N = 20$. The upper plot displays the result of a standard third order interpolation on a fixed stencil. The two lines represent interpolations to the left end points of the grid cells and the right end points of the grid cells. Near the discontinuity at $x = \frac{1}{2}$, the interpolating polynomials deviate from the function, and grid refinement will not resolve the error. This effect can lead to oscillations in numerical schemes for partial differential equations. The lower plot displays the results using a third order WENO interpolation, and appears much smoother, thus resulting in a more stable scheme when applied to an equation. Note that it is not necessary to use algebraic polynomials as interpolants.

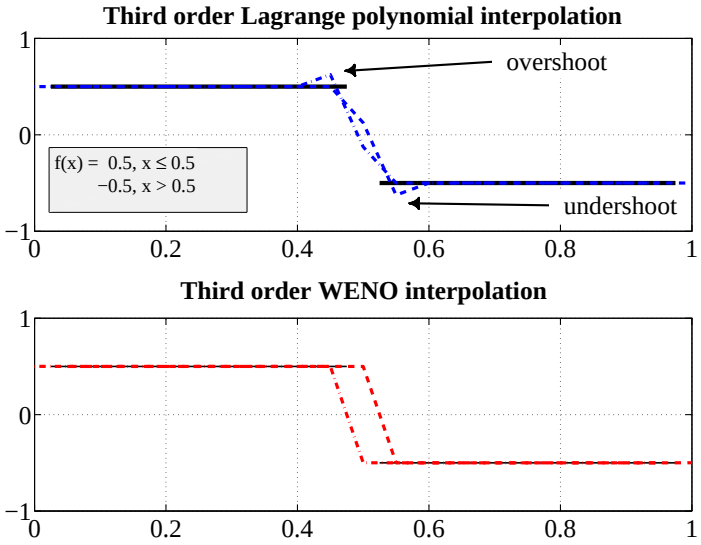


Figure 2.2: Interpolation near a discontinuity

2.4.1 ENO finite difference reconstruction

As in Shu's comprehensive report [59], for polynomial reconstruction in one dimension, establish a uniform grid $\{x_{i-\frac{1}{2}}\}_{i=0}^{N}$, with $\Delta x = x_{\frac{1}{2}} - x_{-\frac{1}{2}}$. That the grid is uniform is critical for the finite difference formulation, but one may relax this requirement by using a finite volume scheme instead, or a smooth transformation to a uniform grid, if available. The method this thesis describes uses finite differences, and hence uniform grids. Define the cells $I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$ for $i = 0, \ldots, N-1$. Assume the values of the function $f(x)$ are available at the cell centers, $\{f(x_i) \equiv f_i\}_{i=0}^{N-1}$. The

goal of ENO is to use these values to construct an approximation, $\hat{f}_{i+\frac{1}{2}}$, such that

$$\frac{\partial f}{\partial x}\Big|_{x_i} = \frac{\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}}{\Delta x} + O(\Delta x^k). \tag{2.26}$$

In the context of the simple conservation law $u_t + u_x = 0$, $\hat{f}_{i+\frac{1}{2}}$ has the role of a numerical flux function for $f(u) = u$.

For each cell I_i, define the k-point stencil $S_r(i)$:

$$S_r(i) = \{x_{i-r}, \ldots, x_{i+s}\}, \tag{2.27}$$

where $r + s + 1 = k$. For example with a fixed stencil, if k is odd, a central approximation would have $r = s$, or an upwind stencil might have $r = s + 2$ if the upwind direction is left to right, and that value of r would remain fixed for each i. In ENO, one begins with a single cell stencil then adaptively adds points depending upon the value of some smoothness measure, e.g., the divided difference, so that r depends on the cell i. The function values f_i are the cell averages of some unknown function, $h(x)$, so that

$$f(x) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi) d\xi. \tag{2.28}$$

Take $H(x) = \int_{-\infty}^{x} h(\xi) d\xi$. Rather than directly approximate $f(x)$, the relationship of the implicitly defined function $h(x)$ to $f(x)$ is $f'(x) = \frac{1}{\Delta x}\left(h(x + \frac{\Delta x}{2}) - h(x - \frac{\Delta x}{2})\right)$, which allows one to define a flux function in terms of the reconstructed values of h at the cell edges, $\left\{h^-_{i+\frac{1}{2}}\right\}_{i=0}^{N-1}$ and $\left\{h^+_{i-\frac{1}{2}}\right\}_{i=0}^{N-1}$. The $-$ and $+$ superscripts indicate

the limits from the left and right, respectively. To find these approximations, seek a polynomial $p_i(x)$ defined on $S_r(i)$ such that p_i approximates h in the sense that

$$\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i(\xi)d\xi = f_i. \tag{2.29}$$

To achieve an estimate satisfying (2.26), require that the polynomial p_i has degree at most $k-1$. Now define the primitive of p_i as $P_i(x) = \int_{-\infty}^{x} p_i(\xi)d\xi$. Then, $P_i(x)$ is related to f as

$$\frac{1}{\Delta x}\left(P_i(x_{i+\frac{1}{2}}) - P_i(x_{i-\frac{1}{2}})\right) = f_i.$$

There is also the relationship

$$h^-_{i+\frac{1}{2}} = p_i(x_{i+\frac{1}{2}}) = P_i'(x_{i+\frac{1}{2}})$$
$$h^+_{i-\frac{1}{2}} = p_i(x_{i-\frac{1}{2}}) = P_i'(x_{i-\frac{1}{2}}).$$

Take $P_i(x)$ to be the Lagrange polynomial interpolating the function $H(x)$ over the $k+1$ points $x_{i-r-1/2}, \ldots, x_{i+s+1/2}$, so P_i satisfies

$$P_i(x) = \sum_{m=0}^{k} H(x_{i-r+m-1/2})\ell^k_{i,r,m}(x).$$

The Lagrange polynomial has the form

$$\ell^k_{i,r,m}(x) = \prod_{\substack{l=0 \\ l\neq m}}^{k} \frac{x - x_{i-r+l-1/2}}{x_{i-r+m-1/2} - x_{i-r+l-1/2}}.$$

Now the function $H(x)$ is only implicitly defined, so to eliminate the dependence on the unknown function, write

$$\begin{aligned} P_i(x) - H(x_{i-r-1/2}) &= \sum_{m=0}^{k} \left(H(x_{i-r+m-1/2}) - H(x_{i-r-1/2}) \right) \ell_{i,r,m}^{k}(x) \\ &= \sum_{m=1}^{k} \left(\sum_{j=0}^{m-1} f_{i-r+j} \Delta x \right) \ell_{i,r,m}^{k}(x), \end{aligned} \tag{2.30}$$

to express $P_i(x)$ in terms of the data, $\{f_j\}$. The extra constant is irrelevant since the quantity of interest is the derivative, $p_i(x)$. Upon differentiating (2.30), rearranging terms, and substituting $x = x_{i+1/2}$ and $x = x_{i-1/2}$, the reconstructed values on the stencil $S_r(i)$ have the form

$$h_{i+\frac{1}{2}}^{-} = \sum_{j=0}^{k-1} c_{rj} f_{i-r+j}, \quad h_{i-\frac{1}{2}}^{+} = \sum_{j=0}^{k-1} \tilde{c}_{rj} f_{i-r+j}, \tag{2.31}$$

where the constants c_{rj} and $\tilde{c}_{rj}$ are grid-dependent as derived from the derivatives of the Lagrange polynomials. Furthermore, given the definitions of $h_{i+\frac{1}{2}}^{-}$ and $h_{i+\frac{1}{2}}^{+}$ and the stencil $S_r(i)$, it is apparent that $\tilde{c}_{rj} = c_{r-1,j}$. Table 2.1 provides the values of c_{rj} as presented in [59] for reference.

Consider the divided differences of the function $H(x)$ at the point $x_{i-1/2}$, the left edge of the cell I_i. The following recursion defines the divided differences of H at this point

$$H[x_{i-1/2}] = H(x_{i-1/2}),\ j = 0,$$

Table 2.1: Constants for reconstruction in (2.31), for $k = 1, 2, 3$

k	r	$j=0$	$j=1$	$j=2$
1	-1	1		
	0	1		
2	-1	$\frac{3}{2}$	$-\frac{1}{2}$	
	0	$\frac{1}{2}$	$\frac{1}{2}$	
	1	$-\frac{1}{2}$	$\frac{3}{2}$	
3	-1	$\frac{11}{6}$	$-\frac{7}{6}$	$\frac{1}{3}$
	0	$\frac{1}{3}$	$\frac{5}{6}$	$-\frac{1}{6}$
	1	$-\frac{1}{6}$	$\frac{5}{6}$	$\frac{1}{3}$
	2	$\frac{1}{3}$	$-\frac{7}{6}$	$\frac{11}{6}$

and for $j > 0$, the j^{th} degree divided difference is

$$H[x_{i-1/2}, \ldots, x_{i+j-1/2}] = \frac{H[x_{i+1/2}, \ldots, x_{i+j-1/2}] - H[x_{i-1/2}, \ldots, x_{i+j-3/2}]}{j\Delta x}. \quad (2.32)$$

If $H(x)$ is sufficiently smooth in the stencil $\{x_{i-1/2}, \ldots, x_{i+j-1/2}\}$, then the divided differences have the property

$$H[x_{i-1/2}, \ldots, x_{i+j-1/2}] = \frac{H^{(j)}(\xi)}{j!},$$

for some point ξ inside the stencil. If there is a point of discontinuity in the stencil, however, then a Taylor series expansion is not valid and

$$H[x_{i-1/2}, \ldots, x_{i+j-1/2}] = O\left(\frac{1}{\Delta x^j}\right).$$

The divided differences of H thus serve as an indicator of the smoothness of H within the stencil. The definition of $H(x)$ along with (2.28) implies that $H(x_{i+1/2})$ satisfies

$$H(x_{i+1/2}) = \Delta x \sum_{j=-\infty}^{i} f_i. \tag{2.33}$$

Using expression (2.33) in (2.32) for $j = 1$ yields

$$H[x_{i-1/2}, x_{i+1/2}] = f_i.$$

By nature of the recursion, the j^{th} divided difference of H can be computed from the $(j-1)^{\text{th}}$ divided difference of the data $\{f_i\}$. Thus, the idea behind the ENO method is to begin with a one-point stencil, $S^{(1)}(i) = \{x_i\}$ and add points to the left or right depending on the value of the divided differences, until the stencil contains $k+1$ points. Algorithm 1 presents pseudocode, assuming data are available for all i, i.e., boundary conditions are provided.

Ultimately the goal is to approximate the spatial operator for the one-dimensional transport equation having the form

$$\frac{\partial f}{\partial t} = -v(x)\frac{\partial f}{\partial x}. \tag{2.34}$$

Algorithm 1 reconstructs the implicitly defined function $h(x)$ at the cell endpoints, but it is still necessary to define an upwind flux in order to approximate the spatial operator, and in particular, $\frac{\partial f}{\partial x}$ at the cell center, x_i. For this application, a basic

Algorithm 1 ENO Reconstruction of $h^+_{i-1/2}$, $h^-_{i+1/2}$

Require: $i \geq 0$, data $\{f_{i+r}\}_{r=-k}^{k}$
  $f[x_i] = f_i$
  for $j = 1, \ldots, k-1$ **do**
    Compute $f[x_i, \ldots, x_{i+j}] = f[x_{i+1}, \ldots, x_{i+j}] - f[x_i, \ldots, x_{i+j-1}]$
    {Since the grid is uniform, the undivided differences may be used}
  end for
  $S^{(1)}(i) = \{x_i\}$
  for $l = 2, \ldots, k$ **do**
    Given $S^{(l)}(i) = \{x_{j+1}, \ldots, x_{j+l}\}$
    if $|f[x_j, \ldots, x_{j+l}]| < |f[x_{j+1}, \ldots, x_{j+l+1}]|$ **then**
      $S^{(1+1)}(i) = \{x_j, \ldots, x_{j+l}\}$
    else
      $S^{(1+1)}(i) = \{x_{j+1}, \ldots, x_{j+l+1}\}$
    end if
  end for
  Construct $h^-_{i+1/2}$ and $h^+_{i-1/2}$ using (2.31)

upwind scheme for (2.34) takes the following form:

$$\frac{\partial f}{\partial t}\Big|_{x_i} = -v(x_i)\frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{\Delta x}. \tag{2.35}$$

where

$$\hat{f}_{i+1/2} = \begin{cases} h^+_{i+1/2}, & v(x_i) \leq 0 \\ h^-_{i+1/2} & v(x_i) > 0. \end{cases} \tag{2.36}$$

2.4.2 WENO finite difference reconstruction

WENO is an extension of ENO, based on the observation that ENO adaptively selects a single stencil for each cell and so effectively uses $2k-1$ cells to obtain order k accuracy. Instead, Liu, Osher, and Chan [39] proposed that all candidate stencils be combined into an order $2k-1$ accurate reconstruction. To accomplish this, define weights on each stencil so that approximations on stencils where the function appears to be smooth are given a significant weight and those on stencils containing a discontinuity are given very small weight. Let $h^{(r)}_{i+1/2}$ represent the reconstruction on the r^{th} stencil $S_r(i) = \{x_{i-r}, \ldots, x_{i-r+k-1}\}$, that is, using (2.31),

$$h^{(r)}_{i+1/2} = \sum_{j=0}^{k-1} c_{rj} f_{i-r+j}, \quad r = 0, \ldots, k-1. \tag{2.37}$$

Then there are weights ω_r satisfying $\omega_r \geq 0$ and $\sum_{r=0}^{k-1} \omega_r = 1$ such that

$$h_{i+1/2} = \sum_{r=0}^{k-1} \omega_r h^{(r)}_{i+1/2} = h(x_{i+1/2}) + O(\Delta x^{2k-1}).$$

A Taylor expansion,

$$h_{i+1/2} = \sum_{r=0}^{k-1} d_r h^{(r)}_{i+1/2}, \quad h_{i-1/2} = \sum_{r=0}^{k-1} d_{k-1-r} h^{(r)}_{i-1/2}, \tag{2.38}$$

approximates $h(x_{i+1/2})$ and $h(x_{i-1/2})$ (by symmetry) with error $O(\Delta x^{2k-1})$. Table 2.2 repeats these constants from [59] for completeness.

Table 2.2: Constants d_r from (2.38), for $k = 1, 2, 3$

k	d_0	d_1	d_2
1	1	-	-
2	$\frac{2}{3}$	$\frac{1}{3}$	-
3	$\frac{3}{10}$	$\frac{3}{5}$	$\frac{1}{10}$

When the stencil contains a discontinuity, the stencil should be assigned a weight $\omega_r << 1$. Thus the weight should be close to d_r in smooth regions, and be moderated by some smoothness factor in non-smooth regions. The weights take the form

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1} \alpha_s} \tag{2.39}$$

for $r = 0, \ldots, k-1$, where

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)}, \quad \tilde{\alpha}_r = \frac{d_{k-1-r}}{(\epsilon + \beta_r)^2}. \tag{2.40}$$

Here β_r is the smoothness factor, and ϵ is just a small constant to prevent division by zero. The choice for β_r given in [33] is based on the variation of the reconstruction polynomial $p_r(x)$ defined by stencil $S_r(i)$ in cell I_i:

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x^{2l-1} \left(\frac{\partial p_r(x)}{\partial x} \right)^2 dx. \tag{2.41}$$

The expressions for computing β_r from the function values $\{f_i\}_{i=0}^{N-1}$ are repeated here from [59] again for reference for $k = 2, 3$. The derivation of these values follows from

the form of the reconstruction polynomial. When $k = 1$, the stencil is fixed, so $\omega_0 = 1$.

For $k = 2$,

$$\beta_0 = (f_{i+1} - f_i)^2$$
$$\beta_1 = (f_i - f_{i-1})^2$$

yields a third order WENO scheme.

For $k = 3$,

$$\beta_0 = \frac{13}{12}(f_i - 2f_{i+1} + f_{i+2})^2 + \frac{1}{4}(3f_i - 4f_{i+1} + f_{i+2})^2$$
$$\beta_1 = \frac{13}{12}(f_{i-1} - 2f_i + f_{i+1})^2 + \frac{1}{4}(f_{i-1} - f_{i+1})^2$$
$$\beta_2 = \frac{13}{12}(f_{i-2} - 2f_{i-1} + f_i)^2 + \frac{1}{4}(f_{i-2} - 4f_{i-1} + 3f_i)^2.$$

yields a fifth order WENO scheme. It is possible to derive higher order WENO schemes as well, however this work implements only third and fifth order WENO. Higher order schemes require larger stencils so there is a trade-off between the computational work and convergence rate, although it is important to note that while higher order methods generally involve more computational work per gridpoint, they do require fewer grid points overall to meet a given error tolerance. Algorithm 2 summarizes the WENO procedure.

Algorithm 2 WENO Reconstruction of $h^+_{i-1/2}$, $h^-_{i+1/2}$

Require: $i \geq 0$, data $\{f_{i+r}\}_{r=-k}^{k}$, $\epsilon > 0$ $\{\epsilon = 10^{-6}$ as implemented$\}$
  for $r = 0, \ldots, k-1$ **do**
    Compute $h^{(r)}_{i+1/2}$ and $h^{(r)}_{i-1/2}$ using (2.31), and Table 2.1
    Compute β_r using the formulae given, or (2.41)
    Compute α_r and $\tilde{\alpha}_r$ using Table 2.2 and (2.40)
  end for
  Compute $\sum_{s=0}^{k-1} \alpha_s$ and $\sum_{s=0}^{k-1} \tilde{\alpha}_s$
  for $r = 0, \ldots, k-1$ **do**
    Set $\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1} \alpha_s}$ and $\tilde{\omega}_r = \frac{\tilde{\alpha}_r}{\sum_{s=0}^{k-1} \tilde{\alpha}_s}$
  end for
  Construct $h^-_{i+1/2} = \sum_{r=0}^{k-1} \omega_r h^{(r)}_{i+1/2}$ and $h^+_{i-1/2} = \sum_{r=0}^{k-1} \tilde{\omega}_r h^{(r)}_{i+1/2}$

Given the reconstructed values, proceed as in the previous subsection to approximate $\frac{\partial f}{\partial x}\big|_{x_i}$ using an appropriate flux function in (2.35).

2.4.3 Time discretization

Applying the WENO (or ENO) procedure as described above leads to the semi-discrete form of (2.34)

$$\frac{\partial f}{\partial t}\Big|_{x_i} = -v(x_i)\frac{\hat{f}_{i+1/2} - \hat{f}_{i-1/2}}{\Delta x} \equiv Q(x_{i+1/2}). \tag{2.42}$$

One requires a suitable scheme to discretize (2.42) in time. For this purpose, consider a specialized class of Runge-Kutta methods. The design of these high-order Runge-Kutta methods ensures that a spatial discretization combined with the first order Euler method for time stepping yields a Total Variation Diminishing (TVD)

approximation. The TVD property is also preserved in the higher order method. Thus, the forward Euler method serves as a building block in the formation of these methods. The total variation of the numerical solution f is

$$TV(f) = \sum_{j=0}^{N-2} |f_{j+1} - f_j|, \tag{2.43}$$

and the TVD property may be expressed as

$$TV(f^{n+1}) \leq TV(f^n). \tag{2.44}$$

In the case of ENO or WENO discretizations for hyperbolic problems, TVD time stepping serves to maintain numerical stability in a higher order method. They are a safe choice for problems containing discontinuities since the TVD property will control the growth of spurious oscillations.

The first order TVD Runge-Kutta (TVDRK) method is the forward Euler method. If Q represents the spatial discretization operator in (2.42), then the Euler method is

$$f^{n+1} = f^n + \Delta t Q(f^n). \tag{2.45}$$

The optimal methods up to fourth order are provided along with proofs in [27]. The optimal second and third order methods are given below for reference. The fourth order method was not implemented for this project; the TVDRK methods of order $k = 1, 2, 3$ are k-stage methods, the fourth order method has five stages.

The second order method is

$$\begin{aligned} f^{(1)} &= f^n + \Delta t Q(f^n) \\ f^{n+1} &= \frac{1}{2} f^n + \frac{1}{2} f^{(1)} + \frac{1}{2} \Delta t Q(f^{(1)}). \end{aligned} \tag{2.46}$$

The third order method is

$$\begin{aligned} f^{(1)} &= f^n + \Delta t Q(f^n) \\ f^{(2)} &= \frac{3}{4} f^n + \frac{1}{4} f^{(1)} + \frac{1}{4} \Delta t Q(f^{(1)}) \\ f^{n+1} &= \frac{1}{3} f^n + \frac{2}{3} f^{(2)} + \frac{2}{3} \Delta t Q(f^{(2)}). \end{aligned} \tag{2.47}$$

The choice of Δt is critical for stability of the fully discrete equation. For the transport problem (2.34) solved with the Euler method (2.45), the Courant-Friedrichs-Levy (CFL) condition for stability (which requires the numerical wave speed to be at least as fast as the physical wave speed) is

$$\Delta t \leq \frac{\Delta x}{\max_x |v(x)|}. \tag{2.48}$$

The second and third order TVD Runge-Kutta methods (2.46) and (2.47) are TVD and hence stable under the same constraint on Δt by construction.

2.4.4 Multiple Dimensions

The previous sections focus on the one-dimensional problem (2.34). However the level set method for acoustics requires solution of equations (2.18) in three-dimensional reduced phase space for a two-dimensional physical problem, or five-dimensional reduced phase space for a three-dimensional physical problem. To accomplish this, apply the schemes dimension by dimension. That is, to apply WENO to a function $f(x, y)$, fix a value of x_i in the grid, and apply Algorithm 2 to $f(x_i, y_j)$ for $j = 0, \ldots, N_y - 1$. This equates to adding an outer loop to the code. Again, a uniform grid is required in each dimension to achieve the advertised accuracy.

Chapter Three

Development of the basic method

3.1 Solving the level set equations for underwater acoustics

The objective of this work is to solve (2.17) for two-dimensional propagation. To accomplish this, solve equations (2.18) in Cartesian coordinates with a uniform line source, parallel to the y-axis as in Figure 3.1. The symmetry of this geometry reduces the wave equation to two dimensions given by (x, z), where $z = 0$ at the surface, and increases with increasing depth. If azimuthal symmetry is present in the domain and sound speed profile, one may obtain a range-depth solution by restricting to the right half of the x-z plane. Take the line source to be located at $(0, z_s)$, then the level set function initialization proceeds by setting

$$\begin{aligned} \phi_1(0, x, z, \theta) &= x \\ \phi_2(0, x, z, \theta) &= z - z_s. \end{aligned} \tag{3.1}$$

Solving the level set equations equates to solving a first order transport equation in the reduced phase space, (x, z, θ):

$$f_t + c(x, z)\cos(\theta) f_x + c(x, z)\sin(\theta) f_z + (c_x \sin(\theta) - c_z \cos(\theta)) f_\theta = 0 \tag{3.2}$$

Section 3.2 offers a discussion of boundary conditions, which provides the moti-

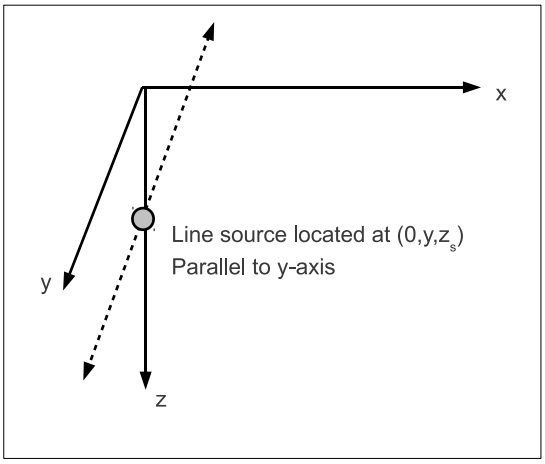


Figure 3.1: Sample problem geometry: a uniform line source parallel to the y-axis

vation to apply WENO and TVDRK solvers to compute (3.2).

3.2 Boundary conditions

Equation (3.2) is a Hamilton-Jacobi equation for which upwind finite differences provide appropriate solutions, as described in [48]. The complications arise from the boundary conditions. The proposed method applies two types of boundary conditions: pure reflection and absorbing. The absorbing condition is a natural implementation; the wavefront simply flows outside of the domain. However, to solve the level set equations in the full domain and for all $-\pi \leq \theta < \pi$ requires imposition of an inflow condition at the boundary.

The θ dimension supports periodic boundary conditions, while in x and z the

upwind direction determines the boundary conditions, discerned from the sign of $\mathbf{V}$, which is fixed given a fixed propagation direction θ. With the upwind methods, it is necessary to impose an inflow boundary condition that specifies function values flowing into the domain. For this, it is sufficient to apply Neumann conditions, for instance.

The domain truncation at inflow will distort the solution near these boundaries, and the effect will worsen over time. For the present application, only the intersection of zero level sets, representing the wavefront, is of interest. Since the wavefront propagates away from the source, the distortion at inflow is sufficiently removed from the zero level set. That is, perturbations to the level set function far from the wavefront do not affect the zero level set. It is important to note that for very long time integration or for very sparse grids, periodic re-initialization of the level set functions may be necessary to prevent distortion of data close to the wavefront. For a discussion of reinitialization, see [47]. A local level set method [50] can improve efficiency. This technique exploits the fact that the values of the level set functions far from the wavefront are not of interest in order to reduce the required amount of computational work by solving the level set equations only within a band of grid points containing the wavefront.

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, is of greater interest. To impose a reflection boundary condition at, say, the

surface $\{z = 0\}$, set

$$\begin{aligned} \phi_1(t, x, 0, \theta_{refl}) &= \phi_1(t, x, 0, \theta_{inc}) \\ \phi_2(t, x, 0, \theta_{refl}) &= \phi_2(t, x, 0, \theta_{inc}) \\ \theta_{refl} &= -\theta_{inc} \end{aligned} \tag{3.3}$$

in accordance with Snell's Law. Here, θ_{inc} and θ_{refl} are, respectively, the angles incident upon and reflected from the surface. For a general boundary, the condition would be [8]

$$\theta_{refl} = 2\theta_B - \theta_{inc} - \pi, \tag{3.4}$$

where θ_B is the angle of the outward normal to the reflecting surface. This construction, under the velocity field given by (2.19) results in a sharp cusp at the boundary that grows like $2c_0t$ in the case of a constant sound speed $c(x, z) = c_0$, at vertical incidence. To see this, note that under a constant wave speed, (2.18) has the solutions

$$\begin{aligned} \phi_1(t, x, z, \theta) &= x - c_0 t \cos\theta \\ \phi_2(t, x, z, \theta) &= z - z_s - c_0 t \sin\theta. \end{aligned}$$

Figure 3.2 illustrates the level set function at a reflecting boundary location as a function of θ. Vertical incidence is a worst-case scenario; at less severe angles, the vertical component of the velocity field, $c_0 \sin\theta$, has reduced magnitude. However, the singularity will worsen in severity for higher sound speeds and for longer periods

of time integration. Such features may produce spurious oscillations in standard finite difference solutions. Actually, the simplest upwind method, forward Euler time integration with first order upwind spatial differencing, is dissipative. In this case, if the source is sufficiently far away from the reflecting boundary, the oscillations are not apparent, but this method will not work when a source is near the boundary and hence it is not practical. This issue motivates the use of WENO methods coupled with TVDRK time integration in this work, c.f. Chapter 2. Because of the cusp singularity, it is also necessary to be careful about interpolating in θ near the boundary. For this purpose, Appendix A derives coefficients for WENO interpolation at an arbitrary point.

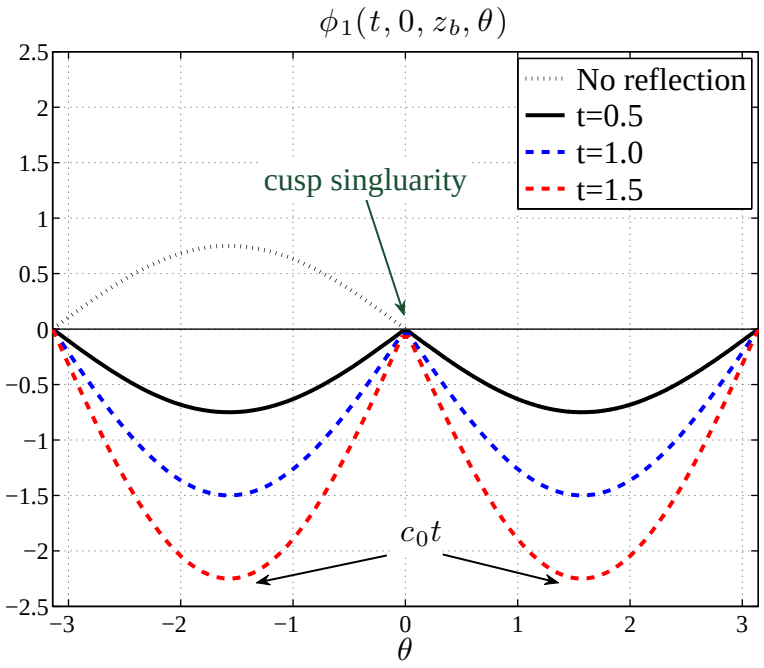


Figure 3.2: A cusp forms at a reflecting boundary, increasing in severity over time

WENO methods are high order; a WENO method based on k candidate stencils

achieves $(2k-1)^{\text{th}}$ order accuracy, whereas the ENO method using the same number of points in the stencil will be a k^{th} order method [59], that is, the numerical solution on a grid with size Δx spacing converges to the true solution like Δx^k as the grid is refined. While higher order methods generally involve more computational work per grid point, there is a trade-off in that high order solvers require fewer grid points to meet a given error tolerance. First, second, and third order TVDRK methods compute the time derivatives. The WENO solvers implemented for this work have orders $\nu = 1, 3$, or 5. The subsequent examples applied a combination of fifth order WENO and third order TVDRK. It is worth noting that Cheng proposes a semi-Lagrangian approach to propagating the level sets in phase space in [7]. In the present work, better performance was observed from the higher order WENO methods when the source was located close to a reflecting boundary.

The other matter with respect to boundary handling is the fact that the boundary location might not be in the grid. One could use a non-uniform grid, but this affects the accuracy of the underlying WENO method. Instead, experimentation has shown that approximating the location of the boundary by the nearest grid points yields convergence. Chapter 4 offers a closer study of this method and of another higher order method that employs a local reflection principle.

3.3 Amplitude Implementation

Recall that for the two-dimensional propagation problem in the x-z plane, the level set method produces two level set functions $\phi_1(t, x, z, \theta)$ and $\phi_2(t, x, z, \theta)$ in the reduced phase space $\subset \mathbb{R}^2 \times [-\pi, \pi]$, where θ gives the propagation direction of the wavefront (normal to the wavefront). Solving the level set equations for ϕ_1 and ϕ_2, one may extract the wavefront from the intersection of the zero level sets of ϕ_1 and ϕ_2. The objective is to use this information to compute the solution given by (2.25). The development proceeds under the assumption that no reflections occur; that is reserved for future extension.

Approach

Let $W(t) = \{\mathbf{x} | \phi_1(t, \mathbf{x}, \mathbf{k}) = \phi_2(t, \mathbf{x}, \mathbf{k}) = 0\}$ be the wavefront as extracted from the level set method at time t. The goal is to compute the spreading loss at a given point $\mathbf{x}(t) \in W(t)$, where $\mathbf{x}$ is in a Cartesian coordinate system, $\mathbf{x}(t) = (x(t), y(t), z(t))^{\mathrm{T}}$. To solve the amplitude problem, consider the scenario in three dimensions with azimuthal symmetry and spherical spreading. It is also possible to treat cylindrical spreading (line source) with appropriate modification to the Jacobian. Suppose the loss is known along the wavefront at the previous time step,

$W(t-\Delta t)$, and identify $\mathbf{x}(t-\Delta t)$ with $\mathbf{x_0}$. Then in light of the above, [16]

$$A(\mathbf{x}(t)) = A(\mathbf{x_0})\frac{c(\mathbf{x}(t))}{c(\mathbf{x_0})}\sqrt{\left|\frac{J(0)}{J(\Delta t)}\right|}. \tag{3.5}$$

In this notation, t is the ray parameter, and $\mathbf{x}(t)$ is the ray trajectory, that is, the point on the wavefront at the current time t evolved according to the ray direction from the point $\mathbf{x}(t-\Delta t) = \mathbf{x_0}$. The Jacobian in this formulation is based on a time parameterization. It is convenient to use time as a parameter here since the level sets formulation is in the time domain. With the initial condition at time $t-\Delta t$, the Jacobian is evaluated for one time step. By definition, $J(0) = 1$. If $c(\mathbf{x}(t))$ is constant, then the rays are simply radial lines directed away from the source.

If the sound speed varies slowly with respect to the time evolution of the wavefront, it is reasonable to approximate the sound speed as locally constant. Assume, without loss of generality, that the acoustic source is located at the origin. Let $\mathbf{x}$ be a point on the wavefront at which to compute the loss, at some time $t = n\Delta t$, with $n > 1$ a positive integer. To find the point on the wavefront along that ray at the previous time corresponding to $\mathbf{x}$, apply a ray trace backward in time one time step, using an ODE integrator (this is essentially a semi-Lagrangian scheme). Call this point at the previous time step $\mathbf{x_0}$. Now let $\mathbf{y} = \mathbf{x} - \mathbf{x_0}$, and let $\bar{c} = c(\mathbf{x_0})$ be the wave speed at the location $\mathbf{x_0}$, assumed to be locally constant in a neighborhood of $\mathbf{x_0}$. Then this ray has the exact form

$$\mathbf{y}(t) = \mathbf{y_0} + \bar{c}t\frac{\mathbf{y_0}}{|\mathbf{y_0}|} \tag{3.6}$$

and the Jacobian matrix is

$$\mathbf{I} + t\frac{\bar{c}}{|\mathbf{y_0}|}\left(\mathbf{I} - \frac{\mathbf{y_0}\mathbf{y_0}^{\mathrm{T}}}{|\mathbf{y_0}|^2}\right). \tag{3.7}$$

The point $\mathbf{y_0} \neq 0$ is an arbitrary point along the ray from the origin to $\mathbf{y}$. The Jacobian determinant J of the transformation from $\mathbf{y_0}$ at time $t = 0$ to $\mathbf{y}(t)$ is

$$J(t) = \left(1 + t\frac{\bar{c}}{|\mathbf{y_0}|}\right)^2, \tag{3.8}$$

Expression (3.8) describes spherical spreading of rays in a medium with constant wave speed, and its value depends only on the distance of the starting location from the origin (source), and the distance traveled by a particle along the ray. It is independent of the actual location in space of the starting point (except that it cannot be evaluated directly for a ray starting at the origin).

To translate back to $\mathbf{x}$ coordinates, take $\mathbf{y_0} = \epsilon\frac{\mathbf{y}}{|\mathbf{y}|}$ where $\epsilon > 0$ is small. Then $\mathbf{y_0}$ maps to a starting point $\mathbf{x_0} + \epsilon\frac{\mathbf{x}-\mathbf{x_0}}{|\mathbf{x}-\mathbf{x_0}|}$. Evaluate (3.8) at $t^* = \Delta t - \frac{\epsilon}{\bar{c}}$ so that the

evaluation point maps to the desired location $\mathbf{x}$, and let $\epsilon \to 0$, yielding

$$J(\Delta t) = \lim_{\epsilon \to 0} \left(1 + \left(\Delta t - \frac{\epsilon}{\bar{c}} \right) \frac{\bar{c}}{\left| \mathbf{x_0} + \epsilon \frac{\mathbf{x} - \mathbf{x_0}}{|\mathbf{x} - \mathbf{x_0}|} \right|} \right)^2$$
$$= \left(1 + \frac{\Delta t \bar{c}}{|\mathbf{x_0}|} \right)^2. \tag{3.9}$$

Substituting the above into (3.5) yields

$$A(\mathbf{x}(t)) = A(\mathbf{x_0}) \frac{|\mathbf{x_0}|}{|\mathbf{x_0}| + \bar{c}\Delta t}. \tag{3.10}$$

Since $c(\mathbf{x})$ is assumed to be constant along the ray from $\mathbf{x_0}$ to $\mathbf{x}$, the term from (3.5) involving the sound speed cancels.

Equation (3.9) is only valid locally, so consider N rays corresponding to wavefront locations at time t given by $\{\mathbf{x}^i\}_{i=1}^N$. These locations correspond to N locations on the wavefront at time $t - \Delta t$, given by $\{\mathbf{x_0}^i\}_{i=1}^N$. Let c_i denote the constant sound speed value for ray i. To compute the spherical spreading, express $\mathbf{x_0}^i = r_0^i \left(\cos \upsilon^i \cos \phi^i, \cos \upsilon^i \sin \phi^i, \sin \upsilon^i \right)$, with $r_0^i = |\mathbf{x_0^i} - \mathbf{x_s}|$ where $\mathbf{x_s}$ is the source location. Then the spreading loss at the point $\mathbf{x}^i(t)$ along the wavefront is approximately

$$A(\mathbf{x}^i(t)) = A(\mathbf{x_0}^i) \frac{r_0^i}{r_0^i + c_i \Delta t}, \ i = 1, \ldots, N. \tag{3.11}$$

One may estimate the value of $A(\mathbf{x_0^i})$ by interpolating the known values along the wavefront. For instance, if the wavefront is in the form of a circle in the x-z plane,

express $A(x,z)$ as

$$A(x,z) = A((x-x_s)\cos(\theta), (z-z_s)\sin(\theta)), \tag{3.12}$$

where (x_s, z_s) is the source location, then interpolate in θ to obtain $A(\mathbf{x_0^i}) = A(\theta_0^i)$.

In the case of cylindrical spreading (line source), the Jacobian $J_{\text{cyl}}(\Delta t)$ takes the analogous form

$$J_{\text{cyl}}(\Delta t) = 1 + \frac{\bar{c}\Delta t}{|\mathbf{x_0}|}. \tag{3.13}$$

The corresponding expression for the amplitude is

$$A(\mathbf{x}^i(t)) = A(\mathbf{x_0^i})\sqrt{\frac{r_0^i}{r_0^i + c_i\Delta t}}. \tag{3.14}$$

Initialization

The preceding derivation assumes that the function $A(\mathbf{x})$ is known on $W(t-\Delta t)$. That information allows one to step forward in time, evolving the amplitude along with the level sets, or to apply the method as post-processing once wavefronts have been computed. At $t = 0$, let the initial condition to the wave equation be a point source located in space at $\mathbf{x_s} = (x_s, y_s, z_s)$, or in the 2D formulation in the x-z plane, let the source be located at $\mathbf{x_s} = (x_s, z_s)$. Assume there is some initial, known, source level, A_{source}. The Jacobian is not defined at the source, so another technique is

necessary to initialize the procedure. For this purpose, proceed as in [32]. The exact solution for the pressure field due to a point source in an unbounded, homogeneous medium with sound speed c_0 is proportional to

$$p(\mathbf{x}(t)) = -S_\omega \frac{e^{i\omega t}}{4\pi c_0 t}. \tag{3.15}$$

The value S_ω represents a known source strength, in terms of the surface displacement of a small, spherical source, and is proportional to A_{source}. Given the general form of the pressure,

$$p(\mathbf{x}) = A(\mathbf{x}) e^{i\omega\tau(\mathbf{x})}, \tag{3.16}$$

one may deduce that

$$A(\mathbf{x}(\Delta t)) = -S_\omega \frac{1}{4\pi c_0 \Delta t}. \tag{3.17}$$

Thus, to initialize the spreading loss computation, apply (3.17). For all subsequent times, one proceeds as in the previous subsection.

Algorithm

An outline of the algorithm for computing the amplitude using the ray-based procedure for two-dimensional propagation in the x-z plane follows below. Suppose there is a known set of points $\{\mathbf{x_0}^i\}_{i=0}^N$ along a wavefront at a time t_0, the values of $A(\mathbf{x_0}^i(t_0))$ for $i = 1, \ldots, N$, and a set of points along the evolved wavefront, $\{\mathbf{x_1}^i\}_{i=0}^M$ at time $t_1 = t_0 + \Delta t$ is available, the following steps compute $A(\mathbf{x_1}^i(t_1))$, $i = 1, \ldots, M$.

Let $\mathbf{x_1}^i = (x_1^i(t), z_1^i(t))$. Then,

1. Solve the backward ray equations in the time domain:

$$\dot{x}(t) = c(\mathbf{x}(t)) \cos\theta(t)$$
$$\dot{z}(t) = c(\mathbf{x}(t)) \sin\theta(t)$$
$$\dot{\theta}(t) = \frac{\partial c}{\partial x} \sin\theta(t) - \frac{\partial c}{\partial z} \cos\theta(t)$$

   subject to $x^i(t_1) = x_1^i$ and $z^i(t_1) = z_1^i$. Also assume $\theta^i(t_1)$ for $i = 1, \ldots, M$ is available from the level set data. For example, apply the Euler method backward in time as

$$x^i(t_0) = x_1^i - c(\mathbf{x_1}^i)\Delta t \cos\theta^i(t_1)$$
$$z^i(t_0) = z_1^i - c(\mathbf{x_1}^i)\Delta t \sin\theta^i(t_1)$$
$$\theta^i(t_0) = \theta^i(t_1) - \Delta t V_\theta$$

   where

$$V_\theta = \frac{\partial c}{\partial x}|_{\mathbf{x_1^i}} \sin\theta^i(t_1) - \frac{\partial c}{\partial z}|_{\mathbf{x_1^i}} \cos\theta^i(t_1).$$

   Note that the sound speed is assumed to be smooth so that these derivatives exist. It might be tempting to use a higher order solver, however the general case may require a nonlinear solve depending on the form of c. A first order method is used here due to its simple description.

2. Compute the range from the acoustic source to $\mathbf{x}^i(t_0)$ for $i = 1, \ldots, N$:

$$r_0^i = \sqrt{(x^i(t_0) - x_s)^2 + (z^i(t_0) - z_s)^2},$$

where (x_s, z_s) is the source location.

3. If $t_1 = \Delta t$ (first step only), compute the arc length

$$s^i = \int_{t_0}^{t_1} c(\mathbf{x}^i(t))\mathrm{dt}$$

for each ray, $\mathbf{x}^i(t)$.

4. Consider the amplitude along the wavefront at t_0 to be a function of the phase space variable θ, $A = A(\theta(t))$ and interpolate to find $A(\theta^i(t_0))$ for each i. This step breaks down when the wavefront bounces or runs outside of the domain. Addressing that case is set aside for future work.

5. If $t_1 > \Delta t$, set

$$A(\mathbf{x_1}^i) = A(\mathbf{x_0}^i)\frac{r_0^i}{r_0^i + \Delta t c(\mathbf{x_0}^i)}.$$

Otherwise, set

$$A(\mathbf{x_1}^i) = A_{\text{source}}\frac{1}{s^i}.$$

6. Step forward in time and repeat.

Remark

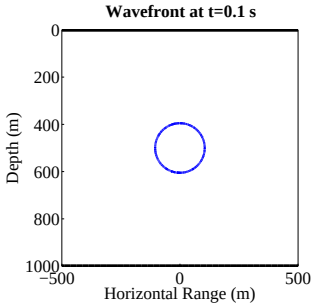
The proposed approach to computing the amplitude has very nice stability properties. Assuming that one applies a linear (or other well-behaved) interpolation scheme is applied to the amplitudes at time $t-\Delta t$, the result $A(\mathbf{x}(t)) \leq A(\mathbf{x}(t-\Delta t))$ holds for any Δt. This is a standard result for semi-Lagrangian type methods. However if Δt is allowed to be too large, the accuracy will suffer from the constant wave speed approximation, and the same issues that appear with ray methods. This result will not apply when A blows up due to a caustic; handling of caustics is a challenge left to future work, possibly by adapting the approaches of [51] or [68]. Both of these works use a Gaussian beam model to propagate the spreading loss; the GRAB (Gaussian Ray Bundles) [68] approach uses an empirically derived threshold to prevent blow-up at caustics. Since the proposed method is based on the Lagrangian form, these approaches could be applied directly as an enhancement. Caustics do not represent a numerical instability, but are a consequence of the high frequency limit in the geometric optics approximation. By construction, this method will not lead to infinite amplitudes, but special treatment will be necessary to produce accurate approximations in the vicinity of and beyond caustics.

3.4 Results

This section presents some computational results to show convergence of the wavefronts computed using the level set method, and to compare the solutions to those of the full wave equation.

3.4.1 Comparison to exact solutions

The test cases in this section are those for which analytical solutions are available to study convergence. In particular, results for two sound speed profiles are presented in the absence of reflections from surface or bottom. The results are based on implementation of a fifth order WENO scheme coupled with third order TVD Runge-Kutta time integration on an $N \times N \times N$ grid. Comparison is made with the exact solutions after 0.1 seconds and errors are reported in the max norm, evaluated in a fixed neighborhood of the wavefront. The theoretical convergence rate is not observed immediately, but even for comparatively sparse grids, observed convergence is faster than first order.



N	Error (max norm)	Effective Order
20	3.306945E-03	–
40	8.509789E-04	1.96
80	1.164058E-04	2.87
160	2.695106E-06	5.43
320	2.312644E-09	10.19

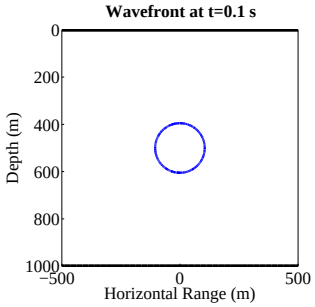
Figure 3.3 & Table 3.1: Accuracy analysis 1: c = constant

Isovelocity profile

This example uses $c = 1.0$ km/sec, independent of location in space, and a point source at $z = z_s = 0.5$ km with $x \in [-1, 1]$ and $z \in [0, 1]$. In this case the characteristics are straight lines so $\theta(t) = \theta_0$ and the solutions to the level set equations are $\phi_1(t, x, z, \theta) = x - t\cos(\theta)$, and $\phi_2(t, x, z, \theta) = z - z_s - t\sin(\theta)$. Figure 3.3 presents the results for the L^∞ error and the extracted wavefront with $N = 40$, evaluated at $t = 0.1$ seconds and without reflection. Initially, the solution appears to be converging like $\frac{1}{N^2}$. The expected fifth order convergence is observed at $N = 160$, then the error decreases even more rapidly as N increases to 320.

Linear profile

An exact solution to the level set equations is available for the case of a profile linear in depth: $c(z) = \alpha z + \beta$, where α and β are constants. For this example, $\alpha = 0.5$ and $\beta = 1.0$ are chosen, where units are again in km/sec. The initial



N	Error (max norm)	Effective Order
20	5.253662E-03	–
40	3.423236E-03	0.62
80	4.712438E-04	2.86
160	1.382231E-05	5.09
320	5.375073E-08	8.01

Figure 3.4 & Table 3.2: Accuracy analysis 2: $c(z)$ linear

condition is a point source at $z = z_s = 0.5$ km with $x \in [-1, 1]$ and $z \in [0, 1]$.

The exact solution for this case is (away from $\theta = \frac{\pi}{2}$)

$$\phi_1(t, x, z, \theta) = x + \hat{\gamma}\left(z + \frac{\beta}{\alpha}\right)\frac{1 - e^{2\alpha t}}{1 + \hat{\gamma}^2 e^{2\alpha t}} \tag{3.18}$$

$$\phi_2(t, x, z, \theta) = \left(z + \frac{\beta}{\alpha}\right)\frac{(1 + \hat{\gamma}^2)\, e^{\alpha t}}{1 + \hat{\gamma}^2 e^{2\alpha t}} - \frac{\beta}{\alpha} - z_s \tag{3.19}$$

where

$$\hat{\gamma} = \tan\left(\frac{\theta}{2} + \frac{\pi}{4}\right). \tag{3.20}$$

A derivation is included in Appendix B. Figure 3.4 presents the results of this investigation in L^∞ error along with the wavefront extracted for $N = 40$.

The observed convergence rate is somewhat slower in this case than for the isovelocity profile, but still fifth order convergence is observed at $N = 160$ and rapidly begins to approach the true solution for larger N. The slower convergence in this case is likely due to the fact that the isovelocity problem is actually a two-dimensional problem in the phase space, since the direction of travel, θ, is constant and hence

the θ component of the velocity field is zero. Upon introduction of a variable sound speed, the problem becomes fully three-dimensional.

3.4.2 Wavefronts with grid-conforming reflection boundary

This section presents computational results for profiles with reflecting boundaries. The examples were computed using a 50 x 50 x 50 grid. The format for presenting the results in Figures 3.5 and 3.6 is of time snapshots of the wavefront superimposed on contours from a full wave equation result for the same profile and domain geometry. Including the amplitude contours from a finite frequency full wave equation solver serves as a check on the accuracy of the solutions. This software, provided by Xueyu Zhu of Brown University, uses a discontinuous Galerkin approach with Perfectly Matched Layers to solve the wave equation. The sound speed profiles were selected for theoretical illustration and are not realistic ocean profiles.

Example with reflection and long time integration

Figure 3.5 provides an example that approximates propagation of a wavefront in a shallow water channel with source location at half the depth of the water column. The sound speed is constant, $c(x, z) \equiv c_0 = 1.5$ km/sec, and solutions propagate out to $t = 4.0$ seconds. The purpose is to show the stability of the wavefronts computed using the level set method over a longer time period.

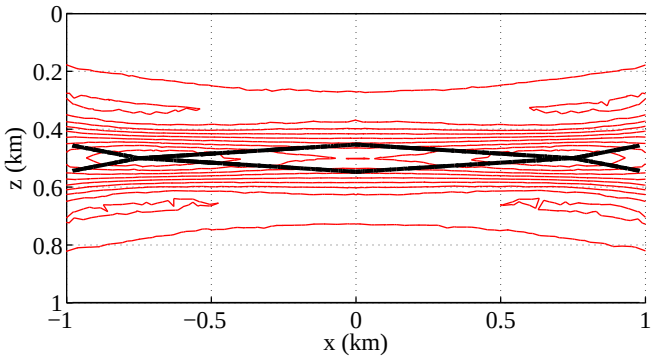


Figure 3.5: Wavefront (thick, black curve) computed using level sets superimposed on full field amplitude contours: constant wave speed $c(x, z) = 1.5$ km/sec, $t = 4.0$ seconds

Linear in depth and range profile with flat, reflecting bottom

Figure 3.6 applies a profile linear in both x and z: $c(x, z) = 0.5x + 0.5z + c_0$ to the same scenario as in Figure 3.5. This example serves to illustrate that sound speed range dependence is a natural feature of this algorithm. The full wave equation solver is a finite frequency solution so it does not produce wavefronts exactly, but the wavefront is located at the leading edge of where the contours appear to be converging together, corresponding to a rapid change in amplitude. The thick, black line representing the wavefront computed at each specified time step lies on the leading edge of the converging contours, as expected, even after multiple reflections.

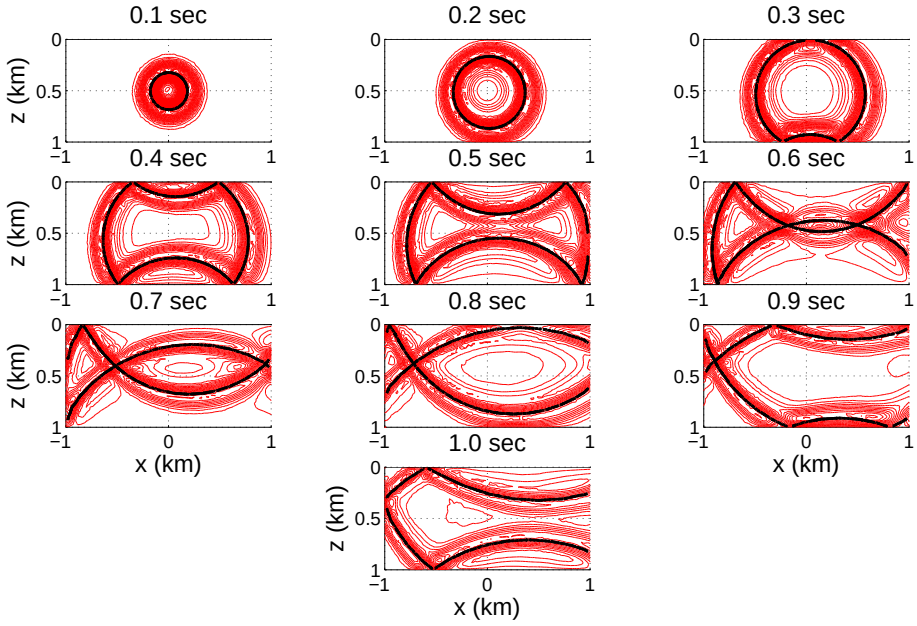


Figure 3.6: Time snapshots of wavefronts (thick, black curve) computed using level sets, superimposed on full field amplitude contours: $c(x, z) = 0.5x + 0.5z + c_0$ km/sec

3.4.3 Examples with variable sound speed

In the first example in this section, a profile that simulates a waveguide presents an interesting comparison between the ray trace and wavefront models. The model is an acoustic analog of the quantum harmonic oscillator, discussed in more detail by Foreman [19]; this profile yields a normal mode solution that is tractable at low frequencies, but the behavior becomes more complicated in the high frequency limit. A source is placed at $z_s = 0.5$ km in a field with $c(z) = \frac{c_0}{\sqrt{1-\left(1-\frac{z}{z_s}\right)^2}}$, with a sound speed of $c_0 = 1.5$ km/sec (Figure 3.7) at the source depth. This scenario represents a waveguide, symmetric about the sound channel axis at $z = z_s$, as $c(z)$ approaches infinity at the channel boundaries $z = 0$ km and $z = 1$ km. The rays are oscillating functions about the sound channel axis. In the wavefront model, this corresponds to a periodically self-crossing wavefront. Figure 3.8 shows snapshots of the wavefront computed using the level set technique with wavefronts extracted from a ray trace superimposed for verification. The match is very good away from the source. The errors are greater closer to the source, but this can be resolved by using a finer grid in the level set implementation.

The next example uses data processed at Woods Hole Oceanographic Institute (WHOI) as part of the Shallow Water 2006 (SW06) experiment and is used with permission from the authors [43]. The data have already been processed and contain values at 15 depths over time; there is no range (or x) dependence and the time variability is small. A cubic spline fit provides a smooth representation of the data

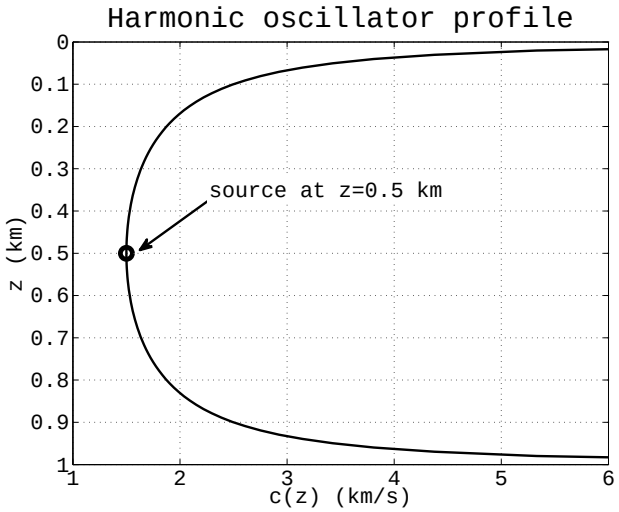


Figure 3.7: Sound speed as a function of depth for harmonic oscillator profile

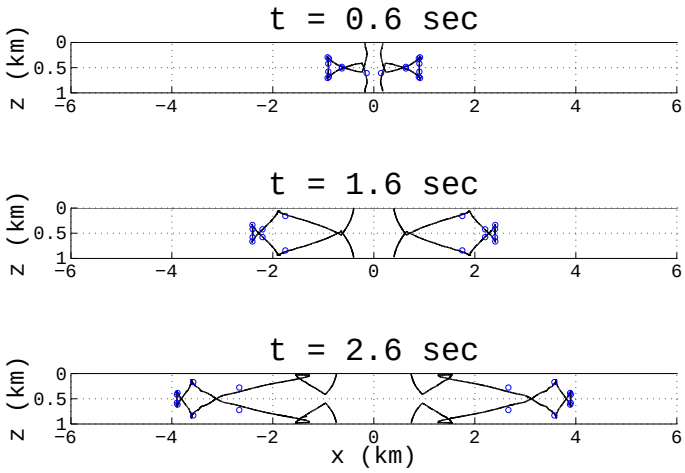


Figure 3.8: A subset of the time snapshots of wavefronts computed using level sets: $c(z) = \frac{c_0}{\sqrt{1-\left(1-\frac{z}{z_s}\right)^2}}$, with wavefronts extracted from an analogous ray trace appearing as circles superimposed on the level sets wavefronts

for the estimates of $\frac{\partial c}{\partial z}$. Figure 3.9, displays one time sample of the profile data, fitted with a spline interpolation.

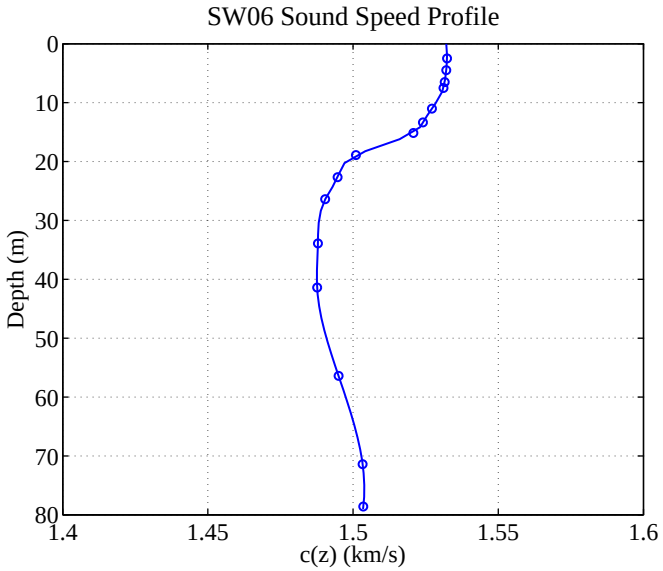


Figure 3.9: Sound speed data from SW06 experiment

Figure 3.10 shows the wavefronts resulting from the level set method at times $t = 0.1, 0.2, 0.3$, and 0.4 seconds, on an $N \times N \times N$ grid with $N = 30$. The water is very shallow, only about 79 meters deep, so the wavefronts spread very quickly. The source location is $(0, z_s = 30 \text{ m})$.

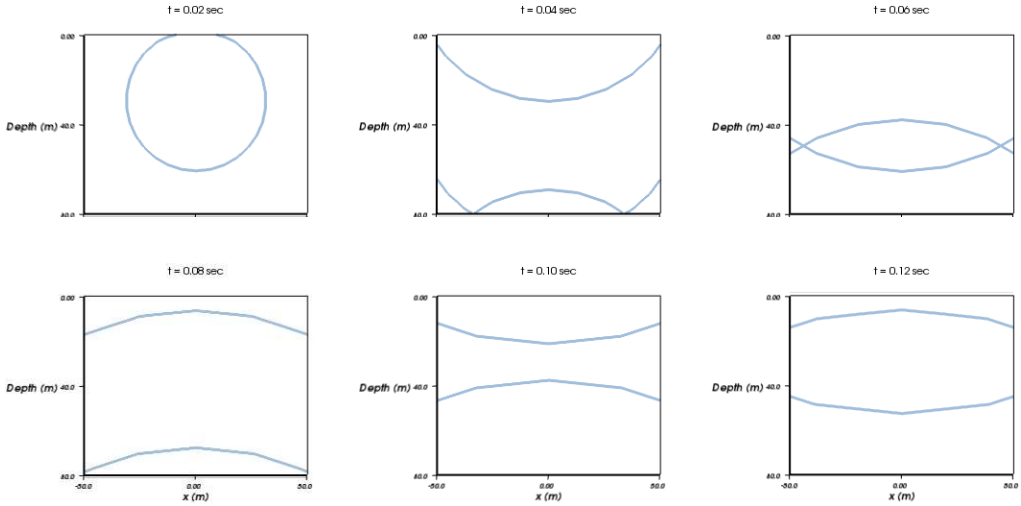


Figure 3.10: Time snapshots of wavefronts computed using level sets with $c(z)$ from the SW06 sea test sound speed data

3.4.4 Examples with scattering

Scattering from bottom features

The following examples explore range dependence a little further by introducing rectangular (grid-conforming) features in the domain. In order to avoid overcomplicating the scenarios, the sound speed profile was held constant at $c = 1.5$ km/sec. Figure 3.12 shows a rectangular scatterer placed on the "ocean" bottom. As would be expected from a pure geometric optics approximation, the level set method does not simulate the diffraction at the corner. Away from the corner, the results agree nicely with the full wave equation model, however a spurious zero level set appears in the phase space when the wavefront reflects off of the object. To see this, consider

a plane wave source as in Figure 3.11. Recall that the wavefront is represented in the model as the intersection of zero level sets of two functions. Thus the function is negative on one side of the front, say the leading part and positive on the other (lagging). When part of the wavefront reflects off the object, the positive values reflect back in the opposite direction in the phase space, but the part that does not encounter the object retains the same sign in that direction, creating a discontinuity. Because there is a change of sign over this discontinuity, a zero level set emerges as an artifact. One could apply edge detection to remove this artifact, but that is not terribly reliable as an approach because it is difficult to distinguish numerically between a discontinuity and a steep gradient in a smooth function. Instead, since this is a non-physical effect, a proper computation of the amplitude, when incorporated, should assign negligible weight to those locations along the computed wavefront.

Scattering from an obstacle

This example, in Figure 3.13, demonstrates a potentially useful application for an acoustic level set method. The rectangular geometry is kept for the sake of conforming to the grid, but this time the scatterer is an object in a domain with absorbing boundaries, which could represent an object in the deep ocean. The point source is at $z_s = 0.5$ km and the sound speed is again $c = 1.5$ km/sec. This example also suffers from the same effects as the above examples (i.e., the spurious zero level sets upon reflection). However this example really demonstrates how this method changes the point of view for a scattering problem from the ray-based model to

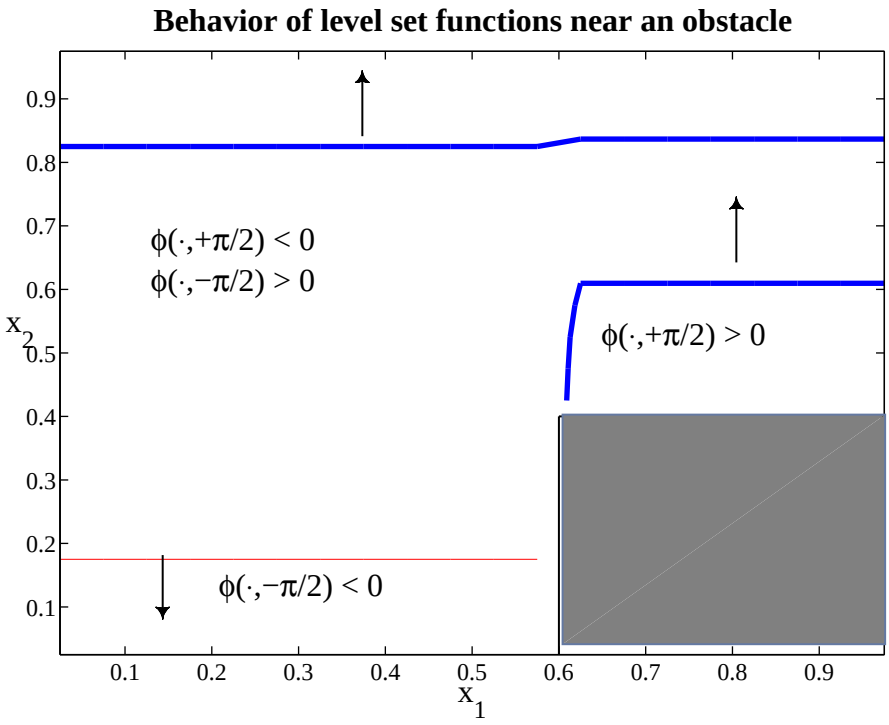


Figure 3.11: A scatterer produces a spurious zero set

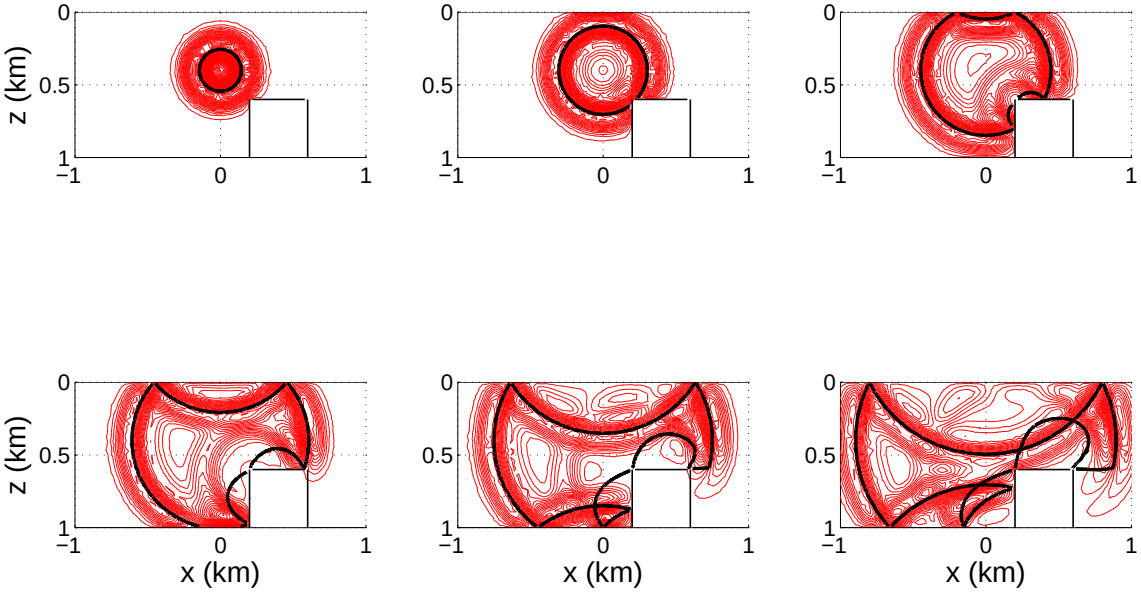


Figure 3.12: Time snapshots of wavefronts (thick, black curve) computed using level sets in the presence of a rectangular bottom scatterer

having the ability to fully capture the shape information in acoustic reflection from an object. A ray-based approach involves the specification of locations on the object, dependent upon its orientation if it's not spherically symmetric, to which to trace eigenrays in order to simulate a response. Using a wavefront model, one can fully utilize the object's shape information and avoid having to compute eigenrays.

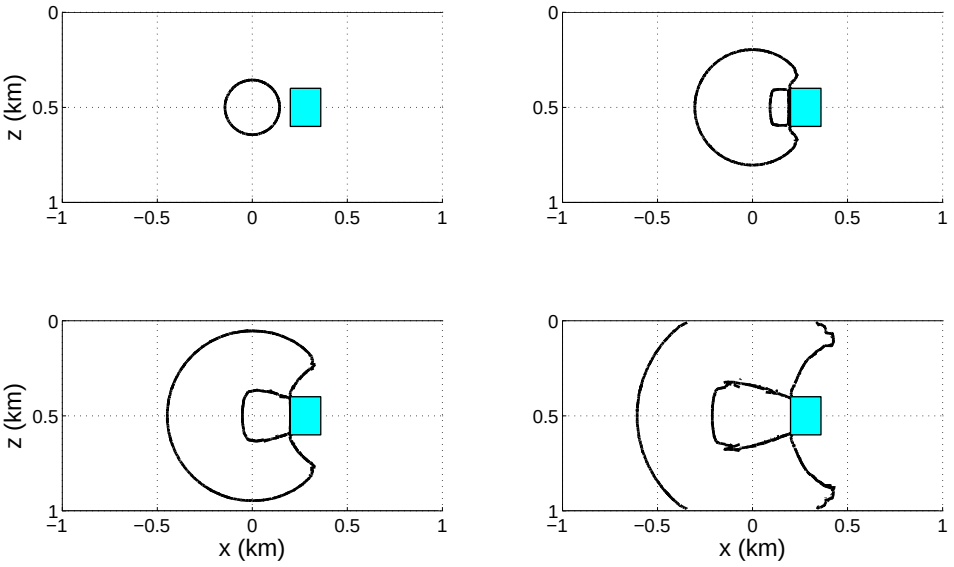


Figure 3.13: Time snapshots of wavefronts computed using level sets in the presence of a scattering obstacle in the domain

3.4.5 Verification of the spreading loss method

This section presents a few examples to show convergence of the wavefronts computed using the level set method. A few test cases confirm that the proposed algorithm for computing the spreading loss from the wavefronts resulting from the

level set method procedure produces an acceptable result. A comparison with the theoretical spreading loss in an isovelocity environment (Figure 3.14) with $c = 1.5$ km/sec shows excellent agreement with the method outlined in section 3.3. In this case, the constant sound speed assumption holds everywhere, so the method should produce an exact solution.

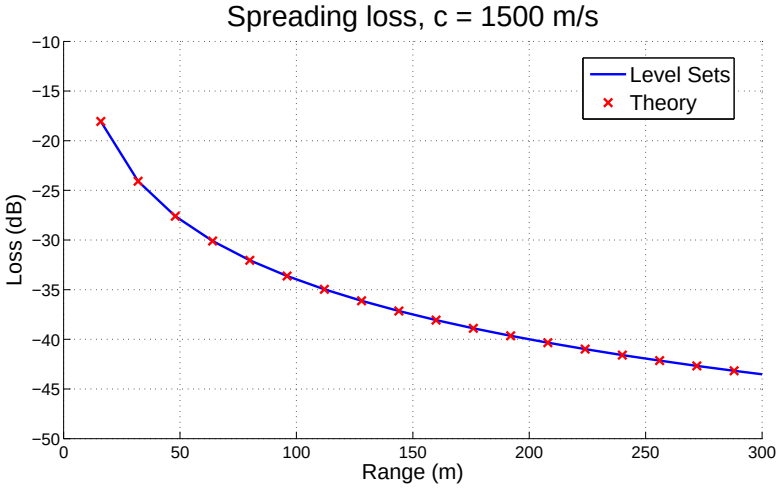


Figure 3.14: Level set method wavefront amplitude as a function of range compared to theoretical result

Further comparisons were also performed using the amplitudes computed using the RAY [3] software package. Each case locates a point source at $z_s = 0.5$ km with $z \in [0, 1]$ km. Under the assumptions of no surface or bottom bounces (deep water only) and short time propagation (so no crossing rays), wavefront data were extracted and accumulated over time and the amplitudes interpolated to produce Figures 3.15, 3.17, and 3.19. Accompanying these figures, Figures 3.16, 3.18, and 3.20 show good agreement in direct comparisons of the fields computed along a horizontal slice at the source depth. Figures 3.15 and 3.16 were generated with an isovelocity

environment ($c = 1.5$ km/sec). Figures 3.17 and 3.18 result from a linear (in z) sound velocity given by $c(z) = 0.5z + c_0$, with $c_0 = 1.5$ km/sec. Figures 3.19 and 3.20 display results using the sound velocity profile $c(z) = \frac{c_0}{\sqrt{1+2.4z/c_0}}$, with $c_0 = 1.55$ km/sec.

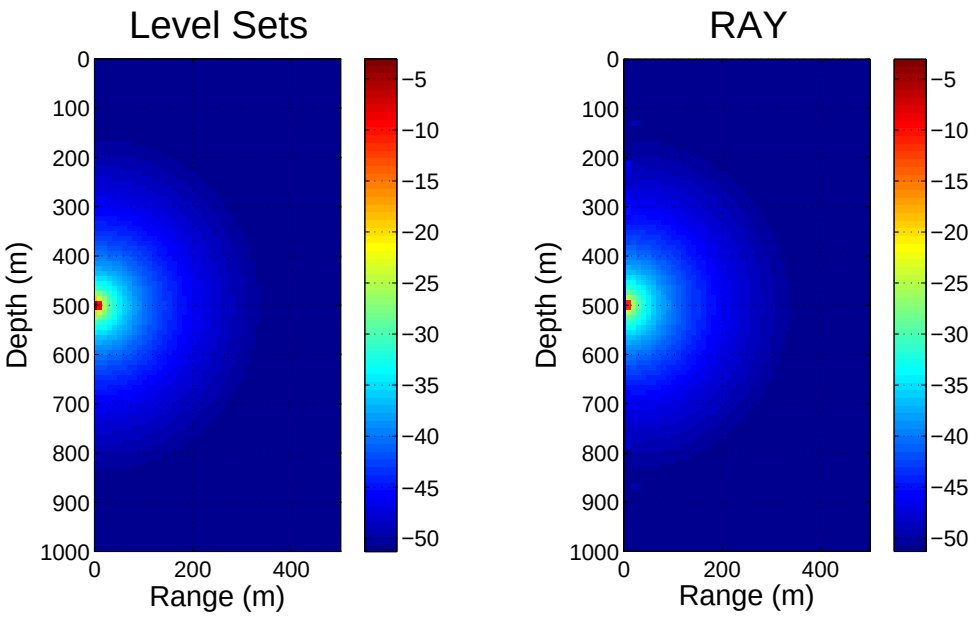


Figure 3.15: Spreading loss computed using Level Set Method (left) compared to extracted loss from RAY (right) for a constant sound speed

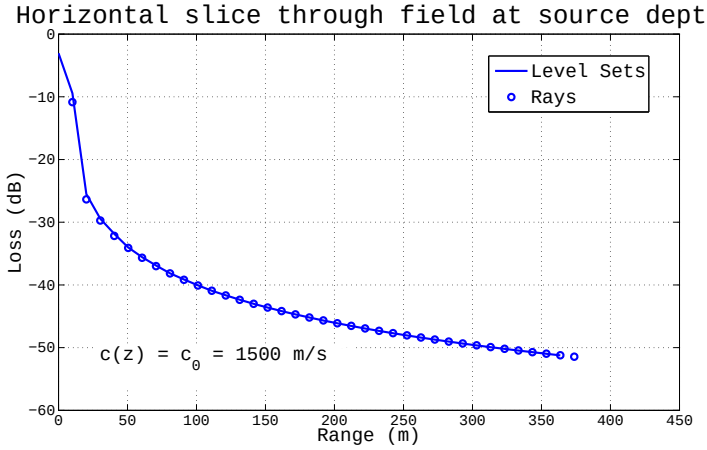


Figure 3.16: Spreading loss verification - horizontal slices through the fields shown in Figure 3.15 at source depth

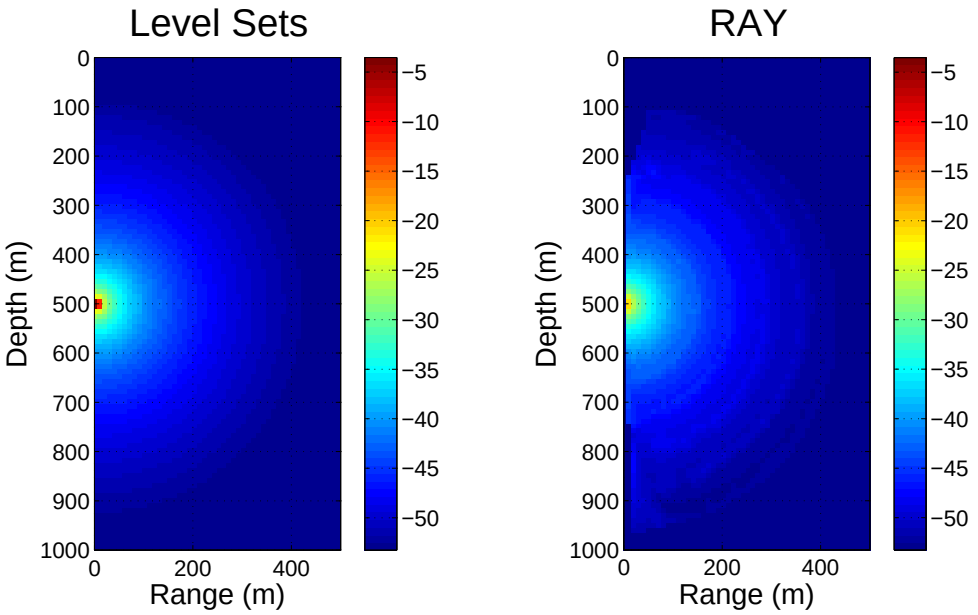


Figure 3.17: Spreading loss computed using Level Set Method (left) compared to extracted loss from RAY (right) for a sound speed function varying linearly in depth

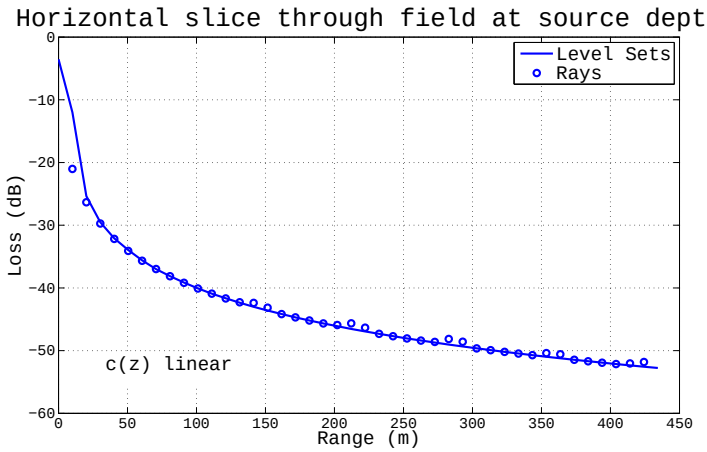


Figure 3.18: Spreading loss verification - horizontal slices through the fields shown in Figure 3.17 at source depth

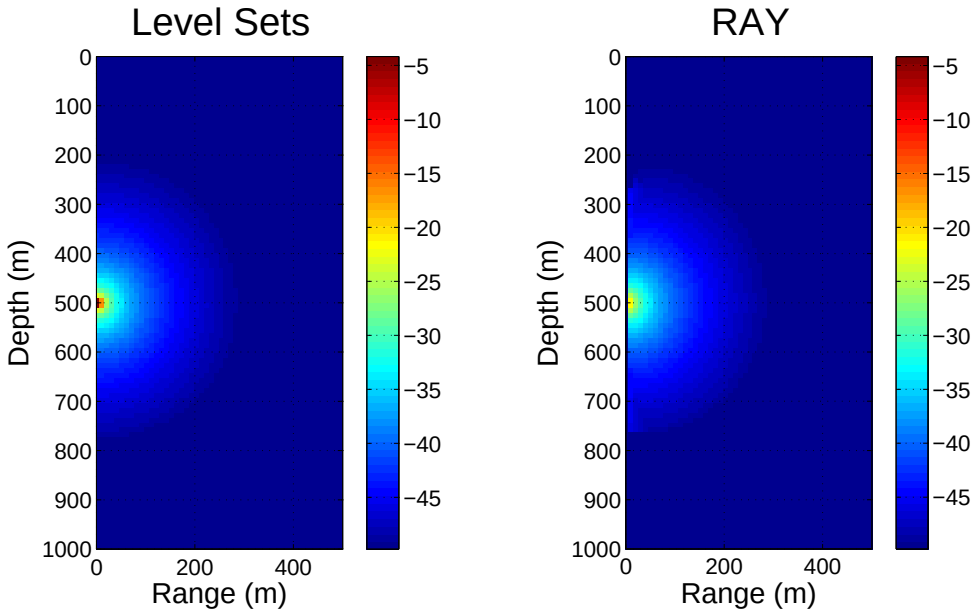


Figure 3.19: Spreading loss computed using Level Set Method (left) compared to extracted loss from RAY (right) for a sound speed function with square inverse varying linearly in depth (n^2 linear)

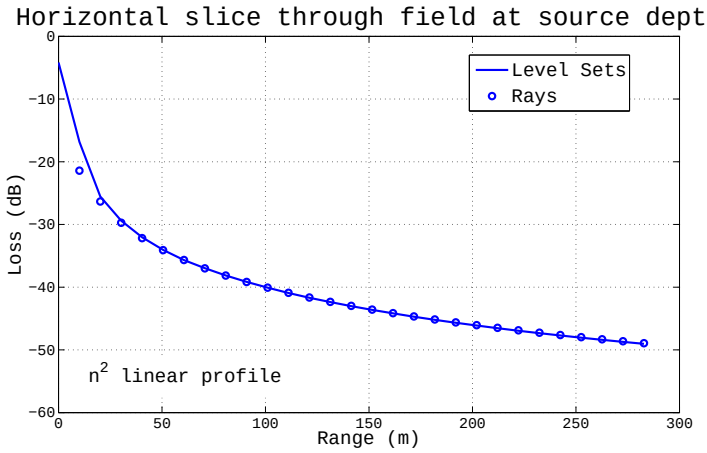


Figure 3.20: Spreading loss verification - horizontal slices through the fields shown in Figure 3.19 at source depth

Chapter Four

Reflection boundaries

This chapter studies reflection boundary conditions for boundaries that do not conform to any uniform, Cartesian grid. Recall that WENO convergence results rely on grid uniformity. In Chapter 3, all examples with reflecting boundaries were designed so that the boundaries conform to the grid. To extend the method to produce practical results, it is necessary to accommodate more general domain geometries. One way to do this is to fit a mesh to the given geometry using triangulation, say, in the physical space. The disadvantage of this approach is that the WENO implementation in the phase space becomes complicated, thus it is preferable to seek a way to work within the Cartesian grid where WENO implementation is more straightforward. The two methods discussed in this chapter take the approach of locally setting boundary conditions at the nearest in-domain mesh location that are consistent with reflection at a nearby interface. An auxiliary problem is that of computing the normals to the reflected surface. For this, Section 4.1 describes an approach based on a separate level set method, based on the approach in [8].

4.1 Level set representation of the boundary and precomputing the normals

As in [48], [50] and [8], an additional 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

$$
\begin{aligned}
x_i &= -1 + (i + \frac{1}{2})h, \\
z_j &= (j + \frac{1}{2})h, \\
\theta_k &= -\pi + \pi(k + \frac{1}{2})h, \text{ and} \\
h &= \frac{2}{N}.
\end{aligned}
$$

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

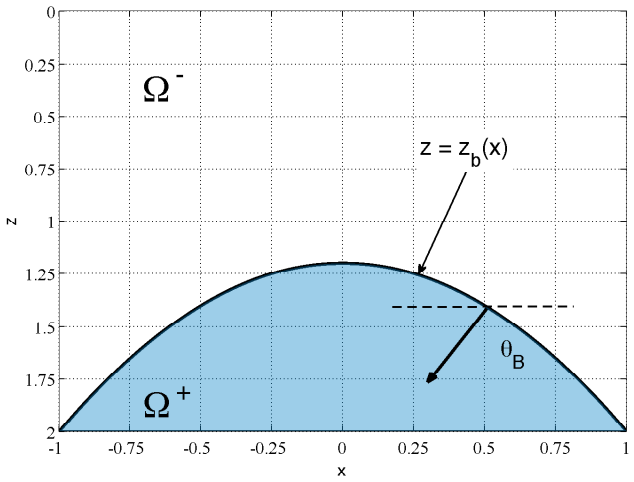


Figure 4.1: Domain geometry imposed on Cartesian grid, $z = z_b(x)$ is the boundary interface, and θ_B is its outward normal

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 $\Omega^- = \Omega - \Gamma - \Omega^+$. Figure 4.1 illustrates this scenario.

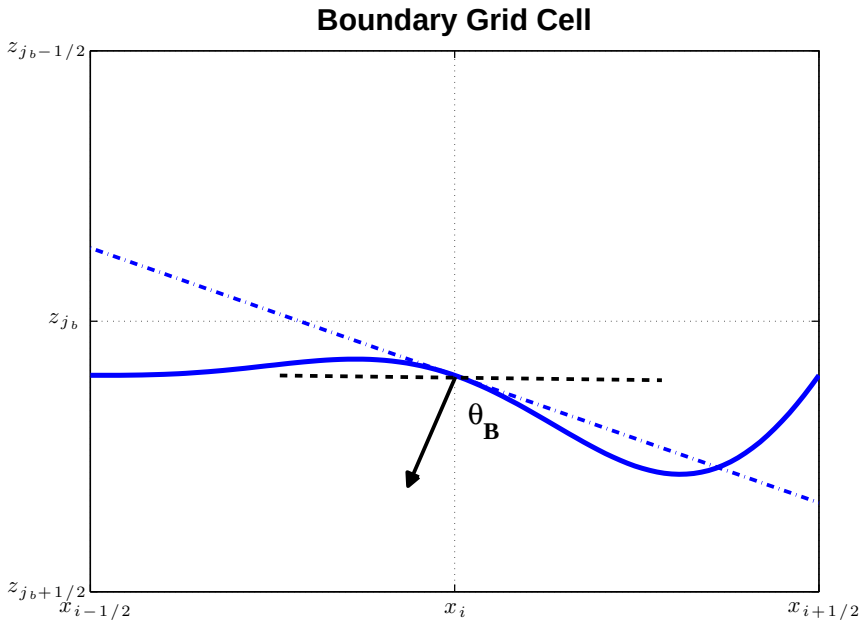


Figure 4.2: A grid cell intersecting the boundary

The objective is to compute an approximation to $\theta_B(x_i)$, the angle of the outward normal to the interface at the center of each grid cell along the boundary $I_{i,j_b,k}$, $i = 0, 1, \ldots, N-1$; j_b is the index of the grid cell containing $z_B(x_i)$ (see Figure 4.2). This computation occurs in the physical domain (x-z), so for now, ignore the k index.

Let $\mathbf{x}$ be a vector in the x-z plane. Define a distance function, $\hat{D}(\mathbf{x})$ by

$$\hat{D}(\mathbf{x}) = \min_{\mathbf{x}_I \in \Gamma}(|\mathbf{x} - \mathbf{x}_I|). \tag{4.1}$$

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

1. $|\hat{d}(\mathbf{x})| = \hat{D}(\mathbf{x})$

2. $\hat{d}(\mathbf{x}) = 0$ when $\mathbf{x} \in \Gamma$

3. $\hat{d}(\mathbf{x}) = \hat{D}(\mathbf{x})$ when $\mathbf{x} \in \Omega^+$

4. $\hat{d}(\mathbf{x}) = -\hat{D}(\mathbf{x})$ when $\mathbf{x} \in \Omega^-$

5. $|\nabla \hat{d}(\mathbf{x})| = 1$

6. $\mathbf{N} = \nabla \hat{d}$ is the unit normal to the interface $\partial\Omega$.

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

To construct $\hat{d}(x, z)$, Sussman, Smereka, and Osher [65] suggested solving the following reinitialization equation

$$\hat{d}_\tau + S(\hat{d}_0)\left(|\nabla \hat{d}| - 1\right) = 0 \tag{4.2}$$

to steady state, where $S(\cdot)$ is a sign function and $\hat{d}_0$ is an initial value of $\hat{d}$. The term *reinitialization equation* is 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 to a signed distance function near the interface (cf. [50]). Peng et al. [50] refined the method by replacing $S(\hat{d}_0)$ in (4.2)

with the smoothed out function

$$s(\hat{d}) = \frac{\hat{d}}{\sqrt{\hat{d}^2 + \hat{d}_x^2 \Delta x^4 + \hat{d}_z^2 \Delta z^4}} \tag{4.3}$$

to improve convergence. To initialize the method, one may apply

$$\hat{d}_0(x, z) = z_b(x) - z, \tag{4.4}$$

as a rough estimate, then apply a fifth order WENO scheme to compute the spatial derivatives. The same TVDRK method as applied to solve the level set equations discretizes time. The partial differential equation (4.2) is of Hamilton-Jacobi type, with Hamiltonian

$$H(\nabla \hat{d}, \hat{d}) = s(\hat{d}) \left(|\nabla \hat{d}| - 1 \right). \tag{4.5}$$

An upwind scheme requires definition of a monotone numerical Hamiltonian. One option, applied in this work, is Godunov's method. For this problem, Godunov's method results in the following discretization

$$\begin{aligned} \hat{H}(\hat{d}_{ij}) =& s_{ij}^{+} \left(\sqrt{\max((A^{+})^2, (B^{-})^2) + \max((C^{+})^2, (D^{-})^2)} - 1 \right) \\ &+ s_{ij}^{-} \left(\sqrt{\max((A^{-})^2, (B^{+})^2) + \max((C^{-})^2, (D^{+})^2)} - 1 \right) \end{aligned} \tag{4.6}$$

where

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

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

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

With first order time integration, the scheme is then

$$\hat{d}_{ij}^{n+1} = \hat{d}_{ij}^n - \Delta\tau \hat{H}(\hat{d}_{ij}^n). \tag{4.7}$$

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

4.2 Reflection boundary conditions

First express the level set equation (2.18) in operator form as

$$\begin{aligned} \frac{\partial \phi_l}{\partial t} &= \mathcal{L}\phi_l \\ \mathcal{L} &= -\mathbf{V}\cdot\nabla, \end{aligned} \tag{4.8}$$

with $\mathbf{V}$ as in (2.19). Then 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.

Fix (x_i, θ_k) and consider the operator $\tilde{\mathcal{L}}_z$ above. 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 $j = 0, \ldots, N-1$. The superscripts $(\cdot)^+$ and $(\cdot)^-$ indicate the limits from the right and left, respectively. This allows one 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

$$\hat{\phi}_{i,j+1/2,k} = \begin{cases} \tilde{\phi}^+_{i,j+1/2,k} & \text{if } \sin(\theta_k) < 0 \\ \tilde{\phi}^-_{i,j-1/2,k} & \text{otherwise .} \end{cases} \tag{4.9}$$

If $z_b(x_i) \equiv 2$, then the grid point $z_{N-1/2}$ coincides with the reflecting boundary so to implement (3.3), set $\tilde{\phi}^+_{i,N-1/2,k} = \tilde{\phi}^-_{i,N-1/2,N-k-1}$. The following discussion considers the situation where

$$z_b(x_i) = z_{j_b-1/2} + \gamma h, \tag{4.10}$$

with $0 < \gamma < 1$ for some index $j_b \in \{0, 1, \ldots, N-1\}$. The one-dimensional restriction is pictured in Figure 4.3.

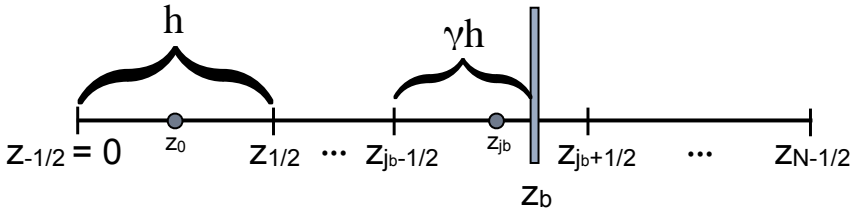


Figure 4.3: Respresentation of the grid in one dimension with boundary located at $z_b = z_{j_b-1/2} + \gamma h$

Two approaches are proposed. In the first, further described in Section 4.2.1, the nearest in-domain grid point approximates the location of the boundary. Section 4.2.2 describes a different approach which assumes that the sound speed is constant in a neighborhood of the boundary in an effort to derive a localized, higher order boundary condition. Each is described with respect to computing $\tilde{\mathcal{L}}_z$. The details are similar for computing $\tilde{\mathcal{L}}_x$.

4.2.1 Method 1: Boundary location approximation

This approach is straightforward and easy to implement. Given j_b for each x_i, apply (3.3) 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. To implement Method 1, apply the following steps in the pre-compute stage:

1. Solve the Hamilton-Jacobi equation for the signed distance function to the boundary as described in Section 4.1 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 $j = 0, \ldots, N-1$, and also the indices j_b associated with each $i = 0, \ldots, N-1$.

3. For each $\theta_b(x_i)$, loop over all angles θ_k, $k = 0, \ldots, N-1$, 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 identified reflected angle, compute and store the associated incident angle using (3.4).

4.2.2 Method 2: Local ray approximation

This approach combines the law of reflection with the fact that (4.8) 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 has a reasonable constant approximation in a neighborhood of $z_b(x_i)$, and that $z_b(x)$ is smooth and has a reasonable piecewise linear approximation. The time domain ray (characteristic) equations for the Eikonal equation are

$$\begin{aligned}
\dot{x}(t) &= c(x(t), z(t)) \cos\theta(t) \\
\dot{z}(t) &= c(x(t), z(t)) \sin\theta(t) \\
\dot{\theta}(t) &= \frac{\partial c}{\partial x} \sin\theta(t) - \frac{\partial c}{\partial z} \cos\theta(t).
\end{aligned} \tag{4.11}$$

The dot denotes differentiation with respect to time. For small t, expand $\theta(t) \approx \theta(0) \equiv \theta_0$, and likewise, using (4.11),

$$\begin{aligned}
x(t) &\approx x_0 + tc(x_0, z_0) \cos\theta_0 \\
z(t) &\approx z_0 + tc(x_0, z_0) \sin\theta_0.
\end{aligned} \tag{4.12}$$

The assumption is that the travel time along a ray for which $||(x, z) - (x, z_b(x))|| < h$ is small in some norm, so that the rays are approximately 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 the transport property of the equation for the level set functions and (3.3),

$$\phi(t, R, \theta_{refl}) = \phi(t - \tau, P, \theta_{refl}) = \phi(t - \tau, P, \theta_{inc}). \tag{4.13}$$

In the absence of a reflecting boundary, the ray would continue to travel in a straight line to a location $R' \in \Omega^+$, and by (4.13), the level set function would satisfy

$$\phi(t, R, \theta_{refl}) = \phi(t, R', \theta_{inc}). \tag{4.14}$$

That is, by propagating the level set function into Ω^+, an exact boundary condition is available when c is (locally) constant and $z_b(x)$ is (locally) linear. Typically R' will not lie in the grid so interpolation is needed. Thus, 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 (4.14).

To compute R', consider the geometry illustrated in Figure 4.4. 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 that the boundary is linear in the cell $(x_{i-1/2}, x_{i+1/2})$, $(R')^T = TR^T$, where

$$T = \begin{pmatrix} \sin^2\theta_B - \cos^2\theta_B & -2\cos\theta_B \sin\theta_B \\ -2\cos\theta_B \sin\theta_B & -\sin^2\theta_B + \cos^2\theta_B \end{pmatrix}. \tag{4.15}$$

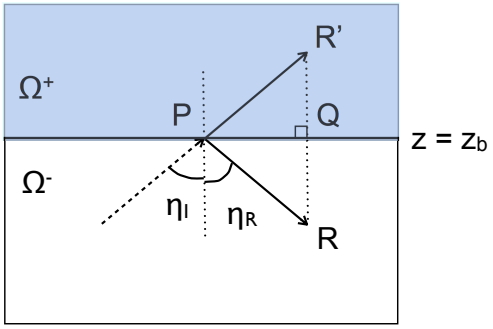


Figure 4.4: Law of reflection

4.2.3 Precomputing and data structures

Implementation of the method described in 4.2.2 requires the following steps in addition to those listed in Section 4.2.1:

5. Extend the domain into Ω^+ to allow for solutions surrounding the image location. The distance beyond the boundary of the image location has γh for an upper bound, so the stencil size of the numerical solver and interpolation used determines the number of grid points to insert beyond. A minimum of two 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. The examples in this Chapter take 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 (4.15).

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

The prototype implementation of the level set method described in Chapters 3 and 4 is written in the C language using the GNU Scientific Library [23]. In order to store all of the precomputed information, the implementation defines the C structures in Listings 4.1 and 4.2.

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, one may store a `gridcell` for each segment, tied to the relevant boundary (left or right).

Listing 4.1: Boundary `gridcell` data structure

```
typedef struct {

        size_t indx;  //Index of this cell
        int boundary_indx; //Boundary index
        double th_normal; //Outward normal to 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 (interpolation)
        gsl_matrix * R; //Reflection map

} gridcell;
```

4.3 Arrival time analysis

To help quantify the error imposed by the approximations in Sections 4.2.1 and 4.2.2, this section studies the proposed boundary conditions more closely. The first part considers the accuracy of the boundary conditions in terms of arrival time errors. The second part looks at the stability. It is also worth noting that the piecewise linear assumption on $z_b(x)$ (or equivalently, piecewise constant for $\theta_B(x)$) results in an additional $O(h^2)$ error.

Listing 4.2: Boundary `Slice` data 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;
```

4.3.1 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, it is more valid to investigate specific errors that occur. This section focuses on how approximation errors in the boundary condition affect the wavefront travel time. Travel time is more physically relevant than global error in ϕ since it provides the phase information for the Eikonal equation (2.11).

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 (4.10), 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

$$e_{t^*} = \frac{\gamma h}{c \sin\theta}. \tag{4.16}$$

To evaluate further, consider $\theta_B = \frac{\pi}{2}$. Then the incident angles satisfy $\theta_{inc} = -\theta_{refl}$. For angles close to normal incidence, $e_{t^*} \approx \frac{\gamma h}{c} = O(h)$ 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, one 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 $\lim_{l\to\infty} \gamma_l = 0$. Too see this, express 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 one chooses the h_l 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 effectively the same as modifying the grid to include the point z_b, and that modification could be different for each fixed x, violating grid uniformity. 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, one would expect any approximation to exhibit this degradation at reflected angles that are nearly parallel to $z_b(x_i)$.

Method 2

The boundary condition posed in section 4.2.2 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 4.2.1 yields $e_{t^*} = 0$. Instead, consider the error made in the approximation $\theta(t) \approx \theta_0$, or equivalently, $c =$ constant. The claim is that τ, the travel time along the ray, is small. To investigate the validity of that assumption, solve for τ using the ray equations (4.12) combined with the locally linear assumption

$$z_b(x) \approx z_b(x_i) + \cot\left(\theta_B(x_i)\right)\left(x - x_i\right), \tag{4.17}$$

taking γ as above, to see that

$$\tau_1 = \frac{\gamma h}{c\left(\cot\theta_B - \sin\theta_{refl}\right)}, \tag{4.18}$$

which for $\theta_B = \frac{\pi}{2}$ is exactly (4.16) in magnitude. Taking $\theta_B = \frac{\pi}{2}$, the remainder term for the small time expansion will be proportional to

$$|R_0(\tau_1)| = \frac{\gamma h}{|\sin(\theta_0)|} \left| \frac{\partial c}{\partial x} \sin\theta(\xi) - \frac{\partial c}{\partial z} \cos\theta(\xi) \right| \tag{4.19}$$

for some $\xi \in (0, \tau_1)$. Again, this term is $O(h)$ 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 it is expected that the $\frac{\partial c}{\partial z} \cos\theta$ term in (4.19) will dominate.

Consider next 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. 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 $\tau = \int_0^s \frac{1}{c(z(u))} du$, where s is the arc length of the ray. For the given c, the travel time is [32] (p. 210)

$$\tau_2 = \left| \frac{1}{g} \log \left[\frac{1 + \sqrt{1 - a^2 c^2}}{ac} \right]_{c(z_b)}^{c(R)} \right|. \tag{4.20}$$

To find the constant a, apply Snell's law to the ray at its turning point, yielding

$$a = \frac{\cos\theta_{refl}}{c(z_b)}. \tag{4.21}$$

Substituting equation (4.21) into (4.20) and using the fact that $c(z_b) = c(R) + g\gamma h$,

$$\tau_2 = \left| \frac{1}{g} \log \left(\frac{c(R) + g\gamma h + \sqrt{c(R)^2 \sin^2 \theta_{refl} + g\gamma h \left(2c(R) + g\gamma h\right)}}{c(R)\left(1 + |\sin \theta_{refl}|\right)} \right) \right|. \tag{4.22}$$

For reflections roughly normal to the boundary, i.e., $\theta_{refl} \approx -\frac{\pi}{2}$, (4.22) simplifies to

$$\begin{aligned} \tau_2 &= \left| \frac{1}{g} \log \left(1 + \frac{g\gamma h}{c(R)} \right) \right| \\ &\approx \left| \frac{\gamma h}{c(R)} - \frac{g\gamma^2 h^2}{c(R)^2} \right| \\ &= \tau_1 + C(\gamma h)^2. \end{aligned} \tag{4.23}$$

Thus, $e_{t^*} = O(h^2)$ as $h \to 0$ for a linear sound speed profile with normal incidence. Recall that section 4.3.1 derived e_{t^*} 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 computing the angles. This result suggests this method offers improved performance over the location approximation method even for the variable coefficient case.

As in Method 1, one would expect to observe degraded performance when the angle of ray incidence at the boundary is near parallel. Given a fixed mesh spacing h, the worst case is $\theta_{refl} = -\pi\frac{h}{2}$, for which $\tau_1 \approx \frac{2\gamma}{\pi c(R)} - O(h^2)$. Expanding (4.22)

about small γh leads to

$$\tau_2 \approx \left| \frac{2\gamma}{\pi c(R)} \sqrt{\frac{c(R)\pi}{2\gamma g}} \sqrt{h} - c(R)\gamma h \right|$$
$$= \tau_1 \sqrt{\frac{c(R)\pi}{2\gamma g}} \sqrt{h} + O(\gamma h). \tag{4.24}$$

Then e_{t^*} is proportional to $\tau_2 - \tau_1 \approx \sqrt{\frac{2\gamma}{\pi c(R)}} \left(\sqrt{\frac{h}{g}} - \sqrt{\frac{2\gamma}{\pi c(R)}} \right)$. 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 4.3.1 when a linear c is approximated by constant c.

4.3.2 Stability

To simplify the analysis, consider the problem for x in one spatial dimension. Then there are only two propagation directions given by $\theta \in \{-\pi, 0\}$. The level set equations satisfy the system

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

where

$$\mathbf{U} = \begin{pmatrix} u \\ v \end{pmatrix} (t, x) \text{ and}$$

$$\mathbf{A}(x) = \begin{pmatrix} c(x) & 0 \\ 0 & -c(x) \end{pmatrix}.$$

Initial and boundary conditions must be supplied to pose the problem completely. In this case,

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

where x^* is a fixed, pre-determined point in $(0, x_B)$. Assume that $c(x) \geq \epsilon > 0$, so that $u(t, x)$ represents a right-traveling wave and $v(t, x)$ is a left-traveling wave. Thus, boundary conditions are needed 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 reflection boundary condition couples the two equations.

To further simplify the analysis, consider the constant coefficient problem in the half plane $x \in (0, \infty)$. Using the standard notation, $k = \Delta t$ (not to be confused with

index k used previously), $\lambda = \frac{k}{h}$, and E^j is the shift operator: $E^j(U(x)) = U(x+jh)$. Apply the first order scheme with reflection at $x = 0$:

$$\begin{aligned}
\mathbf{U}_j^{n+1} &= \left[\begin{pmatrix} c\lambda & 0 \\ 0 & 0 \end{pmatrix} E^{-1} + (1 - c\lambda)\mathbf{I} + \begin{pmatrix} 0 & 0 \\ 0 & c\lambda \end{pmatrix} E^1 \right] \mathbf{U}_j^n,\ j = 1, 2, \ldots \qquad (4.25) \\
&\equiv Q\mathbf{U}_j^n \\
&\equiv \left(\mathbf{A}_{-1}E^{-1} + \mathbf{A}_0 E^0 + \mathbf{A}_1 E^1 \right) \mathbf{U}_j^n \\
\mathbf{U}_0^{n+1} &= \left[\begin{pmatrix} 0 & 1 - c\lambda \\ 0 & 1 - c\lambda \end{pmatrix} + \begin{pmatrix} 0 & c\lambda \\ 0 & c\lambda \end{pmatrix} E^1 \right] \mathbf{U}_0^n. \qquad (4.26)
\end{aligned}$$

Consider first the effect of only the boundary condition on the left, using the techniques presented in [29] and [66].

The matrices in (4.25) must satisfy a few conditions for the analysis to apply:

1. The approximation (4.25) is stable for the Cauchy problem. Then necessarily the von Neumann condition is satisfied. That is, taking $\zeta = e^{i\omega k}$, and operators $\mathcal{A}_l(\zeta) = -\frac{1}{\zeta}\mathbf{A}_l + \mathbf{I}$, then the eigenvalue problem

$$\sum_{l=-1}^{1} \mathcal{A}_l(\zeta) e^{il\xi} \mathbf{y} = 0 \qquad (4.27)$$

has no solution ζ with $|\zeta| > 1$. Here Q is the well-known first order upwind

scheme, and this condition holds provided that $c\lambda < 1$. The eigenvalues are

$$\zeta = 1 - c\lambda(1 - \cos\xi) \pm ic\lambda|\sin\xi|. \tag{4.28}$$

2. Either the eigenvalues satisfy the condition that any with multiplicity > 1 for some $\xi = \xi_0$ with $|\zeta_l| = 1$, $\frac{\partial\zeta_l}{\partial\xi}(\xi_0) \neq 0$, or the the matrices $\mathbf{A}_l$ are diagonalizable. For this scheme, the first is not true (consider $\xi_0 = 2m\pi$, m an integer), but the second condition holds.

3. Either $|\zeta_l| < 1$ for each l and $0 < |\xi| \leq \pi$ or $|\zeta_l| = 1$ for all l, ξ. This is not difficult to see from (4.28).

4. $\mathcal{A}_{-1}(\zeta)$ and $\mathcal{A}_1(\zeta)$ are nonsingular for $|\zeta| \geq 1$. This (not necessary, but convenient - c.f. [29]) also holds since $\mathcal{A}_l$ is singular only when $\zeta = c\lambda < 1$ for $l = -1, 1$.

Under the above assumptions, Q has two linearly independent right-going and two linearly independent left-going solutions when $|\zeta| \geq 1$. Gustafsson, Kreiss, and Sundström established in [29] 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. Now consider solutions in the form

$$U_j^n = \hat{U}\kappa^j\zeta^n \ , \ \kappa, \zeta \in \mathbb{C} - \{0\}\,, \tag{4.29}$$

with $\kappa = e^{i\xi h}$ and $\zeta = e^{i\omega k}$. The dispersion relation relates ω to ξ for waves. Analogously, following [66], consider the numerical dispersion relation for ζ in terms of κ. To obtain an expression, substitute (4.29) into Q and solve. 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

$$\zeta = c\lambda \frac{1 - \kappa}{\kappa} + 1, \; c\lambda(\kappa - 1) + 1. \tag{4.30}$$

The first solution for ζ is the right-going solution component $u(t, x)$ and the second is for the left-going wave, $v(t, x)$. In the subsequent, μ represents left-going solutions and let κ represent only the right-going solutions. The second dispersion relation in (4.30) becomes

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

Next seek solutions to the scheme (4.25) that also satisfy (4.26) and have the form

$$U_j^n = \zeta^n \sum_{m=1}^{2} \alpha_m \hat{P}_m \kappa_m^j + \zeta^n \sum_{m=1}^{2} \beta_m \hat{R}_m \mu_m^j. \tag{4.31}$$

Continuing to follow Trefethen [66], apply the boundary condition (4.26) to (4.31)

and collect terms in α and β to arrive at an expression of the form

$$E(\zeta)\begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} = D(\zeta)\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix}. \tag{4.32}$$

The concern is the existence of 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. The matrices in (4.32) are

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

and

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

so $E(\zeta)$ is singular when

$$\zeta\det(\hat{\mathbf{P}})\,(\zeta + c\lambda(1-\kappa) - 1) = 0. \tag{4.33}$$

Here, $\hat{\mathbf{P}} = \left[\hat{P}_1\hat{P}_2\right]$ 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 system has rank one 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 [66], define the reflection coefficient matrix $A_R(\zeta) = E^{-1}(\zeta)D(\zeta)$ when E is nonsingular. It is straightforward to compute

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

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.

It follows in a similar fashion 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 [66], the model is stable in the sense defined in [29] because the interfaces at $x = 0$ and at $x = 1$ are both individually stable.

4.3.3 Effect of perturbations

The proposed boundary conditions are approximate, thus their application imposes an error at each reflecting boundary at each time step. This section studies how the error produced at the boundary propagates in the one dimensional case presented above. Consider (4.25) with initial condition $\mathbf{U}_j^0 = 0$ and perturbed boundary condition $u_0^n = v_0^n + \epsilon$, where ϵ is the $O(h)$ or $O(h^2)$ approximation error for the boundary condition. Also take the reflection boundary condition at $x = 1$ for v to be $v_N^n = u_N^n$. Consider the u_j^n and v_j^n components separately in the scheme (4.25) to construct the discrete error equation as follows.

The objective is to solve the following difference equation:

$$u_j^n = (1 - c\lambda)u_j^{n-1} + c\lambda u_{j-1}^{n-1} \tag{4.34}$$

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

property further implies that

$$u_j^n = 0 \text{ for } j \geq n. \tag{4.35}$$

Now apply generating functions to solve the difference equation. First, let

$$u_n(x) = \sum_{j=0}^{\infty} u_j^n x^j = \sum_{j=0}^{n-1} u_j^n x^j, \tag{4.36}$$

where the second equality is due to (4.35). Now multiply (4.34) by x^j and sum over j to find

$$\begin{aligned} u_{n+1}(x) &= (1 - c\lambda)u_n(x) + \sum_{k=-1}^{n-2} c\lambda u_k^n x^{k+1} \\ &= (1 - c\lambda)u_n(x) + c\lambda x u_n(x) + c\lambda u_{-1}^n \\ &= (1 - c\lambda + c\lambda x)u_n(x). \end{aligned} \tag{4.37}$$

Now let $u(x, y) = \sum_{n=0}^{\infty} u_n(x)\frac{y^n}{n!}$, multiply (4.37) 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,

$$\frac{\partial u(x, y)}{\partial y} = (1 - c\lambda + c\lambda x)u(x, y), \tag{4.38}$$

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

$$u(x,1) = C(x)e^{1-c\lambda+c\lambda x} = \sum_{n=0}^{\infty} u_n(x)\frac{1}{n!}$$
$$\implies u_n(x) = C(x)\left(1-c\lambda+c\lambda x\right)^n.$$

Taking $n = 1$,

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

Thus,

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

and

$$u_n(x) = \epsilon\left(1-c\lambda+c\lambda x\right)^{n-1}. \tag{4.39}$$

Comparing the above expression to (4.36) and applying the binomial theorem results in

$$u_j^n = \begin{cases} \epsilon\binom{n-1}{j}\left(1-c\lambda\right)^{n-j-1}\left(c\lambda\right)^j & , j = 0,1,\ldots,n-1 \\ 0 & , \text{otherwise} \end{cases}. \tag{4.40}$$

Next, solve the difference equation produced by (4.25) for v_j^n subject to a perturbation at the boundary using the same techniques. The difference equation is

$$v_{N-j}^n = (1-c\lambda)v_{N-j}^{n-1} + c\lambda v_{N-j+1}^{n-1}, \tag{4.41}$$

subject to initial condition $v_{N-j}^0 = 0$, and boundary condition $v_N^n = \epsilon$, $n = 1,2,\ldots$

It is a consequence of the transport equation that $v^n_{N-j} = 0$ for $j \geq n$. Using this property in addition to the boundary condition, one derives the useful identity

$$v^n_{N-n+1} = \epsilon \left(c\lambda\right)^{n-1}. \tag{4.42}$$

Applying the expansion

$$v_n(x) = \sum_{j=0}^{n-1} v^n_{N-j} x^{-j}, \tag{4.43}$$

leads to

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

Using (4.42) and the fact 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

$$\frac{\partial v(x,y)}{\partial y} = \left(1 - c\lambda + \frac{c\lambda}{x}\right) v(x,y) - \epsilon \exp\left(\frac{c\lambda}{x}y\right). \tag{4.44}$$

Equation (4.44) 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 (4.44) one

finds that $D(x) = \frac{\epsilon}{1-c\lambda}$, so the generating function for the coefficients v_{N-j}^n 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 $n = 1$,

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

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 $v_n(x)$,

$$v_n(x) = \frac{\epsilon}{1-c\lambda}\left[\left(1 - c\lambda - \frac{c\lambda}{x}\right)\left(1 - c\lambda + \frac{c\lambda}{x}\right)^{n-1} + \left(\frac{c\lambda}{x}\right)^n\right]. \tag{4.45}$$

Finally, in order to compare (4.45) to the coefficient terms in (4.43), expand using the binomial theorem:

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

That is,

$$v_{N-j}^{n} = \begin{cases} \epsilon\,(1-c\lambda)^{n-1}\left(\frac{n-2j}{n}\right)\binom{n}{j}\left(\frac{c\lambda}{1-c\lambda}\right)^{j} & , j = 0,1,\ldots,n-1 \\ 0 & , \text{otherwise} \end{cases}. \tag{4.46}$$

One may now apply (4.40) and (4.46) 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 $u_N^{N+1} = (c\lambda)^N\,\epsilon$. The reflection boundary condition at $x = 1$ is $v_N^{N+1} = u_N^{N+1} = (c\lambda)^N\epsilon$. Assuming the boundary condition on

the right is exact, then the perturbation returns to $x = 0$ as

$$v_0^{2N+1} = \epsilon\,(c\lambda)^{2N} \binom{2N}{N} \left(1 - \frac{N}{N+1}\right) (1 - c\lambda)^N . \tag{4.47}$$

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. Stirling's approximation,

$$N! \approx \sqrt{2\pi N} \left(\frac{N}{e}\right)^N , \tag{4.48}$$

allows one 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 (4.48) in (4.47),

$$v_0^{2N+1} \approx \epsilon \left[4\,(c\lambda)^2\,(1 - c\lambda)\right]^N \frac{2}{(N+1)\sqrt{\pi N}}. \tag{4.49}$$

The limiting behavior of (4.49) 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, $g(c\lambda) \leq \frac{4}{27} < \frac{1}{4}$. 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, the error multiplier with $c\lambda = \frac{2}{3}$ (this value of $c\lambda$ maximizes $g(c\lambda)$ over $(0, 1)$) was computed for $N = 10$, $N = 20$, and $N = 40$. The added errors were, respectively, $(8.554 \times 10^{-5}) \times \epsilon$, $(1.7026 \times 10^{-7}) \times \epsilon$, and $(1.76 \times 10^{-12}) \times \epsilon$, where the result for $N = 40$ is based on an approximation for large N.

4.4 Results and validation of the reflection boundary conditions

4.4.1 Validation of arrival time analysis

Figures 4.5 and 4.6 show computational results for the arrival time errors. In both cases, the scenario is for an upward sloping bottom with $\theta_B = \frac{\pi}{3}$. Figure 4.5 is for a constant wave speed $c = c_0 = 1.5$ km/sec, and for Figure 4.6, the wave speed is linear, $c(z) = 0.5z + c_0$. 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 Figure 4.5, it appears that the convergence rate for Method 2 is greater than for Method 1. Although Method 2 provides an exact model for this scenario, observed convergence is no faster than $O(h^2)$ due to coupling with the errors in the PDE solver and wavefront extraction routine. It is apparent that the convergence rate for Method 1 is roughly $O(h)$ and for Method 2, about $O(h^2)$. The error levels off for Method 2 at about 10^{-5}, this is likely due to time discretization error.

Figure 4.6 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 $O(h)$ for Method 1. It is less clear for Method 2, but the results appear to be converging faster than $O(h)$ 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 effective dimensionality of the problem.

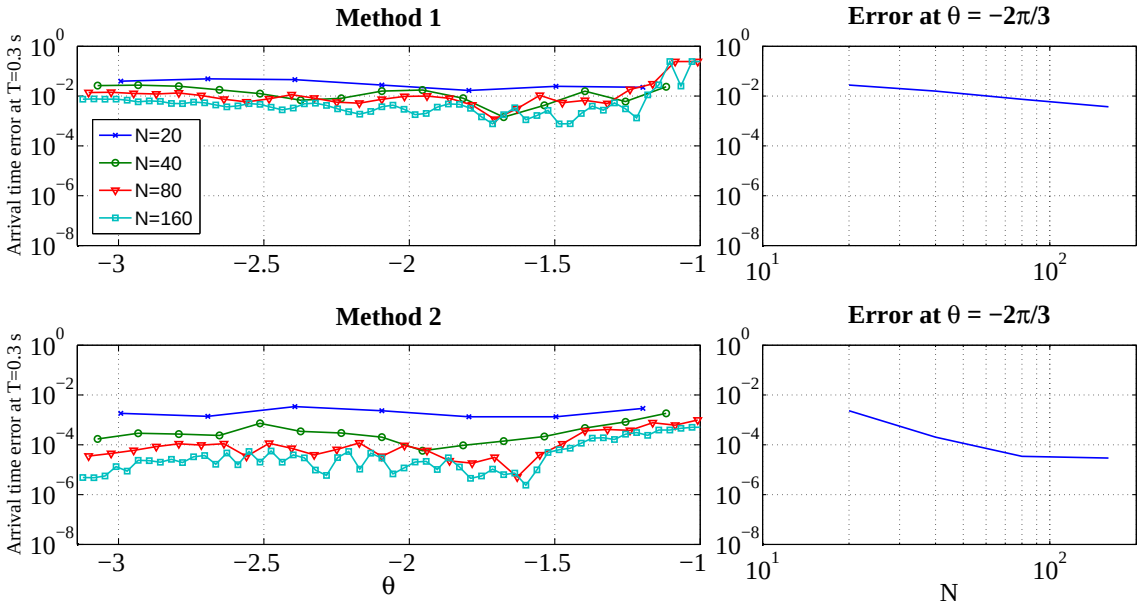


Figure 4.5: Arrival time errors with $c = 1.5$ km/sec

4.4.2 Wavefront examples

To provide some visual wavefront results, this section presents 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

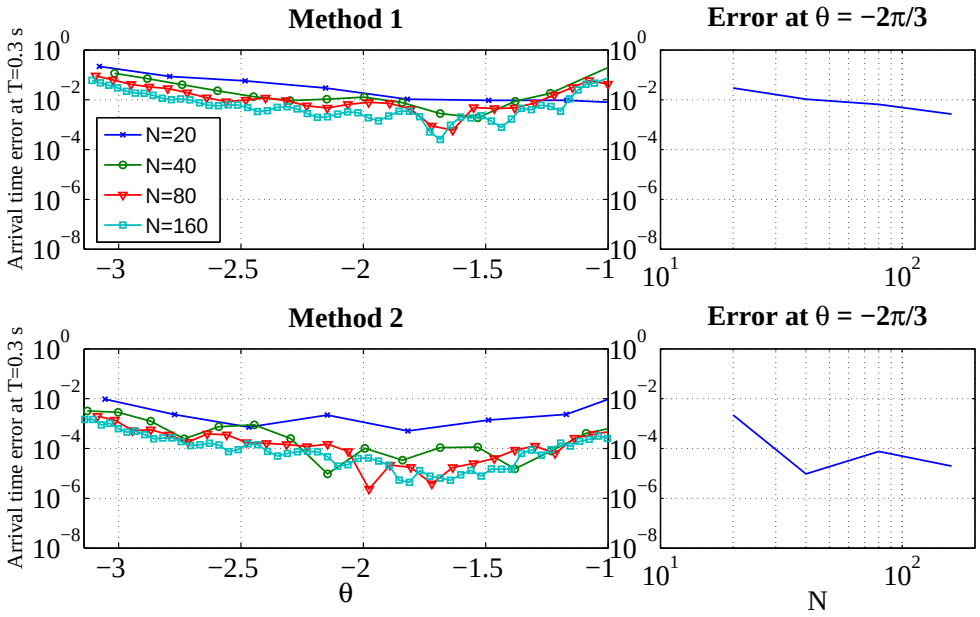


Figure 4.6: Arrival time errors with $c = 0.5z + 1.5$ km/sec

the Mayavi program [55]. The code is provided in Appendix C.

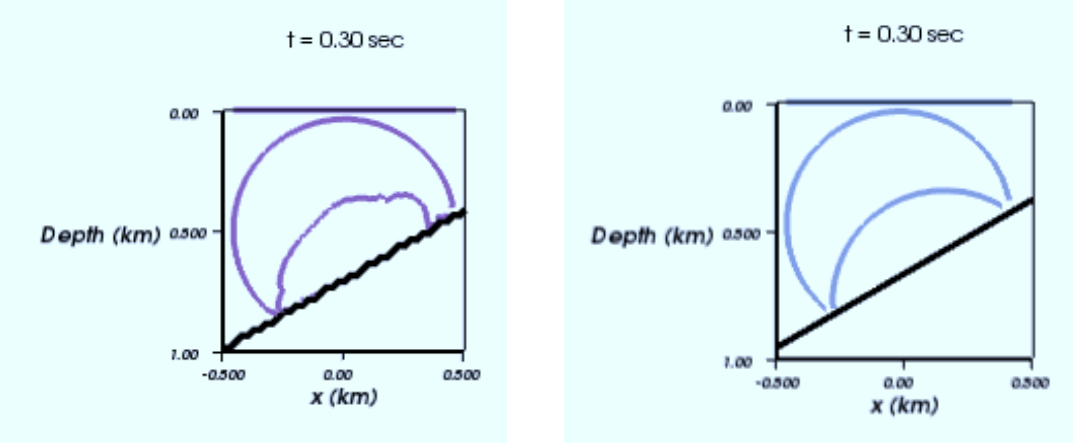
Comparison between Method 1 and Method 2

Figure 4.7 shows the results from a $40 \times 40 \times 40$ grid over

$$\Omega = \{(x, z) \in [-0.5, 0.5] \times [0, 1]\}$$

and a sloping bottom with $\theta_B = \frac{\pi}{3}$ radians. A constant sound speed $c = 1.5$ km/sec is used here. At this resolution, Method 2 clearly produces superior results, 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 method does apply the correct normals. When the grid size is increased to $160 \times 160 \times 160$, the wavefront produced using the Method 1 boundary condition is converging, though very slowly at angles close to parallel with the boundary.

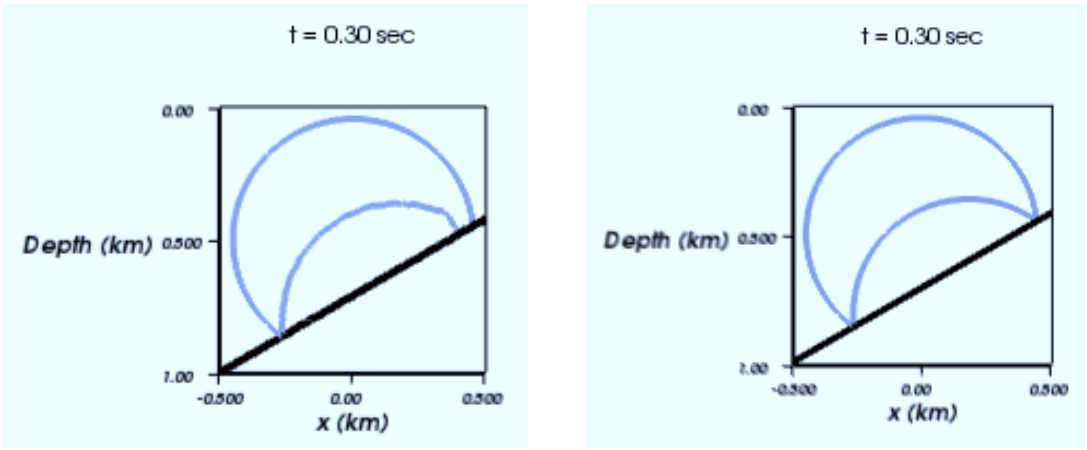


(a) Method 1 **(b)** Method 2

Figure 4.7: Wavefront comparison with $N = 40$ at 0.3 seconds, $c = 1.5$ km/s, $\theta_B = \frac{\pi}{3}$

Examples with non-linear domain geometry

Figures 4.9 and 4.10 are examples of the types of geometry for which Method 2 provides nice results. Again, the examples apply a constant sound speed $c = 1.5$ km/sec for ease of interpretation. The domain for each is $\Omega = \{(x, z) \in [-1, 1] \times [0, 1]\}$. Each subfigure consists of a snapshot of the wavefront at a few times, $t = 0.2, 0.4, 0.6, 0.8$ seconds. In Figure 4.9, a modified (inverted) Gaussian function represents an undersea canyon, and in 4.10, a shifted Gaussian function represents a seamount structure. Note that the Gaussian function is not a limitation, just a convenient choice. Due to



(a) Method 1 **(b)** Method 2

Figure 4.8: Wavefront comparison with $N = 160$ at 0.3 seconds, $c = 1.5$ km/s, $\theta_B = \frac{\pi}{3}$

the multiple reflections produced by the canyon example, Method 1 does not produce reasonable results.

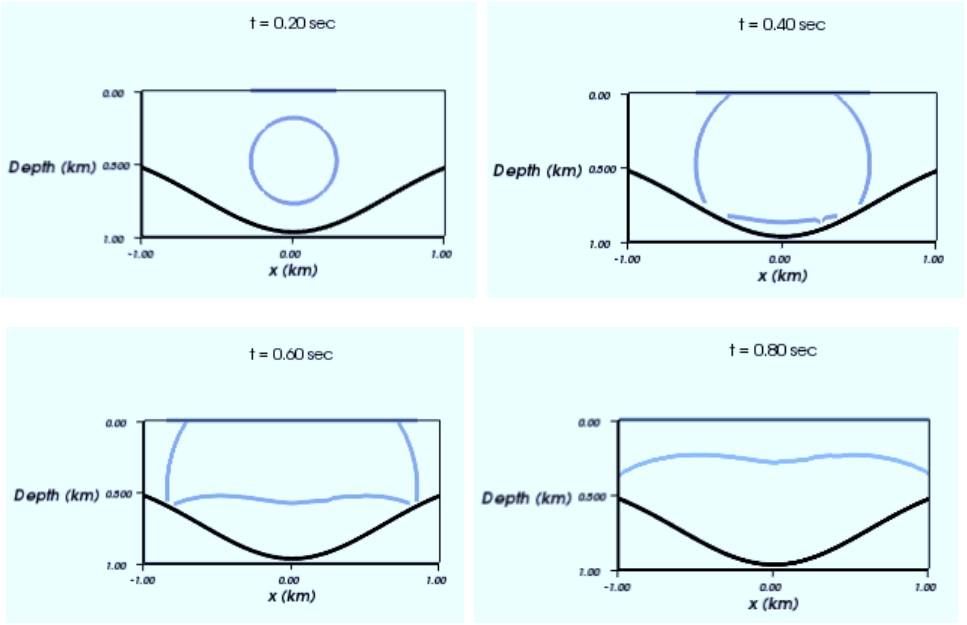


Figure 4.9: Wavefront snapshots with $z_b(x) = 1 - \frac{1}{\sigma\sqrt{2\pi}}\left(1 - \exp[-\frac{x^2}{2\sigma^2}]\right)$

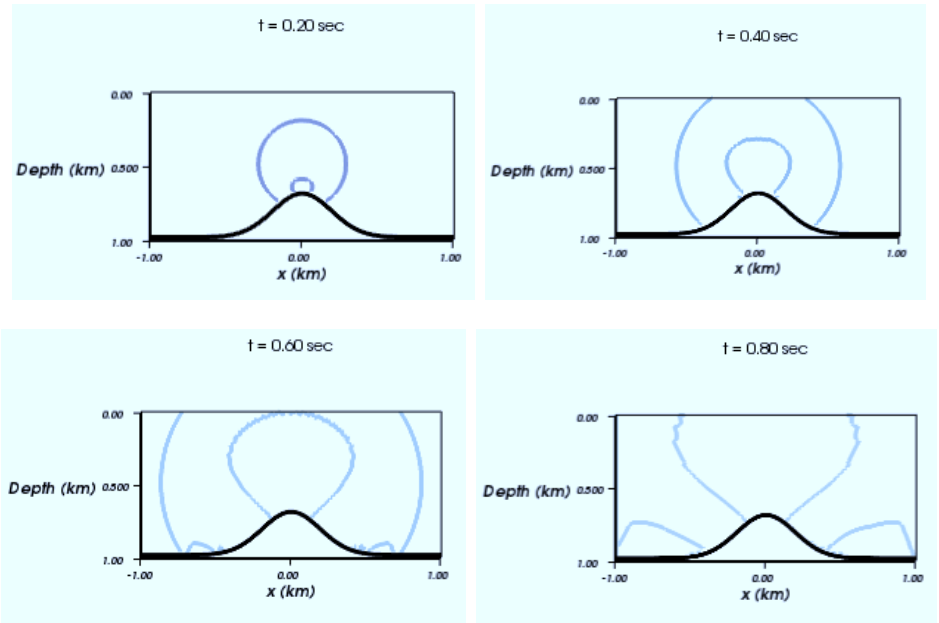


Figure 4.10: Wavefront snapshots with $z_b(x) = 1 - 0.3\exp[-\frac{x^2}{2\sigma^2}]$

Chapter Five

Incorporating uncertainty

5.1 Wavefront propagation in the presence of uncertainty

The level set method described in the preceding chapters requires input in the form of a Sound Speed Profile (SSP) specifying $c(x, z)$, bottom depth as a function of x specifying $z_b(x)$, and a source location to initialize the level set functions. Focusing on the SSP, the data are often computed using empirical models dependent on measurements of depth, salinity and temperature. Variation in x and y coordinates of physical space is less significant and hence, frequently neglected. An example of a simplified model is [32]

$$c(z) \approx 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1.34 - 0.01T)(S - 35) + 0.016z, \quad (5.1)$$

where temperature T is measured in degrees centigrade, salinity S in parts per thousand, and depth z in meters. These quantities are typically available as measured data, subject to errors in the equipment used to take the measurements, as well as the unknown variability in the environment. For instance, the SSP varies not only in space, but with time of day. This is especially true in shallow water where warming and cooling patterns in the atmosphere have a predominant effect. In deep water, these factors affect the upper ocean, forming a *surface duct* and below that, a *thermocline* layer where $c(z)$ decreases with z as the temperature decreases. Below the thermocline, temperatures are fairly constant and the SSP depends primarily on

increasing pressure.

There are also effects of un-modeled (e.g., non-linear) processes on propagation to consider, like internal waves and currents, in which there has been interest of late due to realizations, at least for long range propagation, that these un-modeled effects can result in significant deviations of ray paths [15], [13]. It is also common to ignore SSP variability in x (or range in the azimuthal symmetry model), and the effect of many small scatterers on propagation, e.g., bubbles. This chapter focuses on the application of uncertainty quantification techniques to the level set method as a way to obtain wavefront solutions in a probabilistic sense, in the presence of uncertainty in the SSP. As discussed in Chapter 1, this capability is valuable since the eigenray problem is highly sensitive to small perturbations in initial conditions. The examples in this chapter are simplified, serving to illustrate that these techniques are applicable to the level set method. It is also worth noting that these techniques are not limited to modeling wave speed uncertainty, the same methods apply to models involving multiple random parameters. Other sources of uncertainty that one might wish to model include, but are not limited to, source and receiver positions and rough surface scattering. The result would be an increase in the dimension of the random space.

5.1.1 Related work in uncertainty quantification

The approach taken in this thesis is based on the work of Xiu and Hesthaven [71], which presents a framework for the stochastic collocation approach to solving partial differential equations involving non-white processes. Classical stochastic calculus approaches assume idealized, white (uncorrelated) processes, whereas most physical processes do have some manner of correlation that cannot be neglected. Monte Carlo techniques are effective and easy to implement, but suffer from slow convergence. Stochastic collocation methods using deterministic sampling approaches converge very fast for certain processes, though the performance depends on the dimension. Chauviere, Hesthaven and Lurati later applied the method to computational electromagnetics in [6]. Nobile, Tempone and Webster undertook a study of stochastic collocation on sparse grids for elliptic and diffusion equations [44].

5.1.2 Outline of this section

Chapters 2, 3, and 4 describe a method for computing solutions to the deterministic transport equation (3.2),

$$f_t + c(x,z)\cos(\theta)f_x + c(x,z)\sin(\theta)f_z + (c_x\sin(\theta) - c_z\cos(\theta))f_\theta = 0$$

for the two level set function components $\phi_1(t, x, z, \theta)$ and $\phi_2(t, x, z, \theta)$. This chapter addresses the case of a wave speed given by $c(x, z; \vec{\chi})$, with $(x, z) \in \Omega \subset \mathbb{R}^2$, and a random vector $\vec{\chi}$ taking values in a probability space $(\mathcal{W}, \mathcal{F}, \mathcal{P})$. In the usual notation, $\mathcal{W}$ is the sample space[1], $\mathcal{F}$ is the σ-algebra of events in $\mathcal{W}$, and $\mathcal{P}$ is the probability measure on the space. Assume that $c(x, z; \vec{\chi})$ has the form

$$c(x, z; \vec{\chi}) = c_0(x, z) + \epsilon c'(x, z; \vec{\chi}), \tag{5.2}$$

with $c'(x, z; \vec{\chi})$ a zero-mean process.

Section 5.2 provides background on the tools applied in the proposed model. Section 5.3 derives the relevant expansion for a single parameter expansion where $c'(x, z; \chi)$ is a uniform random variable and presents some results. And Section 5.4 allows $c'(x, z; \vec{\chi})$ to be a zero-mean Gaussian random field on the physical space. The Gaussian assumption is not a limitation, but offers mathematical convenience.

[1]Traditional notation denotes the sample space as Ω, but Ω exists in a different context in Chapter 4. Using $\mathcal{W}$ instead should alleviate potential confusion.

5.2 Generalized Polynomial Chaos

5.2.1 Approximation by Orthogonal Polynomials

The techniques applied in this chapter are based on expansions in orthogonal polynomials. To establish notation, let $Q_n(y)$ represent a degree n polynomial,

$$Q_n(y) = a_n y^n + a_{n-1} y^{n-1} + \cdots + a_1 y + a_0, \tag{5.3}$$

with $a_n \neq 0$. The collection

$$\{Q_n(y), n \in \mathcal{N}\}, \tag{5.4}$$

where $\mathcal{N}$ is some index subset of $\{0, 1, \dots\}$, is an orthogonal system of polynomials with respect to (real) measure μ if

$$\int_{\{\mu>0\}} Q_n(y) Q_m(y) d\mu(y) = \gamma_n \delta_{mn}, \ \forall m, n \in \mathcal{N}. \tag{5.5}$$

The γ_n are normalization constants, and δ_{mn} is the Kronecker delta function,

$$\delta_{mn} = \begin{cases} 1, & m = n, \\ 0, & m \neq n \end{cases}.$$

The system (5.4) is *orthonormal* if $\gamma_n = 1, n \in \mathcal{N}$. One can always normalize an orthogonal system by taking $\hat{Q}_n(y) = \frac{1}{\sqrt{\gamma_n}} Q_n(y)$. An important property is that all orthogonal, real polynomial systems satisfy a three-term recurrence relation,

$$Q_{n+1}(y) = (a_n y + b_n) Q_n(y) - c_n Q_{n-1}(y), \ n \geq 0, \tag{5.6}$$

along with the definitions $Q_{-1}(y) = 0$ and $Q_0(y) = 1$ [70].

The objective is to approximate a function f on some subinterval I of $\mathbb{R}$ as an expansion in orthogonal polynomials of an appropriate type. Take f to be in the weighted space $L^2_w(I)$ defined by

$$L^2_w(I) = \left\{ f : I \to \mathbb{R} \middle| \int_I f^2(y) w(y) dy < \infty \right\}. \tag{5.7}$$

This space has an inner product for all $f, g \in L^2_w(I)$

$$(f, g)_w = \int_I f(y) g(y) w(y) dy, \tag{5.8}$$

which generates the norm

$$||f||_w = \sqrt{\int_I f^2(y) w(y) dy}. \tag{5.9}$$

Let $\mathbb{P}_{N_p} \subset L^2_w$ be the subspace of real-valued polynomials of degree at most N_p and seek a polynomial in $\mathbb{P}_{N_p}$ that approximates $f \in L^2_w$. Choose $\{\psi_n(y)\}_{n=0}^{N_p}$,

orthogonal polynomials with respect to $w(y)$ to form a basis for $\mathbb{P}_{N_p}$. To simplify notation, assume the ψ_n are normalized so that $\gamma_n = 1$. Then the *projection* of f onto $\mathbb{P}_{N_p}$ is

$$P_{N_p} f(y) = \sum_{n=0}^{N_p} \hat{f}_n \psi_n(y), \tag{5.10}$$

$$\hat{f}_n = (f, \psi_n)_w. \tag{5.11}$$

Equation (5.10) along with equation (5.11) provides an approximation to f that is optimal in the sense that (5.11) defines the coefficients that minimize $||f - p||_{L^2_w(I)}$ over $p \in \mathbb{P}_{N_p}$. Convergence as $N_p \to \infty$ occurs for all $f \in L^2_w(I)$, at a rate that increases with regularity of f [22].

This work concerns itself with two families of polynomials: the Legendre family on $[-1, 1]$ and the Hermite family on $\mathbb{R}$. A few key facts about these polynomials are presented below, c.f. [70] and references therein for further details and examples. Both families have an associated weight function, $w(y)$ such that $d\mu(y) = w(y)dy$.

Legendre Polynomials

The Legendre polynomials $L_n(y)$ are defined on the interval $[-1, 1]$ and are orthogonal with respect to weight function $w(y) = 1$. The $L_n(y)$ solve the differential

equation ([70], p. 108)

$$(1-y^2)f''(y) - 2yf'(y) + n(n+1)f(y) = 0. \tag{5.12}$$

The recurrence relation for the Legendre polynomials is

$$L_{n+1}(y) = \frac{2n+1}{n+1} y L_n(y) - \frac{n}{n+1} L_{n-1}(y), \tag{5.13}$$

and the normalization constants are

$$\gamma_n = \frac{2}{2n+1}. \tag{5.14}$$

Hermite Polynomials

The Hermite polynomials $H_n(y)$ are defined on the real line and are orthogonal with respect to weight function $w(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$. This represents a scaled version of the classical Hermite polynomials; out of convenience in relating them to Gaussian probabilities. The $H_n(y)$ solve the differential equation ([70], p. 106)

$$f''(y) - yf'(y) + nf(y) = 0. \tag{5.15}$$

The recurrence relation for the Hermite polynomials is

$$H_{n+1}(y) = yH_n(y) - nH_{n-1}(y), \tag{5.16}$$

and the normalization constants are

$$\gamma_n = n!. \tag{5.17}$$

5.2.2 Polynomial Chaos

The generalized Polynomial Chaos (gPC) expansion is an expansion in terms of an orthogonal polynomial basis on the probability space $(\mathcal{W}, \mathcal{F}, \mathcal{P})$ as follows. Let χ be a random variable, $\chi : \mathcal{W} \to \mathbb{R}$ with distribution function $F(y) = P(\chi \leq y)$ and finite moments $\mathbb{E}\left[|\chi|^{2m}\right] = \int |y|^{2m} dF(y) < \infty, m = 0, \ldots, N_p$. The orthogonality condition for the gPC basis functions is

$$\mathbb{E}\left[\psi_n(\chi)\psi_m(\chi)\right] = \gamma_n \delta_{nm} \tag{5.18}$$

Analogous to (5.5), δ_{nm} is the Kronecker delta, and $\gamma_n = \mathbb{E}[\psi_n^2(\chi)]$. Thus, a good choice of basis functions are those with measure $d\mu(y)$ corresponding to $dF(y)$. That is, the weight function $w(y)$ is related to the probability density function (PDF) $\rho(y)$ defined as $dF(y) = \rho(y)dy$.

Expansion in a uniform random variable

Let $\chi \sim U[-1,1]$ be a uniformly distributed random variable on the interval $[-1,1]$, with PDF

$$\rho(y) = \begin{cases} \frac{1}{2}, & -1 \leq y \leq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.19}$$

One can always transform χ to a $U[a,b]$ random variable by $\tilde{\chi} = \left(\frac{b-a}{2}\right)\chi + \frac{b+a}{2}$. The requirement (5.18) then implies that an expansion in Legendre polynomials is appropriate, with normalization constant $\tilde{\gamma}_n = \frac{\gamma_n}{2}$ where γ_n is as in (5.14) to account for the PDF.

Expansion in a Gaussian random variable

Let $\chi \sim N(0,1)$ be a standard Gaussian random variable (i.e., zero-mean and variance 1), defined on $\mathbb{R}$, with PDF

$$\rho(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}. \tag{5.20}$$

One can always transform χ to a $N(\mu, \sigma^2)$ random variable by $\tilde{\chi} = \sigma\chi + \mu$. The requirement (5.18) then implies that an expansion in Hermite polynomials is appropriate, with normalization constant γ_n as in (5.17). The definition of Hermite polynomials in 5.2.1 accounts for the scaling in the PDF.

Multivariate expansion

It is rarely the case that there is only one source of uncertainty in a system. Also many sources of uncertainty are more realistically modeled as a random process and require characterization in several random variables (c.f. 5.2.3).

Consider now a d-dimensional random vector with mutually independent components $\vec{\chi} = (\chi_1, \ldots, \chi_d)$. Independence ensures that the distribution $F_\chi(y_1, \ldots, y_d) = \prod_{i=1}^d F_{\chi_i}(y_i)$, where $F_{\chi_i}(y_i)$ is the marginal distribution of χ_i. Let each χ_i be associated with a gPC basis $\{\psi_k(\chi_i)\}_{k=0}^{N_p}$ for $\mathbb{P}_{N_p}(\chi_i)$. Define a multi-index $\mathbf{i} = (i_1, \ldots, i_d)$ and let $|\mathbf{i}| = i_1 + \cdots + i_d$. Then the gPC basis functions of $\mathbb{P}_{N_p}^d$, the space of d-variate polynomials of degree at most N_p, are $\{\mathbf{\Psi}(\vec{\chi})\}_{|\mathbf{i}|=0}^{N_p}$ where

$$\mathbf{\Psi}_\mathbf{i}(\vec{\chi}) = \psi_{i_1}(\chi_1) \ldots \psi_{i_d}(\chi_d). \tag{5.21}$$

Again owing to the independence assumption, the orthogonality relation is

$$\mathbb{E}\left[\mathbf{\Psi}_\mathbf{i}(\vec{\chi})\mathbf{\Psi}_\mathbf{j}(\vec{\chi})\right] = \gamma_\mathbf{i}\delta_\mathbf{ij}, \tag{5.22}$$

where $\gamma_\mathbf{i} = \gamma_{i_1} \ldots \gamma_{i_d}$ and $\delta_\mathbf{ij} = \delta_{i_1 j_1} \ldots \delta_{i_d j_d}$. The space $\mathbf{P}_{N_p}^d$ has dimension $\binom{N_p+d}{N_p}$ [70].

5.2.3 Karhunen-Loève expansion

To apply the techniques discussed in this chapter, it is necessary to express the uncertainty in terms of a finite number of mutually independent random variables. Let $X(s) \equiv X(s;\omega), \omega \in \mathcal{W}$, be a random process on $\mathbb{R} \times \mathcal{W}$. For a fixed ω, $X_\omega(s)$ is a sample path of the process. For a fixed s, $X_s(\omega)$ is a random variable on $(\mathcal{W}, \mathcal{F}, P)$. The process has a mean-value $m_X(s) = \mathbb{E}[X_s]$ and a covariance function given by $K_x(r,s) = \mathbb{E}\left[(X_r - m_X(r))\,(X_s - m_X(s))\right]$. Without loss of generality, consider only processes with $m_X(s) = 0$ to simplify notation. The process is (covariance) *stationary* if $K_x(r,s) = K_x\left(|r-s|\right) = K_x(\sigma)$. It is common in practice to characterize stationary processes in terms of the *spectrum*

$$S_x(\xi) = \int R_x(\sigma) e^{-i\xi\sigma} d\sigma, \tag{5.23}$$

which is the Fourier transform of the correlation function $R_x(\sigma) = R_x\left(|r-s|\right) = \mathbb{E}\left[X_s X_r\right]$. For the zero-mean processes under consideration here, $R_x(\sigma) = K_x(\sigma)$.

The Karhunen-Loève (KL) expansion provides a partial characterization of the process in terms of a series expansion in orthogonal functions,

$$X(s,\omega) = m_X(s) + \sum_{i=1}^{\infty} \varphi_i(s) X_i(\omega),\ 0 \le s \le S. \tag{5.24}$$

The orthogonality corresponds to the condition that the X_i are uncorrelated, i.e.,

$$\mathbb{E}\left[X_i X_j\right] = \lambda_i \delta_{ij}. \tag{5.25}$$

The λ_i correspond to the expected energy of the component $\varphi_i(s)$. The random variables are defined by

$$X_i = \int_0^S X(s,\omega)\varphi_i(s)ds. \tag{5.26}$$

The λ_i and $\varphi_i(s)$ are eigenvalue - eigenfunction pairs that solve the integral equation [67]

$$\lambda_i \varphi_i(s) = \int_0^S K_x(r,s)\varphi_i(r)dr. \tag{5.27}$$

Equation (5.27) follows from substituting (5.26) into (5.25). The kernel function is the covariance, which is necessarily positive semi-definite and symmetric. These properties imply that (5.27) possesses at least one solution with $\lambda \neq 0$, the eigenvalues are all non-negative and there are at most countably many. Furthermore, if $K_x(r,s)$ is positive definite, all eigenvalues are positive and the eigenfunctions form a complete, orthonormal set and the KL expansion converges for $s \in [0,S]$ in the mean square sense [67],

$$\lim_{d\to\infty} \mathbb{E}\left[X(s,\omega) - \sum_{i=1}^{d} \varphi_i(s)X_i(\omega)\right] = 0. \tag{5.28}$$

Equation (5.28) assumes that $m_X(s) = 0$.

Gaussian processes

The KL procedure results in an expansion of a random process in uncorrelated random variables. This is where the Gaussian process becomes useful. A Gaussian random process $X(s,\omega)$ is a random process for which every linear functional is a Gaussian random variable. That is, if $I \subset \mathbb{R}$ is an interval on which $X(s,\cdot)$ is defined, then

$$Y(\omega) = \int_I f(s)X(s,\omega)ds \tag{5.29}$$

is a Gaussian random variable for every $f(s)$ such that Y has finite second moments. This work considers only Gaussian processes for the following reasons:

1. The central limit theorem [70] suggests that the Gaussian model is appropriate for many physical systems.
2. The X_i in (5.26) are Gaussian by definition, if $X(s,\omega)$ is a Gaussian process.
3. Since the X_i are Gaussian and uncorrelated by design, they are independent so the multivariate gPC assumptions are satisfied.

In light of (5.25), one may express (5.24) in terms of standard $N(0,1)$ random variables,

$$X(s,\omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\varphi_i(s)\chi_i(\omega),\ 0 \le s \le S. \tag{5.30}$$

It is not always the case that the Gaussian model is an appropriate one. For instance, it may not always be realistic to model certain phenomena such as scattering over the real line. Even allowing the SSP to satisfy the Gaussian process model is unrealistic in the sense that $c(x, z) > 0$; it may be more appropriate to use a truncated Gaussian or Beta distribution. However, the same techniques apply in practice in the non-Gaussian case; only the mathematics are not as tidy since uncorrelated random variables need not be independent.

Additionally, in practice one must truncate the expansion in (5.24) to include only a finite number of terms. The decay rate of the eigenvalues λ_i as i increases is related to the decay rate of the spectrum $S_X(\xi)$. Alternately, the decay rate of the λ_i is inversely related to the spread of the correlation function $R_X(s)$. For example, if $X(s, \omega)$ is an uncorrelated (white) process, then $K_x(r, s)$ is an impulse, and the eigenvalues are all equal. In fact, the KL expansion does not converge in this case since the kernel is not in $L^2(0, S)$. On the other hand, if $S_X(\xi)$ has compact support, i.e., $X(s, \omega)$ is band-limited with bandwidth W, then the eigenvalues decay very rapidly, in fact only the eigenvalues up to $i = 2WS + 1$ are significant [67]. More detailed estimates are available in [69].

Rational spectrum

The rational spectrum case allows an expansion for which an analytical solution to (5.27) is available. Again, this choice is primarily for convenience and to show that

the general techniques apply within the level sets framework. The full derivation is given in [67]. Consider the spectrum for a process $X(s,\omega)$,

$$S_X(\xi) = \frac{2a}{a^2\xi^2 + 1}. \tag{5.31}$$

The covariance function associated with (5.31) is

$$K_x(r,s) = \exp\left(-|s-r|/a\right), \tag{5.32}$$

and a is the correlation length. The correlation length will affect the eigenvalue decay rate, as discussed above. The integral equation becomes

$$\int_0^S \exp\left(-|s-r|/a\right)\varphi(r)dr = \lambda\varphi(s), \quad -S \leq s \leq S. \tag{5.33}$$

The interval is symmetric about zero to make the results more tractable. Differentiating (5.33) twice with respect to s yields the equivalent differential equation

$$\ddot{\varphi}(s) = \frac{1}{a}\left[\frac{1}{a} - \frac{2}{\lambda}\right]\varphi(s). \tag{5.34}$$

This equation has non-trivial solutions only for $0 < \lambda < 2a$, in that case the eigenfunctions are complex exponentials,

$$\varphi(s) = Ae^{i\alpha s} + Be^{-i\alpha s} \tag{5.35}$$

$$\alpha^2 = \frac{2a - \lambda}{a^2\lambda}. \tag{5.36}$$

$$\tag{5.37}$$

Substitution of (5.35) into (5.33) shows that the eigenvalues λ_j satisfy

$$\begin{cases} 1 + a\alpha_j \tan(\alpha_j S) = 0, & j \text{ odd} \\ a\beta_j + \tan(\beta_j S) = 0, & j \text{ even}, \end{cases} \tag{5.38}$$

with

$$\lambda_j = \begin{cases} \frac{2a}{1+a^2\alpha_j^2}, & j \text{ odd} \\ \frac{2a}{1+a^2\beta_j^2}, & j \text{ even}. \end{cases} \tag{5.39}$$

The associated eigenfunctions are

$$\varphi_j(s) = \begin{cases} \frac{\cos(\alpha_j s)}{\sqrt{S+\sin(2\alpha_j S)/(2\alpha_j)}}, & j \text{ odd} \\ \frac{\sin(\beta_j s)}{\sqrt{S-\sin(2\beta_j S)/(2\beta_j)}}, & j \text{ even}. \end{cases} \tag{5.40}$$

For a process on $[0, 2S]$, substitute $r = s+S$ into the above, and (5.33) is still satisfied for $\tilde{\varphi}(r) \equiv \varphi_j(r - S)$. The correlation length a controls the eigenvalue decay rate - if a is small, the eigenvalues decay very slowly. For large a, they decay rapidly. For small a, any truncation will result in significant error. This technique therefore works

best for larger values of a. One may then truncate the series based on a specified error tolerance. Figure 5.1 displays the eigenvalues as a function of the number of expansion terms for $a = 1, 5, 10, 20$. For $a = 1$, the eigenvalues decay quite slowly.

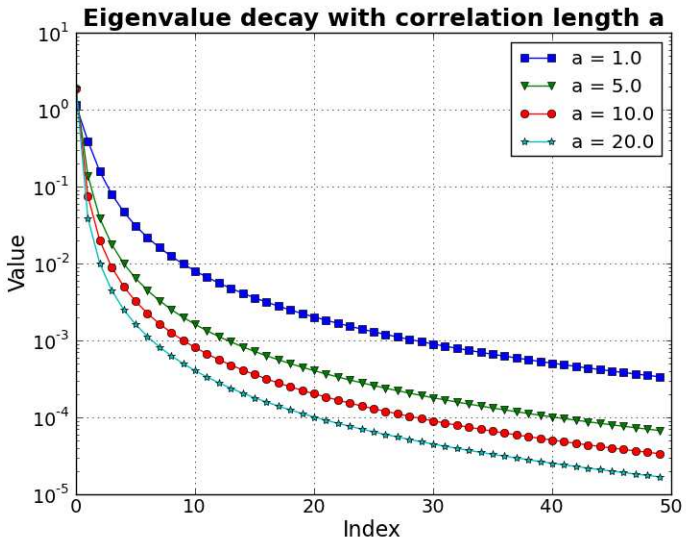


Figure 5.1: Eigenvalues for the exponential covariance function with correlation length a

The rational spectrum extends readily to two dimensions as well. Let

$$K_x\left((r_1, r_2), (s_1, s_2)\right) = \exp\left(-|s_1 - r_1|/a_1 - |s_2 - r_2|/a_2\right), \tag{5.41}$$

and assume that $\varphi(s_1, s_2) = \varphi^{(1)}(s_1)\varphi^{(2)}(s_2)$. Then

$$\begin{aligned}
\lambda\varphi^{(1)}(s_1)\varphi^{(2)}(s_2) &= \int\int K_x\left((r_1, r_2), (s_1, s_2)\right) dr_1 dr_2 \\
&= \int \exp\left(-\frac{|s_1 - r_1|}{a_1}\right)\varphi^{(1)}(r_1)dr_1 \int \exp\left(-\frac{|s_2 - r_2|}{a_2}\right)\varphi^{(2)}(r_2)dr_2 \\
&= \lambda^{(1)}\varphi^{(1)}(s_1)\lambda^{(2)}\varphi^{(2)}(s_2).
\end{aligned}$$

So if $a_1 \neq a_2$, the eigenvalues and eigenfunctions are just products of the one-dimensional eigenvalues and eigenfunctions. If $a_1 = a_2$, each eigenvalue corresponds to two eigenfunctions so the basis functions are

$$\varphi_k(s_1, s_2) = \frac{1}{\sqrt{2}} \left(\varphi_i^{(1)}(s_1)\varphi_j^{(2)}(s_2) + \varphi_j^{(1)}(s_1)\varphi_i^{(2)}(s_2) \right). \tag{5.42}$$

5.3 Stochastic Collocation

Recall that the goal is to solve (4.8):

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

for $l = 1, 2$ with V as in (2.19). Now $\phi_l = \phi_l(t, x, z, \theta; \vec{\chi})$, with $\vec{\chi} = (\chi_1, \ldots, \chi_d)$ a random vector with mutually independent components, $\vec{\chi} : \mathcal{W} \to \mathbb{R}^d$. The initial conditions are $\phi_l(0, \cdot) = \phi_l^0(\cdot)$, and appropriate boundary conditions for $(x, z, \theta) \in \Omega \times [-\pi, \pi)$ are available. For a fixed $\omega_0 \in \mathcal{W}$, ϕ_l is a deterministic function of $(t, x, z, \theta, \vec{\chi}(\omega_0))$, with $\vec{\chi}(\omega_0)$ representing a set of input parameters. Thus one may solve (4.8) using the techniques presented in chapters 2, 3, and 4. In the stochastic collocation method, one prescribes a set of nodes, $\Theta_M = \left\{ \vec{\chi}^{(k)} \equiv \vec{\chi}(\omega^{(k)}) \right\}_{k=1}^{M}$ and applies the deterministic solver to compute a corresponding solution set, $\left\{ \phi_l^{(k)}(\cdot; \vec{\chi}^{(k)}) \right\}_{k=1}^{M}$, $l = 1, 2$. Formally, the stochastic collocation problem is to find $\tilde{\phi}_l(\cdot, \vec{\chi}) \in \mathbb{P}_{N_p}^d$ such

that $\lim_{M\to\infty} ||\tilde{\phi}_l(\cdot,\vec{\chi}) - \phi_l(\cdot,\vec{\chi})|| = 0$, where the norm is typically the mean square norm. There are two methods for constructing the collocation solution, interpolation and discrete projection (pseudospectral) [70]. This work focuses on the discrete projection.

5.3.1 Relation to Monte Carlo

Monte Carlo sampling methods are common techniques for solving systems depending on random parameters. The Monte Carlo method uses independent samples drawn randomly from the distribution $F_{\vec{\chi}}(\chi_1,\ldots,\chi_d)$ to generate the $\vec{\chi}^{(k)}$, $k = 1,\ldots,M$. That is, one generates the set of nodes, Θ_M, from the random distrubution, then applies the deterministic solver at each node in the set. The sample mean, $\bar{\phi}_l(t,x,z,\theta) = \frac{1}{M}\sum_{k=1}^{M}\phi_l^{(k)}(t,x,z,\theta;\vec{\chi}^{(k)})$, and the sample variance, $\sigma^2_{\phi_l}(t,x,z,\theta) = \frac{1}{M-1}\sum_{k=1}^{M}\left(\phi_l^{(k)}(t,x,z,\theta;\vec{\chi}^{(k)}) - \bar{\phi}_l(t,x,z,\theta)\right)^2$ provide estimates for the first and second moments. The advantages of using Monte Carlo are that it is very simple to implement, and that its convergence rate is unaffected by the random dimension d. The disadvantage is that the convergence rate is very slow; the mean-square error decays like $\frac{1}{\sqrt{M}}$. The stochastic collocation method on the other hand, uses a deterministic set of nodes.

5.3.2 Discrete Projection

Express $\tilde{\phi}_l$ as the orthogonal gPC projection onto $\mathbb{P}^d_{N_p}$, and drop the subscript l to simplify notation,

$$\tilde{\phi} = \sum_{|\mathbf{k}|=0}^{N_p} \hat{\phi}_{\mathbf{k}} \Psi_{\mathbf{k}}(\vec{\chi}). \tag{5.43}$$

The $\Psi_{\mathbf{k}}(\vec{\chi})$ are the basis functions of the expansion as in (5.21); take them as scaled to form an orthonormal basis. The $\hat{\phi}_{\mathbf{k}}$ are the expansion coefficients,

$$\hat{\phi}_{\mathbf{k}} = \int \phi(\cdot;\chi)\Psi_{\mathbf{k}}(\chi) dF_{\vec{\chi}}(\chi). \tag{5.44}$$

Computing the $\hat{\phi}_{\mathbf{k}}$ yields an expression of the solution $\tilde{\phi}(\cdot;\vec{\chi})$ for any $\vec{\chi}$. This is in contrast to a Monte Carlo procedure, which only provides estimates of moments. Since the function values are only available as discrete samples, quadrature approximates the integral (5.44), or if $d > 1$, its multi-variate extension cubature rule. The term *pseudospectral* often describes this technique since the optimal choice of nodes and weights makes this the same as a Galerkin method [4].

There are two sources of error for the discrete projection method. An application of the triangle inequality for $L^2_{dF_{\vec{\chi}}}$ shows that

$$||\tilde{\phi} - \phi|| \leq ||\tilde{\phi} - P_{N_p}\phi|| + ||P_{N_p}\phi - \phi||. \tag{5.45}$$

The first term on the right hand side of (5.45) is the error results from approximating

the integrals (5.44), this is the *aliasing* error [70]. The second term is the projection error, due to truncation of the expansion of ϕ.

5.3.3 Quadrature

The quadrature problem is to find a set of node-weight pairs, $\{(y_i, w_i)\}$ with the property

$$\int_I f(y)dy = \sum_{i=0}^{M} f(y_i)w_i + O(\Delta y^M). \tag{5.46}$$

A straightforward approach is to divide the interval I into a partition, $y_i = y_{i-1} + h_i$ for $i = 1, \ldots, M-1$ with y_0, y_M being the interval endpoints. Then on each subinterval $[y_{i-1}, y_i]$, approximate f by some polynomial. For example, piecewise linear approximation results in the trapezoidal rule, with nodes y_i and weights

$$w_i = \begin{cases} \frac{h_i}{2}, & i = 0, M \\ h_i, & i = 1, \ldots, M-1. \end{cases}$$

Quadrature rule errors are typically expressed in terms of the highest degree polynomial for which the rule is exact. Quadratures of this sort have an error term proportional to the $(M+1)$ level derivative of f, which is zero if $f(y)$ is any polynomial of degree at most M. This is not optimal however.

The trapezoidal rule assumes that a set of nodes $\{y_i\}_{i=0}^{M} \in I$ is prescribed, and

takes $f(y)$ equal to its linear interpolant on $[y_{i-1}, y_i]$, then uses this information to compute the weights $\{w_i\}$. Gaussian quadrature seeks to optimize *both* the selection of nodes and weights, obtaining exactness for polynomials of degree up to $2M+1$. The optimal nodes are the zeros of orthogonal polynomials. Consider

$$\int_I f(y)w(y)dy, \tag{5.47}$$

where $w(y)$ is a weight function, and let $\{Q_n(y)\}_{n=0}^M$ form an orthonormal basis of $\mathbb{P}_M$ with respect to $w(y)$ on the domain I. Then the optimal set of $\{y_i\}_{i=0}^M$ is the set of zeros of $Q_{M+1}(y)$. The choice of basis depends on the weight function, e.g., Legendre polynomials are orthogonal for $w(y)=1$, and Hermite polynomials for $w(y)=e^{-y^2}$. In (5.44), the weight function should be related to $dF_{\vec{\chi}}(\chi)$, i.e., the PDF of $\vec{\chi}$. These are the Gauss nodes, and do not include the boundary points.

Tabulated values of the node-weight pairs for Hermite and Legendre polynomials are available, for instance, in [1]. Alternately, an algorithm due to Golub and Welsch [26] computes the pairs using the recurrence relation for the polynomial family to form a tridiagonal system. The weights satisfy [26]

$$w_i = -\frac{A_{M+1}}{A_M}\frac{1}{Q_M(y_i)Q'_{M+1}(y_i)}, \tag{5.48}$$

where A_k is the coefficient of y^k in $Q_k(y)$. Using the three-term recursion (5.6), the

following system holds:

$$y\begin{pmatrix} Q_0(y) \\ Q_1(y) \\ \vdots \\ Q_M(y) \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{a_0} & 0 & \dots & 0 \\ \frac{c_1}{a_1} & 0 & \frac{1}{a_1} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \frac{c_{M-1}}{a_{M-1}} & 0 & \frac{1}{a_{M-1}} \\ 0 & \dots & 0 & \frac{c_M}{a_M} & 0 \end{pmatrix} \begin{pmatrix} Q_0(y) \\ Q_1(y) \\ \vdots \\ Q_M(y) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{a_M} Q_{M+1}(y) \end{pmatrix} \tag{5.49}$$

or more succinctly using matrix notation: $y\mathbf{q} = \Lambda\mathbf{q} + \frac{1}{a_M} Q_{M+1}(y)\vec{e}_{M+1}$. The matrix Λ is tridiagonal, and further, for an orthonormal basis, Λ is symmetric. Also, if y is a zero of Q_{M+1}, then

$$\Lambda\mathbf{q} = y\mathbf{q}, \tag{5.50}$$

so the zeros $\{y_i\}$ are the eigenvalues of Λ. From (5.48) and the Christoffel-Darboux formula [31],

$$\sum_{j=0}^{M} Q_j(x)Q_j(y) = \frac{A_M}{A_{M+1}} \frac{Q_M(y)Q_{M+1}(x) - Q_{M+1}(y)Q_M(x)}{x - y}, \tag{5.51}$$

the eigenvectors of Λ yield the weights as follows. If $\mathbf{q}^{(i)}$ is the eigenvector associated with eigenvalue y_i, normalized so that $||\mathbf{q}^{(i)}|| = 1$, then

$$w_i = \left(\frac{q_0^{(i)}}{Q_0} \right)^2, \tag{5.52}$$

where $q_0^{(i)}$ is the first component of $\mathbf{q}$.

5.3.4 Cubature

To extend the quadrature rules presented in 5.3.3 to multi-dimensional integration, one might apply tensor products in order to treat the multivariate integrals in a dimension-by-dimension manner. Let $\vec{\chi} = (\chi_1, \ldots, \chi_d)$, and let m_ℓ^d be the number of quadrature points for d dimensional quadrature at level ℓ. Define the nodal set $\Theta_\ell^d = \{y_{\ell_i}\}_{i=1}^{m_\ell^d}$, and the one-dimensional quadrature rule applied to a function f

$$\mathcal{Q}_\ell^1 f = \sum_{i=1}^{m_\ell^1} w_{\ell_i} f(y_{\ell_i}). \tag{5.53}$$

The tensor product cubature rule for f is then

$$\left(\mathcal{Q}_{\ell_1}^1 \otimes \cdots \otimes \mathcal{Q}_{\ell_d}^1\right) f = \sum_{i_1=1}^{m_{\ell_1}^1} \cdots \sum_{i_d=1}^{m_{\ell_d}^1} w_{\ell_1,i_1} \ldots w_{\ell_d,i_d} f(y_{\ell_1,i_1}, \ldots, y_{\ell_d,i_d}). \tag{5.54}$$

The tensor quadrature rule is straightforward to implement, however if $m_{\ell_1}^1 = \ldots m_{\ell_d}^1 = m$, the total number of function evaluations is m^d, so there is exponential growth in dimension. Because of this exponential growth, the tensor product cubature is generally only practical for small d, say $d \leq 5$ [70]. Application of a sparse grid formulation due to Smolyak [64] renders computation more feasible for larger d. The Smolyak formulation combines lower order tensor products so that the number of function evaluations grows more slowly with d. Figure 5.2 shows the Gauss-Legendre nodes from a Smolyak sparse grid contrasted with the Gauss-Legendre nodes from a tensor quadrature grid for $\ell = 5$ and $d = 2$.

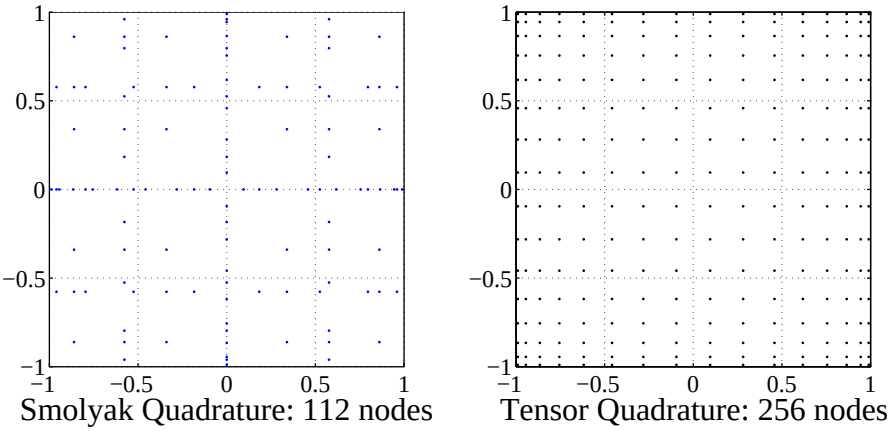


Figure 5.2: Gauss-Legendre nodes for $d = 2$, $\ell = 5$

The Smolyak cubature of level ℓ is [24]

$$\mathcal{Q}_\ell^d f = \sum_{\ell \le |\mathbf{k}| \le \ell+d-1} (-1)^{\ell+d-|\mathbf{k}|-1} \binom{d-1}{|\mathbf{k}|-\ell} \left(\mathcal{Q}_{k_1}^1 \otimes \cdots \otimes \mathcal{Q}_{k_d}^1\right) f, \tag{5.55}$$

where $\mathbf{k} = (k_1, \ldots, k_d)$ is a multi-index. The effect is to use only a subset of the nodes in the full d-dimensional tensor grid. If the level ℓ quadrature formula, $\mathcal{Q}_\ell^1$, uses $2^{\ell-1}$ nodes, then it can be shown that the number of function evaluations is bounded above by

$$M \le C(d) 2^{\ell+d-1} (\ell + d - 1)^{d-1} \tag{5.56}$$

with $C(d)$ a constant depending on d, and the cubature is exact for d-variate polynomials of degree at most $2\ell - 1$ if $\ell < 2d + 1$, otherwise if $\ell = \alpha d + \beta$, the cubature is exact up to degree $2^\alpha(d + \beta) - 1$ [45]. If the quadrature grids satisfy a nesting

property i.e. $\Theta_\ell \subset \Theta_{\ell+1}$, then the bound (5.56) on the total number of nodes in the Smolyak cubature $\mathcal{Q}_\ell^d$ is even lower, but this is not the case for the Gauss-Hermite nodes which are applied in this work. It is also critical that the number of nodes in the one-dimensional component $\mathcal{Q}_\ell^1$ is $2^{\ell-1}$ at least asymptotically in d. Table 5.1 presents a few values of $\frac{M_T}{M_S}$, the ratio of the total number of nodes required for tensor product cubature to the number of nodes for the Smolyak formulation. The values in Table 5.1 were computed using the expression resulting from the results $M_T = 2^{(\ell-1)d}$ and $M_S = \sum_{k=\ell}^{\ell+d-1} \binom{k-1}{d-1} 2^{k-d}$. It is apparent that the advantage of using the sparse grid increases rapidly for $d \geq 4$, whereas for $d < 4$ it may be preferable to employ tensor products since they are much easier to implement.

Table 5.1: Ratio of total number of nodes in tensor product grid to number of nodes in Smolyak sparse grid

d	$\ell = d$	$\ell = d+1$	$\ell = d+2$
2	0.8	1.0	1.5
3	2.1	4.7	11.9
4	19.6	85.3	410.9
5	712.8	6098.6	56,716.

The results quoted above assume the cubature domain $I = I_1 \times \cdots \times I_d$, with $I_i \subset \mathbb{R}$ bounded; to extend to cubature or quadrature over unbounded domains, there are a few options. One is to truncate the domain so that it is bounded. Another is to use a mapping to a bounded domain and use, for example, Legendre points. This work applies Gauss-Hermite quadrature which is defined on the unbounded intervals. Domain truncation is not unreasonable for problems involving Gaussian PDFs, due to the exponential decay, though Gauss-Hermite quadrature appears to

offer improved performance based on experimentation. This work does not address the domain mapping approach, but [58] offers a review of the different approaches.

5.3.5 Computing the statistics

The stochastic collocation yields an approximate solution $\tilde{\phi}(\cdot;\vec{\chi})$ for any $\vec{\chi}$ in the random space. The moments are then readily computed by passing the expectation over the sum, e.g.,

$$\mathbb{E}\left[\tilde{\phi}\right] = \hat{\phi}_0, \tag{5.57}$$

$$\mathrm{Var}\left[\tilde{\phi}\right] = \sum_{|\mathbf{k}|=1}^{N_p} \hat{\phi}_{\mathbf{k}}^2. \tag{5.58}$$

5.4 Results

5.4.1 Single uniform random variable

The first experiment is to perform a basic study using a single random parameter. For this, let

$$c(x,z) = c_0 + \epsilon Y, \tag{5.59}$$

where Y is a uniform random variable on $[-1, 1]$ so that $\mathbb{E}[Y] = 0$ and $\mathrm{Var}[Y] = \frac{1}{3}$. This suggests a Legendre polynomial chaos expansion, and the samples $\left\{Y^{(k)}\right\}_{k=0}^{N_p}$ are given by the Gauss-Legendre quadrature points. The constant sound speed term allows for the computation of errors since the solution is known given the initial conditions 3.1, i.e.,

$$\phi_1(t, x, z, \theta; Y) = x - (c_0 + \epsilon Y)t\cos\theta \tag{5.60}$$

$$\phi_2(t, x, z, \theta; Y) = z - z_s - (c_0 + \epsilon Y)t\sin\theta \tag{5.61}$$

A Monte Carlo sampling scheme applied to the same problem provides a comparison. Figures 5.3 and 5.4 show the errors in computed mean and variance of $\mathbf{\Phi}$ respectively, computed using the L_∞ norm for both the Monte Carlo and Stochastic Collocation approaches at four fixed times. Since the gPC expansion is exact for the single uniform random variable model, the stochastic collocation converges immediately, with the error attributable to numerical quadrature and the error in the level set method solver only. The Monte Carlo simulation requires about 1000 samples to converge to the same level. It is also noteworthy that the errors are time-dependent. As the time at which the errors are computed increases, so does the number of samples required for Monte Carlo to match the errors in the stochastic collocation. For this simple case, the stochastic collocation method has a distinct advantage.

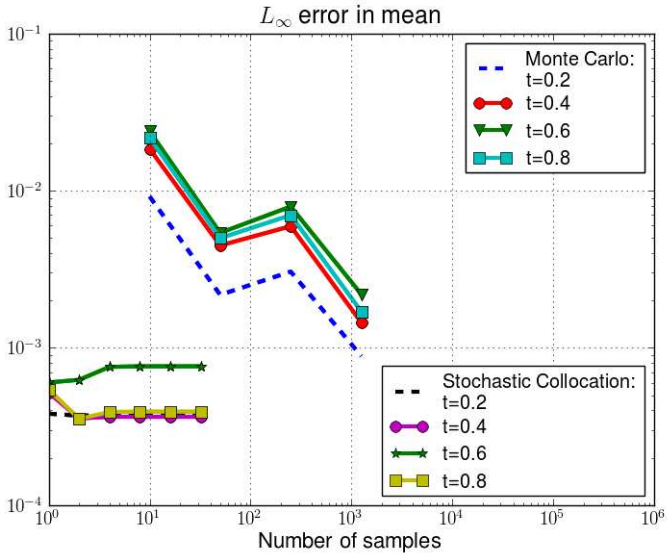


Figure 5.3: Error in Mean for Uniform Random Variable

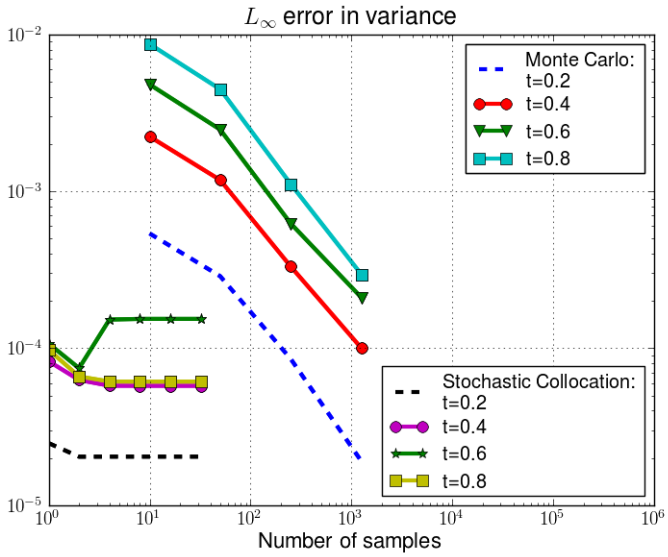


Figure 5.4: Error in Variance for Uniform Random Variable

5.4.2 Gaussian random field

Now let

$$c(x, z) = c_0 + \epsilon X(x, z; \omega), \tag{5.62}$$

where $X(x, z; \omega)$ is a zero-mean, Gaussian random field with exponential covariance as in 5.41. A KL expansion decomposes the process into independent, Gaussian random variables. The techniques in this chapter then apply directly to the expansion. The correlation lengths applied here are $a_x = 10.0$ and $a_z = 6.0$. These values ensure the eigenvalues decay rapidly so as to limit the amount of computation. Figures 5.5 and 5.6 show the error in mean and the computed variance for a few fixed times, for both stochastic collocation and Monte Carlo with $N_s = 500$ random samples. The eigenvalue tolerances were set to yield $d = 1, 2, 3, 4, 5$, and the stochastic collocation expansion is cubic. The result from Monte Carlo with $N_s = 15,000$ serves as the point of comparison in the absence of exact solutions. The behavior for the stochastic collocation examples is very close to that for Monte Carlo with respect to d. Again the error increases with time. These results show that the statistics derived from the stochastic collocation expansion are consistent with the Monte Carlo statistics. Increasing d does not appear to have an error-reducing effect, however it is likely that the error incurred by cubature as dimension d increases offsets the truncation error in the random field expansion.

Alternately, Figures 5.7 and 5.8 show how the results vary for fixed dimension

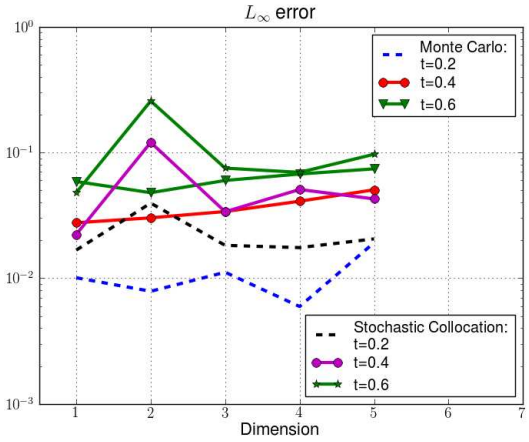


Figure 5.5: Approximate error in mean for Gaussian process by dimension d, times $t = 0.2, 0.4, 0.6$ seconds

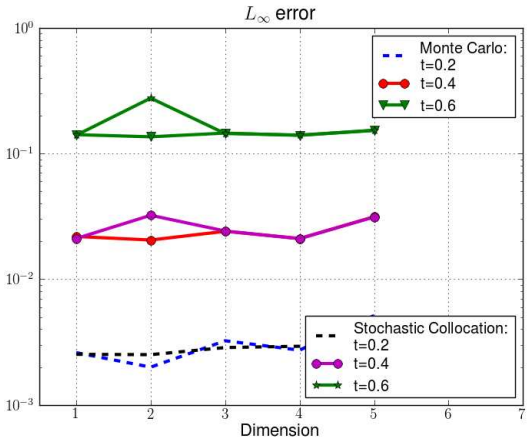


Figure 5.6: Approximate error in variance for Gaussian process by dimension d, times $t = 0.2, 0.4, 0.6$ seconds

$d = 3$ with the number of evaluations of the level set method for the same scenario. The errors are clearly smaller for the collocation approach at a comparable number of function calls.

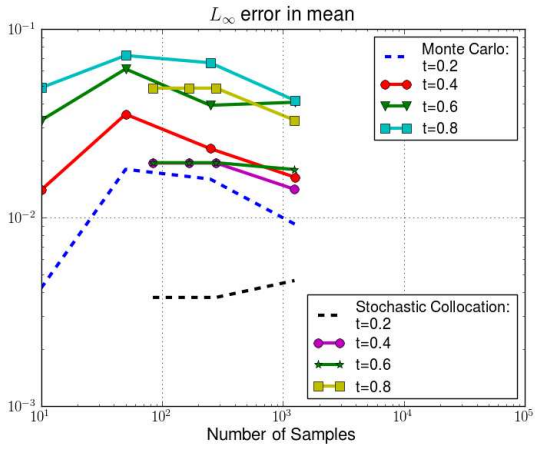


Figure 5.7: Approximate error in mean for Gaussian process by number of function evaluations, times $t = 0.2, 0.4, 0.6, 0.8$ seconds

Finally, Figure 5.9 shows extracted wavefronts from the computed means of the level set function components provided by a stochastic collocation formulation with $d = 3$ and $N_p = 3$. This serves as an example of how one might generate random wavefront samples using the expansion coefficients $\hat{\phi}_{n,l}(t, x, z, \theta)$, $l = 1, 2$ provided by the stochastic collocation representation.

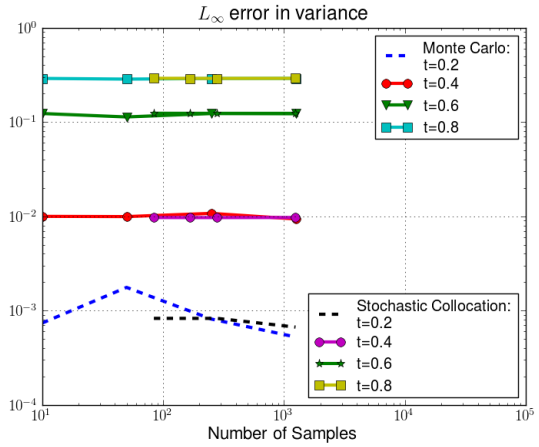


Figure 5.8: Approximate error in variance for Gaussian process by number of function evaluations, times $t = 0.2, 0.4, 0.6, 0.8$ seconds

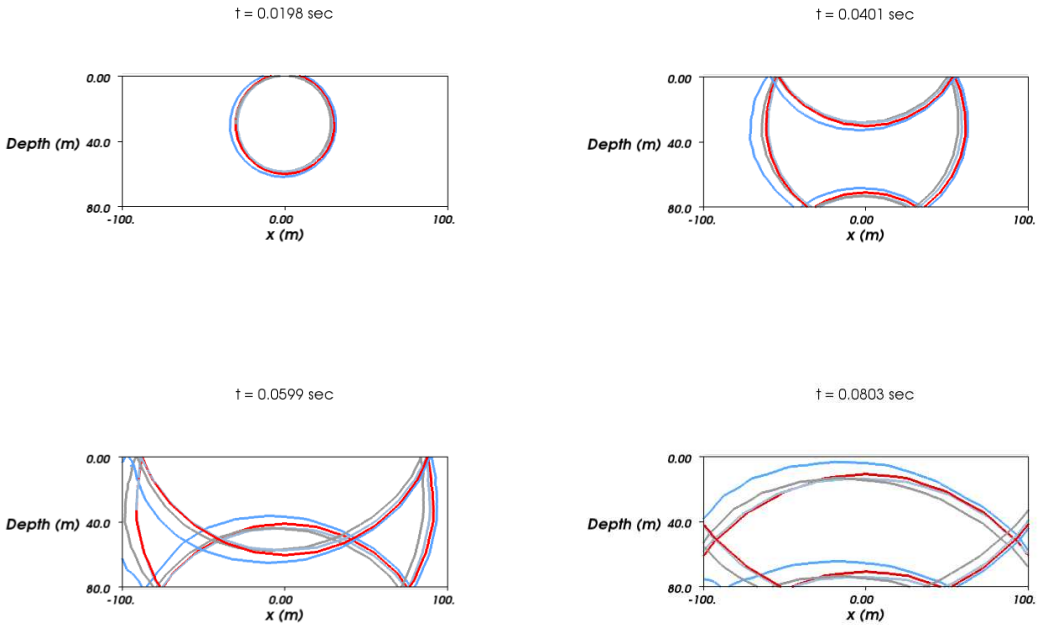


Figure 5.9: Wavefronts extracted from random samples of level set functions for the SW06 SSP data with a Gaussian random process

Chapter Six

Conclusion

6.1 Concluding Remarks

The main contribution of this work is the establishment of a foundation for a full wavefront propagation model applicable to high frequency underwater acoustics. The existence of such a model would fill a gap in the existing capability of ray trace models, thus it is potentially very useful. By combining WENO methods with the framework set forth in [47], the resulting method provides accurate wavefront information in the presence of multiple reflections. This work also includes a wavefront-based semi-Lagrangian method for computing spreading loss. Proposed high order boundary conditions for reflection add to the algorithm, and a method based on a reflection principle allows for greater accuracy in representing reflections from arbitrary geometries. An analysis of arrival time errors close to the embedded boundary confirms that the proposed method converges faster than $O(h)$. Finally, an exploration of stochastic collocation models allows for inclusion of random sound speeds in the method, which in some cases, converge to the true statistics much faster than traditional Monte Carlo approaches. The model has been verified using problems with analytical solutions, and also with comparisons to full wave equation and ray solvers. Examples using smooth boundaries and examples using sound speed profile data collected in a recent sea test exhibit the method's capabilities. On a smaller scale, this work also includes prescription for WENO interpolation at an arbitrary point, an analytical solution to the level set equations with linear wave speed, and a routine to extract and display the embedded wavefronts.

6.2 Future Research Directions

The results in this thesis form the groundwork for further study. Topics for future work loosely comprise three categories: applications, extensions, and computational considerations. Applications consist of improvements geared toward transition to programs in acoustics. Extensions are theoretical considerations that the present work does not address. Computational concerns involve optimizing the implementation for efficiency. The examples below offer some suggestions.

1. Applications

    - Refine the amplitude model to extend beyond bottom bounces. Consider adapting approaches traditionally used in ray theory to address issues, such as caustics, that arise from the high frequency approximation.
    - Incorporate scattering models and absorption loss.
    - Obtain more sound speed and bathymetry data and compare with experimental results to validate the model.
    - Embed multiple sources/receivers to study more complicated scenarios.
    - Integrate with shallow water reverberation models as an alternative to ray tracing. Investigate the stochastic collocation approach for generating wavefront samples.
    - Investigate more realistic spectra for sound speed fluctuations.

2. Extensions

   - Apply the random field model to additional sources of uncertainty, such as rough boundary scattering. This may be implemented initially by representing the reflected angle in a random space.
   - Incorporate mid-frequency effects, e.g. diffraction and amplitude scattering. One could begin with the geometric diffraction model [37], for instance.
   - Extend the method to work with moving source and receiver.
   - Extend to long range propagation, considering the work in [53] and [54]. Such a model could be applied to study internal waves using the stochastic collocation approach described in Chapter 5.

3. Computational

   - Localize the WENO solver using techniques in [50]; significant speed-up should be achievable by solving the level set equations only near the wavefronts.
   - The ocean is not two-dimensional, and is not always well-approximated by a two-dimensional model, thus extension to three dimensional wavefront propagation is of interest. The extension is conceptually straightforward, but computationally challenging.
   - Optimize data structures and routines.

Appendix A

WENO reconstruction at an arbitrary point

Suppose a function $f(x)$ is given, with values known at the points on the grid $\{x_i\}_{i=0}^{N-1}$. As in the setup in Chapter 2, the points x_i lie at the center of a cell I_i, defined by the edge points $\left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$ where the grid is uniform with spacing Δx so that $x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}$. The function $f(x)$ may be non-smooth and it is necessary to find the value at an arbitrary point x^*, where $x^* \in I_i$ for some $i \in \{0, 1, \ldots, N-1\}$. Because of the presence of discontinuities, it is desirable to construct a specialized interpolation scheme that captures the behavior of the function accurately in the smooth regions and limits the degradation of the approximation in non-smooth regions of the grid. Such a scheme is useful for applying the reflection boundary condition for the level set method. This section describes how the WENO procedure is adapted to handle this problem. The WENO procedure results in a degree k polynomial representation of the function $h(x)$, denoted $p(x)$, which relates to $f(x)$ as in (2.28),

$$p(x) = \sum_{j=0}^{k-1} \varsigma_j x^j,$$

with the constants ς_j, $j = 0, \ldots, k-1$ determined by the WENO procedure. In order to find a more convenient form for computing, note that the ultimate goal is to approximate the value of $f(x^*)$ using the values of $f(x)$ at the cell centers, $\{f_i\}_{i=0}^{N-1}$. In fact, for the approximation over the the stencil $S_r(i) = \{x_{i-r}, x_{i-r+1}, \ldots, x_{i-r+k-1}\}$, if $p_f(x)$ is the approximating polynomial to $f(x)$ on cell I_i, then

$$p_f^{(r)}(x^*) = \sum_{j=0}^{k-1} \hat{c}_{rj} f_{i-r+j}, \tag{A.1}$$

for $r = 0, \ldots, k-1$. One should select the constants $\hat{c}_{rj}$ so as to achieve an order k approximation to the function f. To find the constants, let x^* be located in the cell I_i, and define (on a uniform grid)

$$\alpha = \frac{x^* - x_{i-1/2}}{\Delta x} \tag{A.2}$$

so that $0 \leq \alpha < 1$ and

$$x^* = x_{i-1/2} + \alpha \Delta x. \tag{A.3}$$

Given expression (A.3), assume f is smooth in the cell I_i, expand f in a Taylor series about the point $x_{i-1/2}$ (for convenience, this way α is non-negative), and substitute into the sum, matching terms to achieve the desired order of accuracy. Table A.1 lists the resulting constants for $k = 1, 2, 3$.

Table A.1: Constants for reconstruction in (A.1), for $k = 1, 2, 3$

k	r	$j=0$	$j=1$	$j=2$
1	0	1		
2	0	$\frac{3}{2} - \alpha$	$\alpha - \frac{1}{2}$	
	1	$\frac{1}{2} - \alpha$	$\alpha + \frac{1}{2}$	
3	0	$1 - \frac{(\alpha-\frac{1}{2})(\frac{7}{2}-\alpha)}{2}$	$(\alpha - \frac{1}{2})(\frac{5}{2} - \alpha)$	$\frac{(\alpha-\frac{3}{2})(\alpha-\frac{1}{2})}{2}$
	1	$1 - \frac{(\alpha+\frac{1}{2})(\frac{5}{2}-\alpha)}{2}$	$(\alpha + \frac{1}{2})(\frac{3}{2} - \alpha)$	$\frac{(\alpha+\frac{1}{2})(\alpha-\frac{1}{2})}{2}$
	2	$1 - \frac{(\alpha+\frac{3}{2})(\frac{3}{2}-\alpha)}{2}$	$(\alpha + \frac{3}{2})(\frac{1}{2} - \alpha)$	$\frac{(\alpha+\frac{3}{2})(\alpha+\frac{1}{2})}{2}$

Following the above procedure results in k approximations to the value $f(x^*)$, denoted $f^{*,(r)} \equiv p_f^{(r)}(x^*)$, $r = 0, \ldots, k-1$. As in the development of WENO presented

in Chapter 2, if the function $f(x)$ is smooth, there are constants $\hat{d}_r$ such that

$$f^* = \sum_{r=0}^{k-1} \hat{d}_r f^{*,(r)} = f(x^*) + O(\Delta x^{2k-1}). \tag{A.4}$$

To find these coefficients, expand $f^{*,(r)}$ about x^* and match terms in the expansion. Consistency always results in the condition $\sum_{r=0}^{k-1} \hat{d}_r = 1$. Table A.2 lists the constants $\hat{d}_r$ for $k = 1, 2, 3$.

Table A.2: Constants $\hat{d}_r$ from (A.4), for $k = 1, 2, 3$

k	d_0	d_1	d_2
1	1	-	-
2	$\frac{\alpha+\frac{1}{2}}{2}$	$\frac{\frac{3}{2}-\alpha}{2}$	-
3	$\frac{(2\alpha+3)(2\alpha+1)}{48}$	$\frac{(2\alpha+3)(5-2\alpha)}{24}$	$\frac{(2\alpha-3)(-5+2\alpha)}{48}$

The resulting WENO procedure for approximating the value of $f(x^*)$ follows from applying Algorithm 2 directly to the values $f^{*,(r)}$ in place of $h^{(r)}_{i+1/2}$ and $h^{(r)}_{i-1/2}$.

Appendix B

Analytical solution to the level set equation with linear wave speed

The objective of this appendix is to solve (3.2) for the level set equation of two-dimensional acoustics:

$$f_t + c(x,z)\cos(\theta)f_x + c(x,z)\sin(\theta)f_z + (c_x\sin(\theta) - c_z\cos(\theta))f_\theta = 0$$

subject to an initial condition, $f(0,x,z,\theta) = f_0(x,z,\theta)$. This is the transport equation that evolves the wavefront in the direction of its normal vector which is the local ray direction. The level set method solves equation (3.2) twice for two component level set functions that are orthogonal to one another at the wavefront. This is consistent with Osher's formulation for geometric optics. In the case $c(x,z) \equiv c_0$ is constant, the solution is $f(t,x,z,\theta) = f_0(x - c_0 t\cos(\theta), z - c_0 t\sin(\theta), \theta)$ (the characteristics are straight lines so $\theta(t) = \theta_0$). However it is useful to have a test case with an analytical solution available for variable $c(x,z)$ to more adequately test the implementation in the three dimensional phase space.

To derive a solution for the sound speed profile that is linear in z: $c(x,z) = \alpha z + \beta$, where α and β are prescribed constants, the level set function becomes

$$f_t + (\alpha z + \beta)\cos(\theta)f_x + (\alpha z + \beta)\sin(\theta)f_z - \alpha\cos(\theta)f_\theta = 0. \tag{B.1}$$

First simplify by introducing $y = \alpha z + \beta$, then (B.1) reads

$$f_t + y\cos(\theta)f_x + \alpha y\sin(\theta)f_y - \alpha\cos(\theta)f_\theta = 0. \tag{B.2}$$

Proceed by solving the characteristic equations and use the characteristics to construct the solution. While the method of characteristics has its drawbacks, this scenario is simple and well-behaved enough for this problem. The characteristic system for (B.2) is

$$\frac{dx}{dt} = y\cos(\theta) \tag{B.3}$$
$$\frac{dy}{dt} = \alpha y\sin(\theta) \tag{B.4}$$
$$\frac{d\theta}{dt} = -\alpha\cos(\theta) \tag{B.5}$$

Begin by solving (B.5) since it is not coupled to any of the other equations in the system. This equation is separable and has the solution

$$\theta(t) = \mathrm{gd}(-\alpha t + \gamma) = 2\tan^{-1}(\gamma e^{-\alpha t}) - \frac{\pi}{2} = \tan^{-1}(\sinh(-\alpha t + \gamma)). \tag{B.6}$$

The function $\mathrm{gd}(u)$ is the Gudermannian function [1], which has the inverse

$$\mathrm{gd}^{-1}(u) = \ln|\sec u + \tan u|. \tag{B.7}$$

The term γ is an integration constant, which can be solved for as $\gamma = \tan(\frac{\theta_0}{2} + \frac{\pi}{4})$ where θ_0 is the initial condition (ray take-off angle), but will be subsequently referred to as γ for cleaner notation.

Given a solution to $\theta(t)$, the next step is to solve for $y(t)$. Using the fact that $\sin(u - \frac{\pi}{2}) = -\cos(u)$ and the trigonometric identity $\cos(2u) = \frac{1-\tan^2 u}{1+\tan^2 u}$, equation (B.4)

transforms to

$$\frac{dy}{dt} = -\alpha y \left(\frac{1 - \gamma^2 e^{-2\alpha t}}{1 + \gamma^2 e^{-2\alpha t}} \right), \tag{B.8}$$

which has the general solution

$$y(t) = C \exp \left[-\alpha \int_0^t \frac{1 - \gamma^2 e^{-2\alpha \tau}}{1 + \gamma^2 e^{-2\alpha \tau}} d\tau \right]. \tag{B.9}$$

Integrating and using the initial condition $y(0) = y_0$ one arrives at

$$y(t) = y_0 \frac{(1 + \gamma^2) e^{-\alpha t}}{1 + \gamma^2 e^{-2\alpha t}}. \tag{B.10}$$

Next use the expressions for $\theta(t)$ (B.6) and $y(t)$ (B.10) to solve for $x(t)$. Substituting and using the trigonometric identity $\sin 2l = 2 \sin l \cos l = \frac{2 \tan l}{1 + \tan^2 l}$, one sees that x solves (B.3) which becomes

$$\frac{dx}{dt} = 2y_0 \frac{(1 + \gamma^2) \gamma e^{-2\alpha t}}{(1 + \gamma^2 e^{-2\alpha t})^2}. \tag{B.11}$$

The solution is

$$x(t) = 2y_0 \frac{(1 + \gamma^2)}{\gamma} \int_0^t \frac{\gamma^2 e^{-2\alpha \tau}}{(1 + \gamma^2 e^{-2\alpha \tau})^2} d\tau \equiv 2y_0 \frac{(1 + \gamma^2)}{\gamma} I(t). \tag{B.12}$$

The integral $I(t)$ is solved with the substitution $u = \gamma^2 e^{-2\alpha \tau}$, yielding

$$I(t) = \frac{1}{2\alpha(1 + \gamma^2 e^{-2\alpha t})} - \frac{1}{2\alpha(1 + \gamma^2)}, \tag{B.13}$$

so the final solution for $x(t)$ subject to $x(0) = x_0$ is

$$x(t) = x_0 + \frac{\gamma}{\alpha} y_0 \left(\frac{1 - e^{-2\alpha t}}{1 + \gamma^2 e^{-2\alpha t}} \right). \tag{B.14}$$

Using the fact that f is constant along its characteristic curves, construct the solution $f(t, x, z, \theta)$ from (B.6), (B.10), and (B.14) . Thus,

$$f(t, x, z, \theta) = f_0 \Big(x_0(t, x, z, \theta), z_0(t, x, z, \theta), \theta_0(t, x, z, \theta) \Big). \tag{B.15}$$

Finally, to express f properly as a function of t, x, z, and θ, for short time evolution, one may invert the initial conditions so as to substitute

$$\begin{aligned}
\theta_0 &= 2 \tan^{-1} \left(\hat{\gamma} e^{\alpha t} \right) - \frac{\pi}{2} \\
\hat{\gamma} &= \tan \left(\frac{\theta}{2} + \frac{\pi}{4} \right) \\
z_0 &= \left(z + \frac{\beta}{\alpha} \right) \frac{(1 + \hat{\gamma}^2)\, e^{\alpha t}}{1 + \hat{\gamma}^2 e^{2\alpha t}} - \frac{\beta}{\alpha} \\
x_0 &= x + \hat{\gamma} \left(z + \frac{\beta}{\alpha} \right) \frac{1 - e^{2\alpha t}}{1 + \hat{\gamma}^2 e^{2\alpha t}}.
\end{aligned}$$

Note that this expression is not valid at $\theta = \frac{\pi}{2}$; in this case, the characteristic

equations reduce to

$$\frac{dx}{dt} = 0$$
$$\frac{dy}{dt} = \alpha y$$
$$\frac{d\theta}{dt} = 0.$$

To obtain the solution at $\theta = \frac{\pi}{2}$, substitute $x_0 = x$, $\theta_0 = \theta$, and $z_0 = \ [e^{-\alpha t}(\alpha z + \beta) - \beta]$ in (B.15).

Appendix C

Wavefront extraction

In the level set method for geometric acoustics, the physical objects of interest are the acoustic wavefronts, which are embedded in the vector-valued level set function $\mathbf{\Phi}(t, x, z, \theta)$. Extracting the wavefronts for visual display is a computational graphics problem. Recall that the wavefront at a given time t, $W(x, z, t)$ (2.21), is the intersection of the zero level sets of two component level set functions. Standard isosurface routines are available in many software packages, that typically apply some variant of the Marching Cubes algorithm [40]. I am not aware of any standard packages for computing the intersection of the isosurfaces, however. The Visualization Toolkit (VTK) package [57] is available under the Berkeley Software Distribution (BSD) license and includes many powerful graphics routines, though the interface is quite involved. Mayavi [55] integrates with SciPy [36], Python's scientific programming package, to provide a more user-friendly interface for creating 3D graphics. Many VTK commands are accessible from Mayavi through its built-in filters. A Mayavi script was written to produce the wavefront plots displayed in Chapters 4 and 5.

Listings C.1 and C.2 present snippets of the script showing how the Mayavi engine is used to extract and plot the wavefront data. The Python code in Listing C.1 imports the modules needed to execute the script: the data are represented as arrays using the 'numpy' package, which is included in SciPy, and the mlab module is the scripting interface for Mayavi. The next lines access the Mayavi engine, which is used to modify axes and scene properties. The data files are stored in the hdf5 format [28]. The h5py module [10] allows for easy extraction of the data as numpy arrays. Listing C.2 shows the main loop over time, and includes the essential functionality used

to extract the wavefronts. The variable `ptmp` contains the level set function data as a four-dimensional array in (x, z, θ) plus time. The script first extracts the isosurface of the first component of $\mathbf{\Phi}$, then interpolates the second component onto that surface. The remaining code extracts the zero level contour and projects the curve onto physical space, the x-z plane.

Listing C.1: Wavefront extraction script headers

```
import numpy as np
from mayavi import mlab

try:
    engine = mayavi.engine
except NameError:
    from mayavi.api import Engine
    engine = Engine()
    engine.start()
```

Listing C.2: Mayavi script for extracting wavefronts

```
for ii in xrange(len(times)):

    if len(engine.scenes) < (ii+1):
        engine.new_scene()

    scn = engine.scenes[ii]

    # Create a scalar field of the first function:
    f1 = mlab.pipeline.scalar_field(Z, X, TH, ptmp)
    myoutl = mlab.outline(f1)

    # Locate the zero isosurface of the first function:
    iso1 = mlab.pipeline.contour(f1)
    iso1.filter.contours = [0.,]

    # Create a scalar field of the second function:
    f2 = mlab.pipeline.scalar_field(Z, X, TH,ptmp)

    # Extract the vtk data from the isosurface created above:
    ziso, xiso, thiso = iso1.outputs[0].points.to_array().T
    polys = iso1.outputs[0].polys

    # Interpolate the second function over the isosurface:
    f2onsurf = mlab.pipeline.probe_data(f2,ziso,xiso,thiso)

    # Create scalar field from 2nd function on isosurface:
    f2src = mlab.pipeline.scalar_scatter(ziso,xiso,thiso,f2onsurf)
    f2src.data.polys = polys

    # Show the contours
    mlab.pipeline.contour_surface(cln, contours = [0.,])
    mysrf = mymod_mgr.children[0]
    mysrf.contour.number_of_contours = 1
    mysrf.contour.minimum_contour = 0.0

    fig = mlab.gcf()

    # Parallel projection onto physical space (x-z plane)
    mlab.view(azimuth = -90.0, elevation = 0.0, distance = 5.0,
              focalpoint = np.array([0.0,0.5,0.0]), figure = fig)
    scn.scene.parallel_projection = True

    # Display the figure:
    mlab.show()
```

Bibliography

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1972. 10th printing with corrections.

[2] J. Benamou. An introduction to Eulerian geometric optics (1992-2002). *Jour. Sci. Comp.*, 19(1-3):63–93, 2003.

[3] J. Bowlin, J. Spiesberger, T. Duda, and L. Freitag. Ocean acoustical ray-tracing software RAY. Tech. Rept. WHOI-93-10, Woods Hole Oceanographic Institute, October 1992.

[4] J.P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover, New York, second edition, 2001.

[5] L.M. Brekhovskikh and Yu.P. Lysanov. *Fundamentals of Ocean Acoustics*. Springer-Verlag, Secaucus, New Jersey, third edition, 2003.

[6] C. Chauviere, J.S. Hesthaven, and L. Lurati. Computational modeling of uncertainty in time-domain electromagnetics. *SIAM J. Sci. Comput.*, 28:751–775, 2006.

[7] L.-T. Cheng. Efficient level set methods for constructing wavefronts in three spatial dimensions. *J. Comput. Phys.*, 226:2250–2270, 2007.

[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*, 5:347–360, 2004.

[9] 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.*, 208:175–195, 2005.

[10] A. Collette. Hdf5 for python. `http://www.h5py.alfven.org`, 2008.

[11] M.D. Collins. Generalization of the split-step Pade solution. *J. Acoust. Soc. Amer.*, 96:382–385, 1994.

[12] M.D. Collins and W.A. Kuperman. Overcoming ray chaos. *J. Acoust. Soc. Amer.*, 95:3167–3170, 1994.

[13] J. Colosi. Geometric sound propagation through an inhomogeneous and moving ocean: Scattering by small scale internal wave currents. *J. Acoust. Soc. Amer.*, 119:705–708, 2006.

[14] M. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277:1–42, 1983.

[15] T. Duda. Ocean sound channel ray path perturbations from internal-wave shear and strain. *J. Acoust. Soc. Amer.*, 118:2899–2903, 2005.

[16] B. Engquist and O. Runborg. Computational high frequency wave propagation. *Acta Numerica*, pages 181–266, 2003.

[17] B. Engquist, O. Runborg, and A.K. Tornberg. High frequency wave propagation by the segment projection method. *J. Comput. Phys.*, 178:373–390, 2002.

[18] S. Finette. A stochastic response surface formulation of acoustic propagation through an uncertain ocean waveguide environment. *J. Acoust. Soc. Amer.*, 126:2242–2247, 2009.

[19] T.L. Foreman. An exact ray theoretical formulation of the helmholtz equation. *J. Acoust. Soc. Amer.*, 86:234–246, 1989.

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

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

[22] D. Funaro. *Polynomial Approximation of Differential Equations*. Springer-Verlag, Berlin, 1992.

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

[24] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.

[25] O.A. Godin. Restless rays, steady wave fronts. *J. Acoust. Soc. Amer.*, 122:3353–3363, 2007.

[26] G.H. Golub and J.H. Welsch. Calculation of Gauss quadrature rules. *Math. Comp.*, 23:221–230, 1969.

[27] S. Gottlieb and C. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comp.*, 67(221):73–85, 1998.

[28] The HDF Group. Hierarchical data format version 5. `http://www.hdfgroup.org/HDF5/`, 2000-2010.

[29] B. Gustafsson, H.-O. Kreiss, and A. Sundström. Stability theory of difference approximations for mixed initial boundary value problems II. *Mathematics of Computation*, 26(119):649–686, 1972.

[30] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high-order accurate essentially non-oscillatory schemes III. *J. Comput. Phys.*, 71:231–303, 1987.

[31] J.S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral Methods for Time-dependent Problems*. Cambridge University Press, Cambridge, UK, 2007.

[32] F. Jensen, W. Kuperman, M. Porter, and H. Schmidt. *Computational Ocean Acoustics*. Modern Acoustics and Signal Processing. Springer, New York, second edition, 2011.

[33] G. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[34] S. Jin, H. Liu, S. Osher, and R. Tsai. Computing multi-valued physical observables for the high frequency limit of symmetric hyperbolic systems. *J. Comput. Phys.*, 210:497–518, 2005.

[35] S. Jin, H. Liu, S. Osher, and Y.-H. Tsai. Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation. *J. Comput. Phys.*, 205:222–241, 2005.

[36] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python. `http://www.scipy.org/`, 2001–.

[37] J. Keller. Geometrical theory of diffraction. *J. Opt. Soc. Am.*, 52:116–130, 1962.

[38] H. Liu, S. Osher, and R. Tsai. Multi-valued solution and level set methods in computational high frequency wave propagation. *Communications in Computational Physics*, 1(5):765–804, 2006.

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

[40] W. Lorenson and H. Cline. Marching cubes: a high resolution 3D surface reconstruction algorithm. *Comp. Graphics*, 21(4):163–169, 1987.

[41] S. Martinelli. Numerical boundary conditions for specular reflection in a level-sets-based wavefront propagation method. Submitted (2012).

[42] S. Martinelli. An application of the level set method to underwater acoustic propagation. *Commun. Comput. Phys.*, 2012. To appear.

[43] A.E. Newhall, T.F. Duda, et al. Acoustic and oceanographic observations and configuration information for the WHOI moorings from the SW06 experiment. Tech. Rept. WHOI-2007-04, Woods Hole Oceanographic Institute, May 2007.

[44] F. Nobile, R. Tempone, and C. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46:2309–2345, 2008.

[45] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constr. Approx.*, 15:499–522, 1999.

[46] J.A. Ogilvy. Wave scattering from rough surfaces. *Rep. Prog. Phys.*, 50(4):1553–1608, 1987.

[47] 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.*, 179:622–648, 2002.

[48] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, New York, 2003.

[49] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

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

[51] M.B. Porter and H.P. Bucker. Gaussian beam tracing for computing ocean acoustic fields. *J. Acoust. Soc. Amer.*, 82:1349–1359, 1987.

[52] J. Qian, L.-T. Cheng, and S. Osher. A level set-based Eulerian approach for anisotropic wave propagation. *Wave Motion*, 37(4):365–379, 2003.

[53] J. Qian and S. Leung. A level set based Eulerian method for paraxial multivalued traveltimes. *J. Comput. Phys.*, 197:711–736, 2004.

[54] J. Qian and S. Leung. A local level set method for paraxial geometrical optics. *SIAM J. Sci. Comput.*, 28(1):206–223, 2006.

[55] P. Ramachandran and G. Varoquaux. Mayavi: 3D Visualization of Scientific Data. *Computing in Science & Engineering*, 13(2):40–51, 2011.

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

[57] W. Schroeder, K. Martin, and W. Lorenson. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, Inc., Clifton Park, New York, fourth edition, 2006.

[58] J. Shen and L.-L. Wang. Some recent advances on spectral methods for unbounded domains. *Commun. Comput. Phys.*, 5:195–241, 2009.

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

[60] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[61] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II. *J. Comput. Phys.*, 83:32–78, 1989.

[62] A.C. Singer, J.K. Nelson, and S.S. Kozat. Signal processing for underwater acoustic communications. *Communications Magazine, IEEE*, 47(1):90 –96, January 2009.

[63] K.B. Smith, M.G. Brown, and F.D. Tappert. Ray chaos in underwater acoustics. *J. Acoust. Soc. Amer.*, 91:1939–1949, 1992.

[64] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4:240–243, 1963.

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

[66] L. Trefethen. Stability of finite-difference models containing two boundaries or interfaces. *Mathematics of Computation*, 45(172):279–300, 1985.

[67] H.L. Van Trees. *Detection, Estimation, and Modulation Theory: Part I.* Wiley-Interscience, New York, paperback edition, 2001.

[68] H. Weinberg and R.E. Keenan. Gaussian ray bundles for modeling high-frequency propagation loss under shallow-water conditions. *J. Acoust. Soc. Amer.*, 100:1421–1431, 1996.

[69] H. Widom. Asymptotic behavior of the eigenvalues of certain integral equations II. *Arch. Ration. Mech. An.*, 17:215–229, 1964.

[70] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, Princeton, New Jersey, 2010.

[71] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27:1118–1139, 2005.