RICE UNIVERSITY

Discontinuous Galerkin Time Domain Methods for Acoustics and Comparison with Finite Difference Time Domain Methods

by

Xin Wang

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Arts

Approved, Thesis Committee:

Dr. William W. Symes, Chairman
Noah G. Harding Professor of Computational and Applied Mathematics

---

Dr. Tim Warburton
Associate Professor of Computational and Applied Mathematics

Dr. Béatrice M. Rivière
Associate Professor of Computational and Applied Mathematics

Houston, Texas

December 2009

Abstract

Discontinuous Galerkin Time Domain Methods for Acoustics and Comparison with Finite Difference Time Domain Methods

by

Xin Wang

This thesis describes an implementation of the discontinuous Galerkin finite element time domain (DGTD) method on unstructured meshes to solve acoustic wave equations in heterogeneous media. In oil industry people use finite difference time domain (FDTD) methods to simulate seismic surveys, the first step to explore oil and gas in the earth's subsurface, conducted either in land or sea. The results in this thesis indicate that the first order time shift effect resulting from misalignment between numerical meshes and material interfaces in the DGTD method occurs in the same way as interface errors in the finite difference simulation of wave propagation. This thesis describes two approaches: interface-fitting mesh and local mesh refinement, without modifying the DGTD scheme, to decrease this troublesome effect with verifications of 2D examples. The comparison of the DGTD method on the piecewise linear interface-fitting mesh and the staggered FDTD method both applied to a square-circle model and a 2D dome model in this thesis confirms the fact that the DGTD method can

achieve a suboptimal second order convergence rate while the error in the staggered FDTD method is dominated by the first order interface error when the curved material interfaces are presented. I conclude that the DGTD method is more efficient than the staggered FDTD method for the two solutions to have roughly the same accuracy when the accuracy requirement becomes more and more strict and the model becomes more and more complex.

Acknowledgements

I would like to thank my advisor Prof. William Symes for his guidance and advice. His passion and enthusiasm for applied math and scientific computing gives me a lot of inspirations. I would like to express my special gratitude to Prof. Tim Warburton who provided all kinds of helps I needed to complete this thesis. I would also like to thank Prof. Béatrice M. Rivière who reviewed my thesis very carefully and gave me helpful guidance to improve it. Last, but not least, I would like to thank Dr. Dan Sorensen and Dr. Jan Hewitt along with Dr. Symes, who through the Thesis Writing class have greatly helped both my writing and presentation skills. As a foreign student, I am really grateful for their helps from which I could benefit in a long run.

This work was partially supported by The Rice Inversion Project (TRIP) and NSF grant DMS-0810187. I greatly appreciate their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The numerical solutions of wave propagation have numerous applications. One of particular interest and importance is to simulate seismic surveys in oil and gas exploration. Various numerical methods have been developed to carry out this simulation. In this thesis I describe an implementation of the discontinuous Galerkin finite element time domain method (DGTD) applied to acoustic wave equations in heterogeneous media and use computer programs I developed to evaluate its accuracy and efficiency. Furthermore, I will make comparison of DGTD and finite difference time domain method (FDTD), a fully-developed method used daily as the wave propagator in reflection seismology for decades. This comparison may provide a guideline of choosing the proper method for the proper problem.

1.1 MOTIVATION

The first step in oil and gas exploration either in land or sea is to conduct seismic surveys, which typically consist in sending into the ground sound waves generated by

sources at the surface such as air guns in marine surveys or dynamite in land acquisition and through sensors called geophones recording echoes of the sending waves, caused by the heterogeneity of the earth's subsurface. Seismologists and geoscientists then can analyze the recorded time series of data called seismic traces or seismograms and interpret the earth's interior properties by imaging technologies based on the basic mathematical point of view that "waves transfer space-time resolved information from one place to another with (relatively) little loss" (Symes, 2003).

Numerical seismic modeling, duplicating the seismic survey procedure and generating synthetic seismograms provided the earth's subsurface structure is known, has applications at least in two aspects including the inversion process and the experimental design. Reflection seismic inversion problem is usually set to find the geological model given observed seismic data. For example, in linearized inversion problems the oscillatory part of the geological model is updated iteratively by minimizing an objective function, for example, a distance function related to observed data (seismograms) and synthetic data. Since each iteration a wave propagation problem with highly intensive computation must be solved and thousands of iterations are required, an efficient and accurate wave propagation solver is highly demanded in this context.

In addition, numerical modeling has been used for designs of acquisition geometries. Source and receiver geometries decide the subsurface image quality in the later process after seismic surveys. Regone (2007) used the 3D finite difference modeling, instead of unfeasible 3D field surveys due to the cost, to convince people the great improvement in image quality with wide-azimuth towed-streamer surveys over traditional narrow-azimuth towed-streamer surveys, and came up with two newly acquisition geometries: receivers distributed in a sparse grid on the ocean floor with sources distributed in a dense grid on the surface (OBS) and multiple vessels wide-azimuth

towed-streamer surveys (WATS), which both have been used in practice.

Various numerical methods are applicable to wave propagation problems. Finite difference (FD) and discontinuous Galerkin (DG) methods are of this thesis's concerns. FD methods become an industry standard for solving wave propagation problems due to their desirable trade-off between the computation efficiency and accuracy, as well as the relatively easy implementation. DG methods draw a lot of attentions in computational electromagnetic and fluid dynamics communities recently since DG methods specialize in solving hyperbolic partial differential equation (PDE) and dealing with complex geometries. Because (a) seismic wave equations are one class of the most basic linear hyperbolic PDEs, (b) the heterogeneity of the geological model and the irregular surface landform in the land acquisition are in need of discrete grids having the flexibility to represent complex structures precisely, DG methods are a competitive candidate for the seismic wave simulation.

Though a lot of efforts have been paid on both methods respectively, there are few works regarding careful comparison of them. Sjögren (2002) investigated the performance of spectral element and finite difference methods for wave propagation problems with a simple straight-line interface. My goal of this work is to carefully compare the DGTD and FDTD methods within the context of reflection seismology on more complex models, and to come up with some useful conclusions which may provide a guideline of choosing the proper method for the proper problem.

In this work I choose the acoustic wave equations (pressure-velocity formulation) to which the two methods are applied. The acoustic wave equations form a linear PDE system in which the acoustic pressure and particle velocity interact with one another, according to Hooke's law and conservation of momentum, to propagate acoustic waves through materials. Although the elastic wave simulation is much closer to the way

waves propagate in the subsurface, the acoustic modeling is good enough for this study for several reasons.

First, material parameters are easier to prepare for the acoustic modeling. Material density and bulk modulus, or sound speed, are all needed for the acoustic modeling, while in the elastic modeling the stress-strain relation itself can asks for up to 21 independent material parameters, some of which are still mysterious. Even in the isotropic case, how to interpret the shear velocity near the ocean bottom is still a mystery to geoscientists. Second, one can obtain analytic solutions of acoustic wave equations in some cases. Therefore I can carry out conventional convergence tests rather than the ones through error estimations. Third, for free surface boundaries, absorbing boundaries, the ocean floor nearby and so on, the two methods in the elastic modeling have their own delicate numerical treatments, which may render the comparison less objective. Acoustic wave equations as a simplified model don't have these tedious and tricky implementation issues. The comparison on the acoustic modeling can reveal underlying intrinsic natures of the two methods themselves as much as possible. Last but not least, the acoustic modeling has been used as grounds for many of processing technologies. In future work, I may use the acoustic modeling based on DGTD methods as a built-in part of imaging process and evaluate them at the next level of applications in reflection seismology.

1.2 REVIEW

Characterizations of convection phenomena are illustrated through natural and artificial events from the hurricane formation to the oil and gas production. A class of hyperbolic PDEs under certain mathematical assumptions describe the evolution of these phenomena. Toro (1997) from both theoretical and numerical aspects gave

a review of hyperbolic PDEs originating from fluid dynamics. Mathematical derivations of both acoustic and elastic wave equations of this thesis's interest can be seen in Leveque (2002), which also includes theories and numerical methods (finite volume methods) for general hyperbolic problems. Bedford and Drumheller (1994) offered a basic introduction to the linear elasticity for wide audience. The first part of Cohen (2002) also provides a good reference to understand basic definitions and properties of wave equations.

DG methods have been applied to a wide range of hyperbolic problems, including gas and fluid dynamics (Cockburn and Shu, 1989; Bassi and Rebay, 1997; Giraldo et al., 2002), Maxwell's equations (Warburton, 1999; Hesthaven and Warburton, 2002; Cockburn et al., 2004; Cohen et al., 2006), acoustic and elastic wave equations (Atkins and Shu, 1998; Käser and Dumbser, 2006) and so on for more than three decades, since the first DG scheme was introduced by Reed and Hill (1973) for solving a neutron transport equation. DG methods can be viewed as an extension of finite volume (FV) methods in the sense that DG methods allow discontinuous approximation of solutions. Techniques developed for FV methods fit into DG methods naturally (Cockburn et al., 1999). For example, numerical fluxes, through which underlying physical laws could be imposed into the numerical scheme so as to ensure the numerical stability, have been studied for a while by authors (Toro, 1997; Leveque, 2002), whose results can be directly used in the DG framework. Hesthaven and Warburton (2008) scrutinized DG methods together with other popular numerical methods for PDEs and outline the general properties of DG methods and conclude that DG methods possess every useful feature discussed. Cockburn et al. (1999) and Hesthaven and Warburton (2008) listed generic properties of DG methods as follows,

- DG methods are good at dealing with complex geometries and usually form

easy implementations for boundary conditions.

- For time dependent PDEs, DG methods can formulate an explicit semi-discrete form due to the locally defined mass matrix. This avoids inverting a large-scale mass matrix such as in conventional finite element methods.

- High order accuracy under the DG framework is achievable for problems whose exact solutions are smooth. This is a good news for this work because solutions of wave equations usually have certain smoothness property.

- Implementations of hp-adaptivity in DG methods are less troublesome than conforming finite element methods, because no continuity restrictions are imposed on numerical solutions.

The convergence analysis of DG methods can go back to Lesaint and Raviart (1974). By assuming analytic solutions are smooth, they proved $h^{p+\epsilon}$ convergence rate, where h is grid size, p is the degree of local basis polynomials and the value ϵ is determined by ways of the triangulation, for example, 0 for general grids and 1 for Cartesian grids. Johnson and Pitkaranta (1986) later ameliorated ϵ to be $1/2$ for general grids. Ainsworth (2004) studied the dispersive behavior of DG methods and showed an exponential decay of the relative phase error provided $2p + 1 \approx ckh$ for some fixed constant $c > 1$, where k is the spatial wave-number. This result is critical because it states a rule for choosing the order of basis functions with respect to the grid size so as to achieve the prescribed accuracy in wave simulations with DG methods. A similar criterion for the finite-difference acoustic modeling can be found in Alford et al. (1974).

For time dependent problems, a proper time integration method, i.e., an ODE solver, is required after the spatial discretization with DG methods. To avoid solving

large-scale linear systems each time step, an explicit ODE solver is preferred. Chavent and Salzano (1982) formed a DG method for the spatial discretization in 1D scalar conservation law and used a forward Euler method for the time integration. This method by a numerical analysis, however, is shown to be stable only if $\Delta t \sim \mathcal{O}(\Delta x^{3/2})$, which implies unacceptably small time step. High-order accurate Runge-Kutta (RK) DG methods, generalized by Cockburn and Shu (1989), are widely used because they match the temporal and spatial discretization accuracy, and just require $\Delta t \sim \mathcal{O}(\Delta x)$. The ADER-DG methods in Käser and Dumbser (2006) are also an interesting way to achieve high order accuracy in both space and time by using arbitrary high order derivatives to construct the time discretization. This work uses the low-storage five-stage fourth-order explicit RK method for the time discretization in the DG implementation as discussed in Hesthaven and Warburton (2008).

Finite difference (FD) methods are a robust tool and an industry standard for seismic wave simulation, because FD methods perform a good balance between the accuracy and efficiency and are easily implemented. Owing to efforts of many authors (Richtmyer and Morton, 1967; Mitchell and Griffiths, 1994; Cohen, 2002), basic issues related to FD methods, such as consistency, stability and convergence, are well studied and criteria for designing FD methods for different problems are provided. One can expect reliable solutions from a suitable FD method.

In FD methods, space and time are discretized by grids, where field variables as well as material parameters are defined. Differential operators are approximated by finite-difference formulas. In seismic wave simulation, discrete grids are usually distributed equally on a Cartesian coordinate system, though other systems are possible. The benefits of using such grids are explicit grid-neighbor relation and efficient memory access pattern.

There are two approaches of FD methods, the conventional-grid approach and the staggered-grid approach differentiated with each other in the way whether unknowns of field variables and/or material parameters are defined on the same grid or not. In the 1960s and 1970s, the conventional-grid approach was popular for the displacement formulation of elastic wave equations, of which functions are approximated at the same discrete grid (Alford et al., 1974; Alterman and Karal, 1968; Boore, 1970, 1972; Dablain, 1986; Kelly et al., 1976). Unfortunately, grid dispersion and numerical instabilities when material parameters have high contrast discontinuities influence numerical solutions badly in this approach. The staggered-grid approach, in which several grids are employed, overcomes these difficulties and so is widely used. Yee (1966) first applied a staggered-grid second order FD scheme to Maxwell's equations in an isotropic medium. Madariaga (1976) later used a second order staggered-grid FD scheme to model the earthquake rupture. Virieux (1984, 1986) formally established the staggered-grid approach to solve velocity stress formulation of elastic wave equations. The fourth-order staggered-grid FD method introduced by Bayliss et al. (1986) was then proved to be more efficient and requires less memory than the second-order method, since the grid interval in the fourth-order method can be twice as much as the one in the second-order method according to the rule of thumb proposed by Alford et al. (1974).

High-order FD time domain methods are constructed to control the grid dispersion effectively for models with smooth parameters. However, FD methods for seismic models with high contrast discontinuities of material parameters lead to a first-order interface error, which is irrelevant to FD methods, but stems from the insufficient representation of model parameters on discrete grids. Brown (1984) first analyzed this first order error component of FD methods for an interface problem. Symes

and Vdovina (2009) theoretically and numerically quantified the first order interface misalignment error for the second order in time and space staggered-grid FD method applied to the pressure-velocity formulation of the acoustic wave equations, and provided an explicit expression of a non-zero time shift of numerical solutions due to the first order interface error. In their discussion, the interface error of staggered-grid FD methods is unavoidable for models of heterogeneous media, because several grids are employed in staggered-grid FD methods and the misalignment with the material interface must occur for at least one grid.

1.3 CLAIM

This thesis through numerical examples demonstrates DG methods somehow can remedy the interface error by using piecewise linear interface-fitting meshes and the local mesh refinement technique. Given a mesh of elements with straight-line edges (triangles for example), each element can be considered as a homogeneous medium. In this way, one actually glues piecewise constant medium in each element together to approximate the real model. Interface-fitting meshes align the vertices of elements with material interfaces. Provided the seismic model can be detached into regions of different homogeneous media and an interface-fitting mesh for this model is generated, (a) when interfaces are composed of line segments, the model approximation on this mesh is exact, so the numerical error only comes from DG methods; (b) when interfaces are curved and this mesh is not too coarse, the model approximation brings in a second order error since line segments of elements can not fully represent the curved interfaces, therefore the second order convergence rate is the best one can expect in this case. In short, by using interface-fitting meshes, the second order convergence rate is achievable with DG methods on interface-fitting meshes. However this kind

of meshes sometimes are hard to generate especially when the material structure is complicated.

The local mesh refinement technique is another approach proposed in this thesis to decrease the interface error. By locally refining the mesh near the interface, we reduce the interface error as the distance between the interface and the nearest grid points is shortened. The drawback of this technique is that one can get very small time step due to the small elements by the refinement.

I compare the computation cost, measured by the wall clock time or the number of float point operations, for the two solutions by FD and DG methods to have roughly the same accuracy. For the model composed of piece-wise constant media, DG methods on an interface-fitting mesh can achieve the second order convergence rate while the numerical error of staggered-grid FD methods is dominated by the first order interface error. In this case, DG methods are more efficient for a prescribed accuracy (5% for example). This conjecture is demonstrated by numerical examples.

I also implement the low-storage curvilinear DG method. In this method curvilinear elements instead of straight sided elements are used to approximate the geometry of the model, such as the material interfaces or boundaries. The geometry representation by curvilinear elements complements the accuracy of the DG solver. We can then expect the optimal convergence rate for the numerical simulation. Compared to the 2nd order convergence rate on straight sided elements, this method is more tempting. In order to form the curvilinear elements, one need the geometry information as precisely as possible. In practice the exact expression of the geometry for a realistic model is usually unknown. Therefore it is impossible to build curvilinear elements and the curvilinear DG method is not applicable.

1.4 AGENDA

This thesis is organized as follows. First the model problem, i.e., pressure-velocity formulation of the acoustic wave equations, is introduced. For the model problem, analytic solutions of three special cases are discussed. The first two cases are used in the convergence tests later. The last one, called the Riemann problem, showing a way to construct the numerical flux for the linear PDE system, is recalled when the DG method is formulated. Then I construct the DG method following Hesthaven and Warburton (2008). Later, two types of boundary conditions widely used in seismic simulation are discussed. At the end of the method chapter, I present the 2-4 staggered-grid FDTD method for acoustics used for the comparison.

Several numerical experiments are carried out for the convergence tests of DGTD methods and comparison of DGTD and FDTD. I first present the point source wave and plane wave experiments for the purpose of the convergence tests. Then the interface error in DGTD methods is illustrated by using the misaligned mesh, while the interface-fitting mesh and the local refined mesh examples show the approaches at the mesh level in DGTD methods to reduce this unpleasant error component. At last I make comparison of the two methods with respect to the computation cost and the numerical accuracy. The square-circle model and the 2D dome model are used for such comparison.

After numerical experiments, I will discuss the formulation of the low-storage curvilinear DG method. This method is then applied to the square-circle model as well as the 2D dome model. As seen in numerical experiments the curvilinear DG method achieves the optimal convergence rate.

Chapter 2

Methods

2.1 INTRODUCTION

This chapter first introduces the pressure-velocity formulation of the acoustic wave equations, for which three special cases are discussed. Analytic solutions of the first two cases are provided for the convergence tests. The last case forms a Riemann problem. The process for the construction of its solution shows a way to formulate the numerical flux for the acoustic wave equations (AWEs) in DG methods. Next, the DG discretization for AWEs in space is built step by step following Hesthaven and Warburton (2008). I omit basic theoretical discussions, but focus on presenting a self-sustained construction procedure, with which one can develop and implement a DG method by himself/herself. Finally, Richardson extrapolation is discussed for the convergence rate estimation in the case where analytic solutions are inaccessible.

2.2 MODEL PROBLEM

The AWEs are expressed as a linear PDE system, in which the acoustic pressure and particle velocity interact with one another to propagate waves through materials. The purely hyperbolic property makes such system a good prototype to understand the behaviors of the hyperbolic PDEs and develop efficient numerical methods.

This first-order linear system in terms of the pressure p and the velocity $\mathbf{v}$ reads,

$$\begin{aligned} \rho(\mathbf{x})\frac{\partial \mathbf{v}}{\partial t} + \nabla p &= 0, \\ \frac{1}{\kappa(\mathbf{x})}\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{v} &= f(\mathbf{x}, t; \mathbf{x}_s), \end{aligned} \tag{2.1}$$

where f represents the source density function with respect to the source location $\mathbf{x}_s \in \mathbb{R}^n$ $(n = 1, 2, or\ 3)$ as well as space $\mathbf{x} \in \mathbb{R}^n$ and time t; ρ and κ denote the mass density and bulk modulus, respectively, which only depend on $\mathbf{x}$; the wave speed is $c(\mathbf{x}) = \sqrt{\dfrac{\kappa(\mathbf{x})}{\rho(\mathbf{x})}}$ and the acoustic impedance is $Z(\mathbf{x}) = \sqrt{\kappa(\mathbf{x})\rho(\mathbf{x})}$.

Point source wave

Ideally, the source density function f is viewed as an isotropic point radiator with a known time-dependent function as,

$$f(\mathbf{x}, t; \mathbf{x}_s) = w(t)\delta(\mathbf{x} - \mathbf{x}_s), \tag{2.2}$$

which is a quite coarse approximation to the real source function. Usually, the source pulse $w(t)$ is of compact support in time, that is $w(t) = 0, |t| > t_0$. This mimics the acoustic energy generated by airguns or dynamite during a limited time. To see how

waves expand starting from a point, I hereby assume the material is homogeneous, that is $\rho(\mathbf{x}) = \rho_0$ and $\kappa(\mathbf{x}) = \kappa_0$, $\forall \mathbf{x}$. The model problem Eqs.(2.1) are then equivalent to the second order wave equation for the acoustic potential u with proper initial and boundary conditions,

$$\frac{1}{\kappa_0}\frac{\partial^2 u}{\partial t^2} - \nabla \cdot \frac{1}{\rho_0}\nabla u = w(t)\delta(\mathbf{x} - \mathbf{x}_s), \tag{2.3}$$

where $u(\mathbf{x}, t) = \int_{-\infty}^{t} p(\mathbf{x}, s)ds$, thereby $p = \dfrac{\partial u}{\partial t}$ and $\mathbf{v} = -\dfrac{1}{\rho_0}\nabla u$. The solution of Eq.(2.3) in 3D with vanishing initial conditions describes outgoing spherical waves at speed c_0 as

$$u(\mathbf{x}, t) = \rho_0 \frac{w(t - r/c_0)}{4\pi r}, \quad r = \|\mathbf{x} - \mathbf{x}_s\|. \tag{2.4}$$

Two layer media in 1D

The material parameters ρ and κ are usually non-constant, and vary with space. Due to the material discontinuity, or called the material interface, the incident wave is decomposed into a transmitted wave and a reflected wave. A two-layer medium is a good simplified model to replicate this wave decomposition procedure. In two layer media, the material parameters have a jump at $x = 0$,

$$(\kappa(x), \rho(x)) = \begin{cases} (\kappa_l, \rho_l), & x \leq 0 \\ (\kappa_r, \rho_r), & x > 0. \end{cases} \tag{2.5}$$

Eqs.(2.1) without the source term can be further simplified into 1D case, because of field variables only dependent of x,

$$\begin{aligned} \rho(x)\frac{\partial v(x,t)}{\partial t} &= -\frac{\partial p(x,t)}{\partial x}, \\ \frac{1}{\kappa(x)}\frac{\partial p(x,t)}{\partial t} &= -\frac{\partial v(x,t)}{\partial x}. \end{aligned} \tag{2.6}$$

Assuming the incident wave impacts on the interface from the left ($x < 0$), the solution of Eqs.(2.6) with the continuity restriction at the interface $x = 0$ is then given by,

$$\begin{aligned} p(x,t) &= \begin{cases} g(t - x/c_l) - \dfrac{\rho_l c_l - \rho_r c_r}{\rho_l c_l + \rho_r c_r} g(t + x/c_l), & x \le 0 \\ \dfrac{2\rho_r c_r}{\rho_l c_l + \rho_r c_r} g(t - x/c_r), & x > 0 \end{cases} \\ v(x,t) &= \begin{cases} \dfrac{1}{\rho_l c_l}\left(g(t - x/c_l) + \dfrac{\rho_l c_l - \rho_r c_r}{\rho_l c_l + \rho_r c_r} g(t + x/c_l)\right), & x \le 0 \\ \dfrac{2}{\rho_l c_l + \rho_r c_r} g(t - x/c_r), & x > 0 \end{cases} \end{aligned} \tag{2.7}$$

where g is a continuous function compatible with the initial condition.

A two-layer medium model as well as its solution set up an interesting and practicable example to study the error components in numerical methods, see Symes and Vdovina (2009). I will use the same example to illustrate the error components in DG methods later and propose approaches to suppress the interface error.

Riemann problem

Previously I have studied the cases with continuous solutions due to the continuous initial conditions. Now I consider the problem with piece-wise constant initial data, called the Riemann problem. I start with Eqs.(2.6) in a constant medium

and construct its solution. Then I extend to the model of heterogeneous media and multidimensional cases. The discussion below is partly adopted from Leveque (2002).

Eqs.(2.6) in a homogeneous medium with piece-wise constant initial data can be rewritten into matrix-vector form as,

$$\frac{\partial \mathbf{q}}{\partial t} + A(x)\frac{\partial \mathbf{q}}{\partial x} = 0, \tag{2.8}$$
$$\mathbf{q}(x,0) = \begin{cases} \mathbf{q}_L, & x < 0 \\ \mathbf{q}_R, & x > 0 \end{cases}$$

with

$$\mathbf{q} = \begin{bmatrix} v(x,t) \\ p(x,t) \end{bmatrix}, \quad \mathbf{q}_L = \begin{bmatrix} v_L \\ p_L \end{bmatrix}, \quad \mathbf{q}_R = \begin{bmatrix} v_R \\ p_R \end{bmatrix}, \quad A(x) = \begin{bmatrix} 0 & \kappa_0 \\ 1/\rho_0 & 0 \end{bmatrix},$$

where A is a constant matrix and diagonalizable, that is $A = R\Lambda R^{-1}$ with

$$R = [r_1 \ r_2] = \begin{bmatrix} -\sqrt{\rho_0\kappa_0} & \sqrt{\rho_0\kappa_0} \\ 1 & 1 \end{bmatrix}, \Lambda = \begin{bmatrix} -c_0 & 0 \\ 0 & c_0 \end{bmatrix}. \tag{2.9}$$

By introducing characteristic variables $W = R^{-1}Q = [w_1, w_2]^T$ and multiplying Eq.(2.8) byR^{-1}, it is then deduced into two decoupled advection equations,

$$\begin{aligned} &\frac{\partial w_1}{\partial t} - c_0\frac{\partial w_1}{\partial x} = 0, \\ &\frac{\partial w_2}{\partial t} + c_0\frac{\partial w_2}{\partial x} = 0, \end{aligned} \tag{2.10}$$

with piece-wise constant initial conditions,

$$W(x,0) = \begin{cases} R^{-1}\mathbf{q}_L := [w_{1,l}, w_{2,l}]^T, & x < 0 \\ R^{-1}\mathbf{q}_R := [w_{1,r}, w_{2,r}]^T, & x > 0 \end{cases}$$

whose solutions are

$$w_1(x,t) = w_1(x + c_0 t, 0) = \begin{cases} w_{1,l}, & x + c_0 t < 0 \\ w_{1,r}, & x + c_0 t > 0 \end{cases}$$

$$w_2(x,t) = w_2(x - c_0 t, 0) = \begin{cases} w_{2,l}, & x - c_0 t < 0 \\ w_{2,r}, & x - c_0 t > 0 \end{cases}$$

Finally, I can recovery the solution of Eq.(2.8) as a linear combination of the right eigenvectors r_1, r_2 of A.,

$$\begin{aligned} \mathbf{q}(x,t) = RW(x,t) &= r_1 w_1(x,t) + r_2 w_2(x,t) \\ &= \begin{cases} r_1 w_{1,l} + r_2 w_{2,l}, & x < -c_0 t \\ r_1 w_{1,r} + r_2 w_{2,l}, & -c_0 t < x < c_0 t \\ r_1 w_{1,r} + r_2 w_{2,r}, & x > c_0 t \end{cases} \end{aligned} \tag{2.11}$$

The two characteristic lines $x = \pm c_0 t$ separate the upper-half $x-t$ plane into three regions. In each region, the solution $\mathbf{q}(x,t)$ is constant.

One thing to notice is that the above derivation is valid not only for a constant medium, but also for cases where $R^{-1}\dfrac{\partial \mathbf{q}}{\partial x} = \dfrac{\partial (R^{-1}\mathbf{q})}{\partial x}$, i.e., R is independent of x. This is true when the left and right acoustic impedances equal to each other, i.e., $Z_l = Z_r$ for two layer media. The solution in this case has the same structure as Eqs.(2.11), but the left and right wave speeds are different.

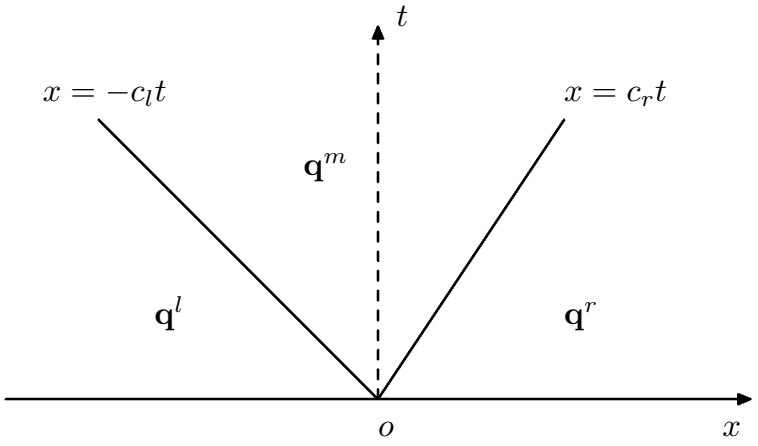

Figure 2.1: The solution for the Riemann problem in two layer media in $x-t$ plane has three states separated by the two characteristic lines $x = -c_l t$ and $x = c_r t$.

If $R(x)$ depends on x then $R^{-1}(x)\dfrac{\partial \mathbf{q}(x,t)}{\partial x} = \dfrac{\partial (R^{-1}(x)\mathbf{q}(x,t))}{\partial x} - \dfrac{dR^{-1}(x)}{dx}\mathbf{q}(x,t)$. The extra term $\dfrac{dR^{-1}(x)}{dx}\mathbf{q}(x,t)$ as a source term ruins the independence of the left-going and right-going waves and couples them together. Thus the solutions are complicated compared to the constant impedance case.

Leveque (2002) suggests to solve a general Riemann problem at the interface between two different materials. As indicated in Fig.(2.1), the two characteristic lines $x = -c_l t$ and $x = c_r t$ cut the upper half $x-t$ plane into three regions. From Eqs.(2.11), it is seen that the solution for the Riemann problem is a linear combination of the two waves represented by the two right eigenvectors of the material matrix A. Since the waves propagate the initial data to the left region ($x < -c_l t$) and the right region ($x > c_r t$), $\mathbf{q}^l(x,t) = \mathbf{q}_L$ and $\mathbf{q}^r(x,t) = \mathbf{q}_R$. The intermediate state $\mathbf{q}^m(x,t)$ fills in the middle region ($-c_l t < x < c_r t$) left by the left-going and right-going waves. We also notice that the solution jump across any of the two characteristic lines is parallel

with one right eigenvector of coefficient matrix $A(x)$. Using this feature here gives,

$$\mathbf{q}^m - \mathbf{q}^l = \alpha_1 \begin{bmatrix} -Z_l \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{q}^r - \mathbf{q}^m = \alpha_2 \begin{bmatrix} Z_r \\ 1 \end{bmatrix},$$

Combining the two equations yields

$$\mathbf{q}^r - \mathbf{q}^l = \alpha_1 \begin{bmatrix} -Z_l \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} Z_r \\ 1 \end{bmatrix} = R_{lr} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

Thus one can obtain $\mathbf{q}^m(x,t) = [v^m(x,t)\ p^m(x,t)]^T$, after solving α_1, α_2 in this linear system. When $-c_l t < x < c_r t$,

$$\begin{aligned} v^m(x,t) &= \frac{Z_l v_L + Z_r v_R}{Z_l + Z_r} - \frac{1}{Z_l + Z_r}(p_R - p_L), \\ p^m(x,t) &= \frac{Z_r p_L + Z_l p_R}{Z_l + Z_r} - \frac{Z_l Z_r}{Z_l + Z_r}(v_R - v_L). \end{aligned}$$

Here the left-going and right-going waves are coupled together in the intermediate state $\mathbf{q}^m(x,t)$ through R_{lr}, which is composed of the left-going eigenvector from the left medium and the right-going eigenvector from the right medium.

In the two or three dimensions, a two-layer medium is too simple to describe the structure of the real model. However, two-layer medium structures are still a good approximation locally for a small piece of model where two different materials are presented and the interface is almost flat. The solution in 1D two layer media may extend to 2D or 3D over such a piece of model in a short period of travel time. In a long period of time, the extension does not work because the waves propagating from other places interfere with the solution in an unexpected way.

As seen in Fig.(2.2), the material parameters vary along the direction $\vec{n}$ $(= (n_x, n_z))$ at the interface, but keep constant along the tangent direction $\vec{\tau}$. After rotating the $x-z$ Cartesian system to $\xi-\eta$ coordinate system whose axes are along $\vec{n}$ and $\vec{\tau}$, the 2D Riemann problem can be decoupled into two 1D Riemann problems with respect to ξ and η. The 1D Riemann problem with respect to τ is trivial since material parameters and initial data are all homogeneous (constant) along the tangent direction. Along the normal direction, the acoustic wave equations in term of the pressure p and normal velocity $v_{\vec{n}}(= n_x v_x + n_z v_z$, where v_x, v_z are velocity components) with piecewise constant initial data (Riemann problem) read,

$$\rho(\xi)\frac{\partial v_{\vec{n}}}{\partial t} + \frac{\partial p}{\partial \xi} = 0, \quad \frac{1}{\kappa(\xi)}\frac{\partial p}{\partial t} + \frac{\partial v_{\vec{n}}}{\partial \xi} = 0, \tag{2.12}$$

$$(v_{\vec{n}}(\xi,0), p(\xi,0)) = \begin{cases} (v^1_{\vec{n}}, p^1), & \xi < 0 \\ (v^2_{\vec{n}}, p^2), & \xi > 0 \end{cases}$$

$$(\kappa(\xi), \rho(\xi)) = \begin{cases} (\kappa_1, \rho_1), & \xi < 0 \\ (\kappa_2, \rho_2), & \xi > 0 \end{cases}$$

where I use the fact that the field variables are independent of η

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} = n_x \frac{\partial}{\partial \xi},$$
$$\frac{\partial}{\partial z} = \frac{\partial \xi}{\partial z}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial z}\frac{\partial}{\partial \eta} = n_z \frac{\partial}{\partial \xi},$$

because along the direction $\vec{\tau}$ field variables keep constant.

In three dimensions, the Riemann problem in two layer media for acoustics is decoupled into three 1D Riemann problems along the normal direction and two tangent directions of the interface plane. The only nontrivial Riemann problem again is the

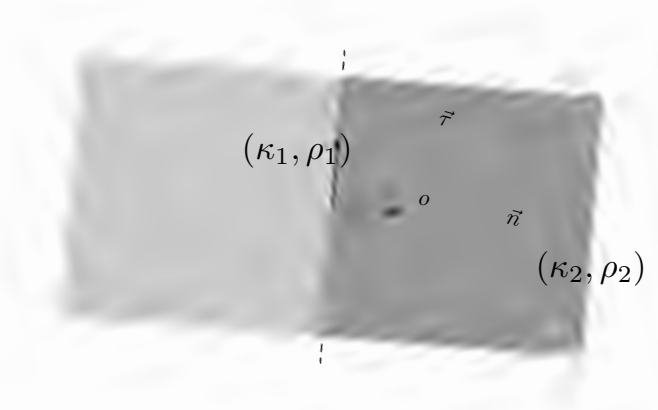

Figure 2.2: An illustration of a two-layer medium structure in 2D. $\vec{n}$,$\vec{\tau}$ are the unit normal vector and the unit tangent vector of the interface (dashed line), respectively. Here o is the origin.

one along the normal direction, which is the same as Eqs.(2.12).

The Riemann solver discussed here is intended to construct numerical fluxes in DG methods, rather than to derive analytic solutions. As mentioned before, two copies of degrees of freedom are defined at the same spatial point on an element edge in DG methods, since the DG solution has no continuity requirement. A reasonable solution at those points in a period of time combining information from both elements sharing the same edge is needed to advance the numerical solution. A Riemann problem in two layer media fits for this setting ideally. For each point on an element edge, a small tube centered at this point and orthogonal to the edge is considered as the two-layer medium model. The initial data are the numerical solutions at this common point from both elements. Since this point as time goes by stay in the middle region of the upper half $x-t$ plane as shown in Fig.(2.1), the intermediate state solution is the very ingredient to formulate the numerical flux.

For other hyperbolic PDEs, especially when non-linearity presents, it is sometimes impossible to construct an exact Riemann solver . In those cases, one may refer to Toro (1997), who discussed a variety of approximate Riemann solvers for fluid

dynamics.

2.3 DISCONTINUOUS GALERKIN TIME DOMAIN METHODS FOR ACOUSTICS

In this section, I first derive a DG spatial discretization for Eqs.(2.1) . Then the time integration method used in this thesis is discussed. To complete the numerical method for acoustic wave equations, I also include two types of boundary conditions for the numerical simulation.

DG Spatial Discretization

The pressure-velocity formulation of the acoustic wave equations in 2D can be expressed as,

$$\begin{aligned} \rho(x,z)\frac{\partial v_x}{\partial t} + \frac{\partial p}{\partial x} &= 0, \\ \rho(x,z)\frac{\partial v_z}{\partial t} + \frac{\partial p}{\partial z} &= 0, \\ \frac{1}{\kappa(x,z)}\frac{\partial p}{\partial t} + \frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z} &= w(t)\delta(x-x_s)\delta(z-z_s). \end{aligned} \tag{2.13}$$

To simplify the notation, I rewrite Eqs.(2.13) into a matrix-vector compact form,

$$\frac{\partial \mathbf{q}}{\partial t} + A(x,z)\frac{\partial \mathbf{q}}{\partial x} + B(x,z)\frac{\partial \mathbf{q}}{\partial z} = [0,0,\kappa w(t)\delta(x-x_s)\delta(z-z_s)]^T., \tag{2.14}$$

where $\mathbf{q} = [v_x, v_z, p]^T$,

$$A(x,z) = \begin{bmatrix} 0 & 0 & 1/\rho(x,z) \\ 0 & 0 & 0 \\ \kappa(x,z) & 0 & 0 \end{bmatrix} \quad B(x,z) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1/\rho(x,z) \\ 0 & \kappa(x,z) & 0 \end{bmatrix}.$$

$\mathscr{T}_h$ is a conforming triangulation of the computational domain $\bar{\Omega} \subset \mathbb{R}^2$ made of non-overlapping simplices $\mathcal{T}_k$ $(k = 1, \cdots, K)$, i.e., triangles, with radius of the inscribed circle denoted by h_k. The intersection of two triangles $\mathcal{T}_j$ and $\mathcal{T}_k$ is either an edge or a vertex shared by them, or an empty set. For example, no neighbor triangles in the mesh are as shown in Fig.(2.3). In each triangle, the mass density

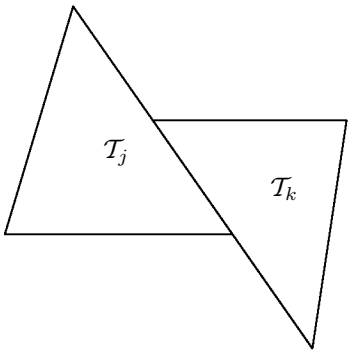

Figure 2.3: fault intersection of two triangles, which is not allowed in a conforming triangulation

and bulk modulus are assumed to be constant, that is $\rho(\mathbf{x})|_{\mathcal{T}_k} = \rho_k, \kappa(\mathbf{x})|_{\mathcal{T}_k} = \kappa_k$. Under this assumption the spatial derivatives of material parameters are excluded in the variational equations.

Unlike finite element methods giving a globally continuous approximation of the solutions, the DG approximation does not have such requirement and allows discontinuity between elements. On a triangular element $\mathcal{T}_k$, the interpolating Lagrange polynomials $\left\{l_j^k\right\}_j$ of degree N $(j = 1, \cdots, N_p$, where $N_p = (N+1)(N+2)/2)$ on

the α-optimized warp & blend nodal set $\left\{\mathbf{x}_j^k\right\}_j \subset \bar{\mathcal{T}}_k$ ($l_i^k(\mathbf{x}_j^k) = \delta_{ij}$) as indicated in Fig.(2.4) by Warburton (2006) are the basis functions for the spatial discretization. The numerical solution $\mathbf{q}_h$ of Eqs.(2.13) on $\mathcal{T}_k$ is then expressed as a time-dependent linear combination of basis functions,

$$\begin{aligned}
\mathbf{q}_h(\mathbf{x},t)|_{\mathcal{T}_k} &= [v_{x,h}(\mathbf{x},t), v_{y,h}(\mathbf{x},t), p_h(\mathbf{x},t)]^T|_{\mathcal{T}_k} \\
&= \mathbf{q}_h^k(\mathbf{x},t) = [v_{x,h}^k(\mathbf{x},t), v_{z,h}^k(\mathbf{x},t), p_h^k(\mathbf{x},t)]^T = \sum_{j=1}^{N_p} \mathbf{q}_h(\mathbf{x}_j^k,t) l_j^k(\mathbf{x}) \\
&:= \sum_{j=1}^{N_p} \mathbf{q}_j^k(t) l_j^k(\mathbf{x}) = \sum_{j=1}^{N_p} [v_{x,j}^k(t), v_{z,j}^k(t), p_j^k(t)]^T l_j^k(\mathbf{x}). \qquad (2.15)
\end{aligned}$$

The test function space V_h is then defined as the direct sum of broken polynomial spaces on each element, i.e., $V_h = \bigoplus_{k=1}^{K} \text{span}\left\{l_j^k\right\}_j$. This construction is a nodal version of the DG method discussed in Hesthaven and Warburton (2008). The representation of $\mathbf{q}_h$ with Lagrange basis functions allows me to get the facial values at the boundary interpolating points directly. The advantage of using nodal basis functions will be seen when computing the boundary integral and the numerical flux.

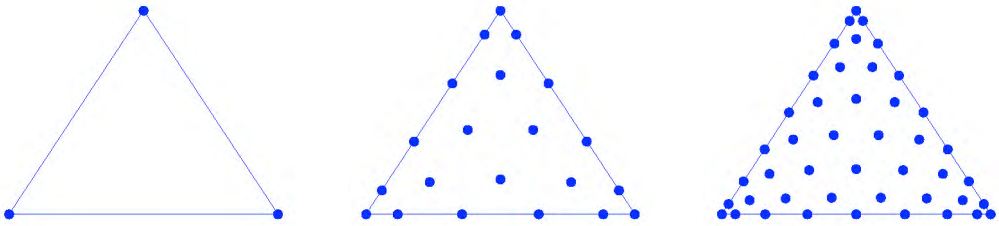

Figure 2.4: Illustrations of α-optimized warp & blend nodal distribution on the equilateral triangle. I use the α-optimized warp & blend nodes as the interpolating points of Lagrange polynomial basis functions. $N+1$ points live on each edge of the triangle when using basis functions of degree N. The three figures show the nodal distributions for different N. On the left, $N = 1$, $N_p = 3$; in the middle, $N = 5$, $N_p = 21$; on the right, $N = 8$, $N_p = 45$.

Multiplying Eqs.(2.14) by a test function v_h in V_h and taking integration over $\mathcal{T}_k$

yields,

$$\int_{\mathcal{T}_k} v_h \frac{\partial \mathbf{q}}{\partial t} + \int_{\mathcal{T}_k} v_h (A\frac{\partial \mathbf{q}}{\partial x} + B\frac{\partial \mathbf{q}}{\partial z}) dV = [0, 0, \kappa_k w(t) v_h(x_s, z_s)]^T.$$

Applying integration by parts gives,

$$\int_{\mathcal{T}_k} v_h \frac{\partial \mathbf{q}}{\partial t} + \int_{\partial \mathcal{T}_k} v_h \mathbf{F} dS - \int_{\mathcal{T}_k} (\frac{\partial v_h}{\partial x} A\mathbf{q} + \frac{\partial v_h}{\partial z} B\mathbf{q}) dV = [0, 0, \kappa_k w(t) v_h(x_s, z_s)]^T,$$

where $\mathbf{F}(\mathbf{x}, t) = (n_x A + n_z B)\mathbf{q}(\mathbf{x}, t)$ is the flux defined on $\mathcal{T}_k$'s boundary $\partial\mathcal{T}_k$ of which $\vec{n} = [n_x, n_z]$ is the unit normal vector. Since I assume A and B are constant over $\mathcal{T}_k$, no spatial derivatives of A and B are shown in the above equation. Next, substituting $\mathbf{q}$ and v_h with $\mathbf{q}_h^k$ and l_i^k respectively and then partitioning $\partial\mathcal{T}_k$ into the edges e_m^k $(m = 1, 2, 3)$ shared by $\mathcal{T}_k$'s neighbor triangle $\mathcal{T}_{k_m}$ gives the DG scheme,

$$\begin{aligned} \int_{\mathcal{T}_k} l_i^k \frac{\partial \mathbf{q}_h^k}{\partial t} + \sum_{m=1}^{3} \int_{e_m^k} l_i^k (\mathbf{F}_h^{k,m})^* dS \\ - \int_{\mathcal{T}_k} (\frac{\partial l_i^k}{\partial x} A^k \mathbf{q}_h^k + \frac{\partial l_i^k}{\partial z} B^k \mathbf{q}_h^k) dV = [0, 0, \kappa_k w(t) l_i^k(x_s, z_s)]^T. \end{aligned} \tag{2.16}$$

Several things need to be clarified here. Since the support of l_i^k is $\bar{\mathcal{T}}_k$, $l_i^k(x_s, z_s) = 0$ if $(x_s, y_s) \notin \bar{\mathcal{T}}_k$. For the numerical stability consideration, I assume $(x_s, y_s) \in \mathring{\mathcal{T}}_k$ for some k. Therefore, the right hand side of Eq.(2.16) is nonzero only for one element. $\vec{n}^{k,m} = (n_x^{k,m}, n_z^{k,m})$ is the unit normal vector of e_m^k pointing from $\mathcal{T}_k$ to $\mathcal{T}_{k_m}$. A^k and B^k denote the values of A and B on $\mathcal{T}_k$. On the left hand side of Eq.(2.16), $(\mathbf{F}_h^{k,m})^*$ is called the numerical flux approximating $\mathbf{F}$ on e_m^k. To match with the approximation of the solution, $(\mathbf{F}_h^{k,m})^*$ is defined as a linear combination of the basis functions' traces on e_m^k. Because $l_i^k(\mathbf{x}) \equiv 0$ on e_m^k if $\mathbf{x}_i^k \notin e_m^k$ (since $l_i^k(\mathbf{x})\big|_{e_m^k}$ is of degree N and vanishes at $N + 1$ points), $(\mathbf{F}_h^{k,m})^*$ is the sum of $N + 1$ terms associated with the

$N+1$ interpolating points on e_m^k. For m^{th} face of a triangular element, define the index set $\{n_j^m\}_{j=1}^{N+1} \subseteq \{1, \cdots, N_p\}$ such that $\mathbf{x}_{n_j^m}^k \in e_m^k, \forall\ k$. Denote $\mathbf{x}_j^{k,m} := \mathbf{x}_{n_j^m}^k$ and $l_j^{k,m} = l_{n_j^m}^k\big|_{e_m^k}$. $(\mathbf{F}_h^{k,m})^*$ is then expressed as,

$$\begin{aligned}(\mathbf{F}_h^{k,m})^*(\mathbf{x},t) &= \sum_{j=1}^{N+1}(n_x^{k,m}A^k + n_z^{k,m}B^k)\mathbf{q}_h^*(\mathbf{x}_j^{k,m},t)l_j^{k,m}(\mathbf{x}) \\ &= \sum_{j=1}^{N+1}(n_x^{k,m}A^k + n_z^{k,m}B^k)(\mathbf{q}_j^{k,m})^*(t)l_j^{k,m}(\mathbf{x}), \\ &= \sum_{j=1}^{N+1}(n_x^{k,m}A^k + n_z^{k,m}B^k)[(v_{x,j}^{k,m})^*(t),(v_{y,j}^{k,m})^*(t),(p_j^{k,m})^*(t)]^T l_j^{k,m}(\mathbf{x}),\end{aligned} \tag{2.17}$$

in which $\mathbf{q}_h^*(\mathbf{x}_j^{k,m},t)$ can be viewed as a reconstruction of the numerical solution $\mathbf{q}_h$ at $\mathbf{x}_j^{k,m}$. To see how $\mathbf{q}_h^*$ is related to the solution of a Riemann problem, expanding $(n_x^{k,m}A + n_z^{k,m}B^k)\mathbf{q}_h^*(x_j^{k,m},t)$ yields,

$$\begin{aligned}&[\frac{n_x^{k,m}}{\rho_k}p_h^*(\mathbf{x}_j^{k,m},t), \frac{n_z^{k,m}}{\rho_k}p_h^*(\mathbf{x}_j^{k,m},t), \kappa_k(n_x^{k,m}v_{x,h}^*(\mathbf{x}_j^{k,m},t) + n_z^{k,m}v_{z,h}^*(\mathbf{x}_j^{k,m},t))]^T \\ &:= [\frac{n_x^{k,m}}{\rho_k}(p_j^{k,m}(t))^*, \frac{n_z^{k,m}}{\rho_k}(p_j^{k,m}(t))^*, \kappa_k(v_{\vec{n},j}^{k,m}(t))^*]^T\end{aligned}$$

in which $(v_{\vec{n},j}^{k,m}(t))^*, (p_j^{k,m}(t))^*$ are the intermediate state solutions of the local Riemann problem at time t stated in Eqs.(2.12) where the initial data on the left and right of the interface are $\mathbf{q}_h^k$ and $\mathbf{q}_h^{k_m}$ at $\mathbf{x}_j^{k,m}$. The benefit of using nodal basis functions is that evaluations of $\mathbf{q}_h^k$, $\mathbf{q}_h^{k_m}$ at $\mathbf{x}_j^{k,m}$ cost nothing by recalling that $\mathbf{x}_j^{k,m}$ is an alias of an interpolating point $\mathbf{x}_{n_j^m}^k$ of $\mathcal{T}_k$ for some index n_j^m and by symmetry of the nodal points $\mathbf{x}_j^{k,m}$ is also an interpolating point of $\mathcal{T}_{k_m}$ on e_m^k, but denoted by another notation.

Applying Green's formula once again in Eq.(2.16) gives,

$$\int_{\mathcal{T}_k} l_i^k \frac{\partial \mathbf{q}_h^k(t)}{\partial t} + \sum_{m=1}^{3} \int_{e_m^k} l_i^k ((\mathbf{F}_h^{k,m})^* - \mathbf{F}_h^{k,m}) dS$$
$$+ \int_{\mathcal{T}_k} (A^k \frac{\partial \mathbf{q}_h^k}{\partial x} + B^k \frac{\partial \mathbf{q}_h^k}{\partial z}) l_i^k dV = [0, 0, \kappa_k w(t) l_i^k(x_s, z_s)]^T, \quad \forall\ i = 1, \cdots, N_p, \quad (2.18)$$

where $\mathbf{F}_h^{k,m} = (n_x^{k,m} A + n_z^{k,m} B)\mathbf{q}_h^k$. Substituting $\mathbf{q}_h^k$ in Eq.(2.18) with the linear combination of basis functions as indicated in Eq.(2.15) completes the spatial DG discretization construction and yields a time-dependent ordinary differential equation (ODE) system,

$$\begin{aligned} \sum_{j=1}^{N_p} \frac{\partial \mathbf{q}_j^k(t)}{\partial t} (l_i^k, l_j^k)_{\mathcal{T}_k} + \sum_{m=1}^{3} \sum_{j=1}^{N+1} (n_x^{k,m} A^k + n_z^{k,m} B^k)((\mathbf{q}_j^{k,m})^*(t) - \mathbf{q}_{n_j^m}^k(t))(l_i^k, l_j^{k,m})_{e_m^k} \\ + \sum_{j=1}^{N_p} A^k \mathbf{q}_j^k(t)(l_i^k, \frac{\partial l_j^k}{\partial x})_{\mathcal{T}_k} + \sum_{j=1}^{N_p} B^k \mathbf{q}_j^k(t)(l_i^k, \frac{\partial l_j^k}{\partial z})_{\mathcal{T}_k} \quad (2.19) \\ = [0, 0, \kappa_k w(t) l_i^k(x_s, y_s)]^T, \quad \forall\ i = 1, \cdots, N_p, \end{aligned}$$

where $(\cdot,\cdot)_{\mathcal{T}_k}, (\cdot,\cdot)_{e_m^k}$ are the standard L^2 inner product over $\mathcal{T}_k$ and e_m^k. Gathering Eq.(2.19) for each l_i^k together then gives a compact matrix-vector form by introducing

some new notations. First denote,

$$\mathbf{v}_x^k(t) = [v_{x,1}^k(,t), \cdots, v_{x,N_p}^k(t)]^T,$$
$$\mathbf{v}_z^k(t) = [v_{z,1}^k(t), \cdots, v_{z,N_p}^k(t)]^T,$$
$$\mathbf{p}^k(t) = [p_1^k(t), \cdots, p_{N_p}^k(t)]^T,$$
$$(\mathbf{p}^{k,m})^*(t) - \mathbf{p}^{k,m}(t) = [(p_1^{k,m})^*(t) - p_{n_1^m}^k(t), \cdots, (p_{N+1}^{k,m})^*(t) - p_{n_{N+1}^m}^k(t)]^T,$$
$$(\mathbf{v}_{\vec{n}}^{k,m})^*(t) = [(v_{\vec{n},1}^{k,m})^*(t), \cdots, (v_{\vec{n},N+1}^{k,m})^*(t)]^T,$$
$$\mathbf{v}_{\vec{n}}^{k,m}(t) = [v_{\vec{n},n_1^m}^k(t), \cdots, v_{\vec{n},n_{N+1}^m}^k(t)]^T,$$

where $v_{\vec{n},n_j^m}^k = n_x^{k,m} v_{x,n_j^m}^k + n_z^{k,m} v_{z,n_j^m}^k$. Then define the local volume mass matrix $M^k \in \mathbb{R}^{N_p \times N_p}$, the local stiffness matrices $S^{x,k}$, $S^{z,k} \in \mathbb{R}^{N_p \times N_p}$, the local edge mass matrix $M^{k,m}\mathbb{R}^{N_p \times (N+1)}$ as,

$$M_{ij}^k = (l_i^k, l_j^k)_{\mathcal{T}_k},\ S_{ij}^{x,k} = (l_i^k, \frac{\partial l_j^k}{\partial x})_{\mathcal{T}_k},\ S_{ij}^{z,k} = (l_i^k, \frac{\partial l_j^k}{\partial z})_{\mathcal{T}_k},\ M_{ij}^{k,m} = (l_i^k, l_j^{k,m})_{e_m^k}.$$

With these notations the compact matrix-vector form of Eq.(2.19) can be expressed as,

$$M^k \frac{d\mathbf{v}_x^k(t)}{dt} + \frac{1}{\rho_k} S^{x,k} \mathbf{p}^k(t) + \sum_{m=1}^{3} \frac{n_x^{e_m^k}}{\rho_k} M^{k,e_m^k}((\mathbf{p}^{k,m})^*(t) - \mathbf{p}^{k,m}(t)) = 0,$$
$$M^k \frac{d\mathbf{v}_z^k(t)}{dt} + \frac{1}{\rho_k} S^{z,k} \mathbf{p}^k(t) + \sum_{m=1}^{3} \frac{n_z^{e_m^k}}{\rho_k} M^{k,e_m^k}((\mathbf{p}^{k,m})^*(t) - \mathbf{p}^{k,m}(t)) = 0,$$
$$M^k \frac{d\mathbf{p}^k(t)}{dt} + \kappa_k S^{x,k} \mathbf{v}_x^k(t) + \kappa_k S^{z,k} \mathbf{v}_z^k(t) + \sum_{m=1}^{3} \kappa_k M^{k,e_m^k}((\mathbf{v}_{\vec{n}}^{k,m})^*(t) - \mathbf{v}_{\vec{n}}^{k,m}(t)) = \kappa_k w(t) \mathbf{l}^k(\mathbf{x}_s),$$

where $\mathbf{l}^k(\mathbf{x}_s) = [l_1^k(\mathbf{x}_s), \cdots, l_{N_p}^k(\mathbf{x}_s)]^T$. Furthermore multiplying the above equations

by $(M^k)^{-1}$ yields an fully explicit form,

$$\begin{aligned}
\frac{d\mathbf{v}_x^k(t)}{dt} &= -\frac{1}{\rho_k} D^{x,k}\mathbf{p}^k(t) - \sum_{m=1}^{3} \frac{n_x^{e_m^k}}{\rho_k} L^{k,m}((\mathbf{p}^{k,m})^*(t) - \mathbf{p}^{k,m}(t)), \\
\frac{d\mathbf{v}_z^k(t)}{dt} &= -\frac{1}{\rho_k} D^{z,k}\mathbf{p}^k(t) - \sum_{m=1}^{3} \frac{n_y^{e_m^k}}{\rho_k} L^{k,m}((\mathbf{p}^{k,m})^*(t) - \mathbf{p}^{k,m}(t)), \\
\frac{d\mathbf{p}^k(t)}{dt} &= -\kappa_k D^{x,k}\mathbf{v}_x^k(t) - \kappa_k D^{z,k}\mathbf{v}_z^k(t) \\
&\quad - \sum_{m=1}^{3} \kappa_k L^{k,m}((\mathbf{v}_{\vec{n}}^{k,m})^*(t) - \mathbf{v}_{\vec{n}}^{k,m}(t)) + \kappa_k w(t)(M^k)^{-1}\mathbf{l}^k(\mathbf{x}_s),
\end{aligned} \tag{2.20}$$

where $D^{x,k} = (M^k)^{-1}S^{x,k}$, $D^{z,k} = (M^k)^{-1}S^{z,k}$, $L^{k,m} = (M^k)^{-1}M^{k,e_m^k}$. Since each element is considered to be the image of the reference triangle $\hat{D} = \{(r,s)| -1 \leq r,s; r+s \leq 0\}$, any point $\mathbf{x}$ in the triangle T_k with vertices $\mathbf{x}^{k,1}, \mathbf{x}^{k,2}, \mathbf{x}^{k,3}$ is the image of a point (r,s) in the reference triangle under the following affine transform,

$$\mathbf{x} = -\frac{r+s}{2}\mathbf{x}^{k,1} + \frac{1+r}{2}\mathbf{x}^{k,2} + \frac{1+s}{2}\mathbf{x}^{k,3}.$$

. Then the mass matrix M^k for the k'th element is given by

$$M_{ij}^k = \int_{\hat{\mathcal{D}}} l_i(r,s) l_j(r,s) J^k(r,s) dr ds, \tag{2.21}$$

where $\{l_j(r,s)\}_{j=1}^{N_p}$ are the basis functions on $\hat{\mathcal{D}}$. Since The Jacobian $J^k(r,s)$ is constant, because the transform from the reference triangle $\hat{D}$ to T_k is a linear map. The mass matrix for $\mathcal{T}_k$ can be expressed as a scalar J^k multiple of the mass matrix on the reference element $\hat{\mathcal{D}}$,

$$M_{ij}^k = J^k M_{ij}, \tag{2.22}$$

where $M_{ij} = \int_{\hat{\mathcal{D}}} l_i(r,s) l_j(r,s) dr ds$. Therefore we do not need to store multiple mass matrices or their inverses. Similarly we can template the stiffness matrices. For the edge mass matrices, the reference element becomes $I = \{r | -1 \leq r \leq 1\}$. To get the explicit formulation of these matrices, one may refer to Hesthaven and Warburton (2008).

Now it can be seen clearly that once the numerical fluxes are computed, the time update is explicit and completely independent on each element. This feature is useful especially for parallelization.

Time Discretization

After the spatial discretization, an ODE system needs to be solved to march the numerical solution along time,

$$\frac{d\mathbf{Q}_h^k}{dt} = \mathcal{R}_h(\mathbf{Q}_h^k, (\mathbf{Q}_h^{k,1})^*, (\mathbf{Q}_h^{k,2})^*, (\mathbf{Q}_h^{k,3})^*, t). \tag{2.23}$$

where $\mathcal{R}_h$ is the operator grouping the right hand side terms in Eq.(2.20) together, and

$$\mathbf{Q}_h^k = \begin{bmatrix} v_x^k \\ v_y^k \\ (p^k) \end{bmatrix}, \quad (\mathbf{Q}_h^{k,m})^* = \begin{bmatrix} (v_{\vec{n}}^{k,m})^* \\ (p^{k,m})^* \end{bmatrix}$$

In this thesis, I use the low-storage five-stage fourth-order explicit Runge-Kutta (RK) method described in Alg.(1) by Carpenter and Kennedy (1994) to solve Eq.(2.23).

The coefficients a_i, b_i, c_i can be found in Tab.(2.1). The advantage of using such

Algorithm 1 Time Discretization

for each k **do**
  $\mathbf{U}_0^k = \mathbf{Q}_h^k(t), \quad \mathbf{h}_0^k = 0$
end for
for $i = 1$ to 5 **do**
  for each k **do**
    update $(\mathbf{Q}_h^{k,m})^*$with $\mathbf{U}_i^k$ and $\mathbf{U}_i^{k_m}, m = 1, 2, 3$
    $\mathbf{h}_i^k = a_i \mathbf{h}_{i-1}^k + \Delta t \mathcal{R}_h(\mathbf{U}_{i-1}^k, (\mathbf{Q}_h^{k,1})^*, (\mathbf{Q}_h^{k,2})^*, (\mathbf{Q}_h^{k,3})^*, t + c_i \Delta t)$,
    $\mathbf{U}_i^k = \mathbf{U}_{i-1}^k + b_i \mathbf{h}_i$
  end for
end for
$\mathbf{Q}_h^k(t + \Delta t) = \mathbf{U}_5^k$

scheme is the low memory usage, since only one additional storage is required. One more stage is present in this fourth-order scheme and seems to add cost. However, this scheme allows to use a larger time step Δt and thus reduces the overall computation cost.

For the numerical stability consideration, the time step Δt should not be taken too large. As proved, the high-order accurate RK DG methods require $\Delta t \sim \mathcal{O}(\Delta x)$. To be specific, the time step I use here has a bound like,

$$\Delta t \leq C \frac{\min\limits_k h_k}{(N+2)(N+1)}, \tag{2.24}$$

where h_k indicates the radius of the inscribed circle in the element $\mathcal{T}_k$ and C is a constant.

Boundary Conditions

In this thesis, the numerical simulation is always carried out on a bounded domain, whose boundaries are either the physical sea surface, landform or the fields far away

Table 2.1: Coefficients for the low-storage five-stage fourth-order explicit RK method

i	a_i	b_i	c_i
1	0	$\frac{1432997174477}{9575080441755}$	0
2	$-\frac{567301805773}{1357537059087}$	$\frac{5161836677717}{13612068292357}$	$\frac{1432997174477}{9575080441755}$
3	$-\frac{2404267990393}{2016746695238}$	$\frac{1720146321549}{2090206949498}$	$\frac{2526269341429}{6820363962896}$
4	$-\frac{3550918686646}{2091501179385}$	$\frac{3134564353537}{4481467310338}$	$\frac{2006345519317}{3224310063776}$
5	$-\frac{1275806237668}{842570457699}$	$\frac{2277821191437}{14882151754819}$	$\frac{2802321613138}{2924317926251}$

from the domain of interest. The free surface boundary condition and the absorbing boundary condition are two types of boundary conditions exclusively used in the seismic simulation.

Due to the high contrast discontinuities of material parameters between water and air, soil and air, waves are almost reflected back with a neglected amount of energy passing through when they hit the surface. The reflection boundary condition is used to replicate this action of waves in the numerical simulation. Until now, I always assume that $\mathcal{T}_k$ has three neighbor elements $\mathcal{T}_{k_m}$. But when the $\mathcal{T}_k$'s edge e_m^k belongs to the free surface boundary (the sea surface or landforms), $\mathcal{T}_{k_m}$ does not exist in the triangulation $\mathscr{T}_h$. I have to assign the initial data on the right of the interface in the Riemann problem Eqs.(2.12) so as to compute the numerical flux $(F_h^{k,m})^*$ on e_m^k. Applying the free surface boundary condition to the local Riemann problem yields

the initial data of Eqs.(2.12) as

$$v^2_{\vec{n}} = v^1_{\vec{n}}, \quad p^2 = -p^1.$$

This is an analogy to the free surface boundary condition in finite difference method in which the free surface boundary condition is applied to ghost grids in order to update the numerical solutions defined on the grid whose stencil is out of the computational domain.

In the seismic simulation, the domain of interest is relatively small compared to the distance waves propagate. The absorbing boundary condition does not try to mimic any physical scenarios, but is used to truncate the open domain problem into a finite one so that the numerical method can handle. In this thesis, I use a perfectly matched layer (PML), one kind of the absorbing boundary condition first designed by Berenger (1994) for Maxwell's equations. PML can be considered as layers wrapping the original computational domain to absorb outgoing incident wave from any incidence angle without any false reflection. A PML for acoustic wave

equations proposed by Abarbanel and Gottlieb (1998) can be stated as

$$\begin{aligned}
\frac{\partial v_x}{\partial t} + 2\eta_x v_x + \frac{1}{\rho(x,z)}\frac{\partial p}{\partial x} &= \eta_x P_x, \\
\frac{\partial v_z}{\partial t} + 2\eta_z v_z + \frac{1}{\rho(x,z)}\frac{\partial p}{\partial z} &= \eta_z P_z, \\
\frac{\partial p}{\partial t} + \kappa(x,z)\nabla\cdot\mathbf{v} + \eta_{x_p} Q_x + \eta_{z_p} Q_z &= \kappa(x,z) w(t)\delta(\mathbf{x}-\mathbf{x}_s), \\
\frac{\partial P_x}{\partial t} + \eta_x v_x &= 0, \\
\frac{\partial P_z}{\partial t} + \eta_z v_z &= 0, \\
\frac{\partial Q_x}{\partial t} + \eta_x Q_x &= \kappa(x,z) v_x, \\
\frac{\partial Q_z}{\partial t} + \eta_z Q_z &= \kappa(x,z) v_z,
\end{aligned} \tag{2.25}$$

where (P_x, P_z, Q_x, Q_z) are four auxiliary variables, and

$$\eta_\alpha = \begin{cases} \eta_{\alpha_{max}}\left(\dfrac{L_\alpha/2+\alpha}{d}\right)^2 & \alpha \in [-d - L_\alpha/2, -L_\alpha/2], \\ 0 & \alpha \in (-L_\alpha/2, L_\alpha/2], \\ \eta_{\alpha_{max}}\left(\dfrac{L_\alpha/2-\alpha}{d}\right)^2 & \alpha \in (L_\alpha/2, L_\alpha/2+d], \end{cases} \qquad \eta_{\alpha_p} = \frac{d\eta_\alpha}{d\alpha},$$

where $\alpha \in \{x, z\}$, and $\{L_x, L_z\}$ is the domain size, and d is the PML thickness. One can see that inside the domain Eq.(2.25) is exactly the original acoustic wave equations and in the PML the auxiliary variables work as damping sources to absorb the acoustic pressure as well as the velocity. Furthermore, the auxiliary variables are governed by ordinary differential equations, which can be computed with very little cost.

2.4 FINITE DIFFERENCE TIME DOMAIN METHOD FOR ACOUSTICS

In this section, I use 2-4 (second-order in time and fourth-order in space) staggered-grid finite difference method for Eqs.(2.13). This method later is used to compare with DGTD methods described before.

Two-dimensional staggered grids for Eqs.(2.13) employ three different sets of grids. As shown in Fig.(2.5), the grids for velocity components have half grid size displacement with the one for the pressure field along the component directions respectively. The 2-4 staggered-grid finite different method for Eqs.(2.13) is given as follows,

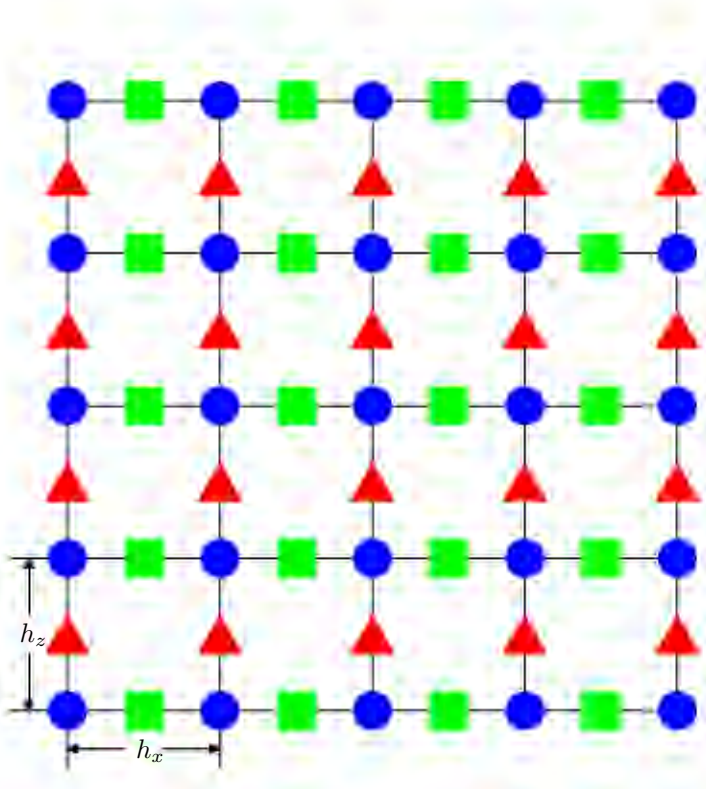

Figure 2.5: 2D staggered grids for the pressure-velocity formulation of acoustic wave equations. h_x and h_z denote spatial steps along x-axis and z-axis. The pressure grid is represented by circle. The horizontal and vertical velocity grids are represented by square and triangle respectively.

$$
\begin{aligned}
p^n_{i,j} &= p^{n-1}_{i,j} - \Delta t \frac{\kappa_{i,j}}{h_x}\Big(-\frac{1}{24}(v_x)^{n-1/2}_{i+3/2,j} + \frac{9}{8}(v_x)^{n-1/2}_{i+1/2,j} - \frac{9}{8}(v_x)^{n-1/2}_{i-1/2,j} + \frac{1}{24}(v_x)^{n-1/2}_{i-3/2,j}\Big) \\
&\quad - \Delta t \frac{\kappa_{i,j}}{h_z}\Big(-\frac{1}{24}(v_z)^{n-1/2}_{i,j+3/2} + \frac{9}{8}(v_z)^{n-1/2}_{i,j+1/2} - \frac{9}{8}(v_z)^{n-1/2}_{i,j-1/2} + \frac{1}{24}(v_z)^{n-1/2}_{i,j-3/2}\Big) \\
&\quad + \Delta t \kappa_{i,j} w^{n-1/2} \delta_{i,i_s}\delta_{j,j_s}, \\
(v_x)^{n+1/2}_{i+1/2,j} &= (v_x)^{n-1/2}_{i+1/2,j} - \Delta t \frac{1}{\rho_{i+1/2,j} h_x}\Big(-\frac{1}{24}p^n_{i+2,j} + \frac{9}{8}p^n_{i+1,j} - \frac{9}{8}p^n_{i,j} + \frac{1}{24}p^n_{i-1,j}\Big), \\
(v_z)^{n+1/2}_{i,j+1/2} &= (v_z)^{n-1/2}_{i,j+1/2} - \Delta t \frac{1}{\rho_{i,j+1/2} h_z}\Big(-\frac{1}{24}p^n_{i,j+2} + \frac{9}{8}p^n_{i,j+1} - \frac{9}{8}p^n_{i,j} + \frac{1}{24}p^n_{i,j-1}\Big),
\end{aligned}
$$

where Δt is the time step, $p^n_{i,j}$ for example denotes the value of the pressure variable at (ih_x, jh_z) and time $n\Delta t$, and i_s, j_s are the indices such that $x_s = i_s h_x$ and $z_s = j_s h_z$.

In this thesis, I use an open source software Iwave by Terentyev et al. (2008) as the FD solver. Iwave provides a general FD framework for solving time domain PDEs on both distributed and shared memory computer architectures. In Iwave, one can design FD schemes by providing the associated stencil information. The 2-4 staggered-grid finite difference method I use for comparison has been implemented in Iwave.

Chapter 3

Numerical Experiments

3.1 INTRODUCTION

In this section, I present numerical examples regarding the convergence tests of DG methods and the numerical comparison between DGTD methods and FDTD methods. First, I make convergence tests of DGTD methods for two cases: the point source wave in a homogeneous medium and the plane wave propagation in a two-layer medium. Since I have the analytic solutions for both cases, I can estimate the error and convergence rate precisely. The purpose of these tests is to make sure that my DGTD implementation is valid. As mentioned before, the interface error can downgrade the convergence rate of staggered-grid FD methods to 1st order. This error also appears in DGTD methods. In my DGTD implementation, each element in the triangulation is considered as a homogeneous medium. If the mesh misaligns with the material interface, the approximate model on this mesh would falsely represent the real model and then produce the interface error. Through numerical examples, I display the interface error in DGTD methods. Examples of a two-layer medium on

the interface-fitting mesh and the local refined mesh exhibit two ways to reduce this error. At last, the comparison of the DGTD and FDTD methods is carried out on two examples. Since analytic solutions are inaccessible in both examples, I first discuss the error estimate via Richardson extrapolation. Then I compare the computing cost measured in GFLOP (approximately 10^9 float point operations) for the two solutions by the DGTD and FDTD methods to have roughly the same accuracy (5% RMS error).

3.2 CONVERGENCE TESTS OF DGTD

I make convergence tests of DGTD methods through the point source wave and the plane wave. In the point source wave case, I measure the error of a trace at a given point on a series of globally refined meshes. In the plane wave case, the L^2 error of the pressure at a given time is computed to estimate the convergence rate. In both cases, DGTD methods behave the way as theories predict.

Point Source Wave

In this experiment, the computation domain $[-0.5, 0.5] \times [-0.5, 0.5]$ is a unit square centered at the origin. The mirror reflection boundary condition is applied. The material is homogeneous with $\rho = 1.0$, $c = 1.0$. A point source is located at $\mathbf{x}_s = (0, 1/4)$ with the source pulse,

$$w(t) = (t - t_0)e^{-(\pi f_0(t-t_0))^2}, \tag{3.1}$$

where $f_0 = 10$, $t_0 = 1.2/f_0$. A trace is recorded at $\mathbf{x}_r = (0, -1/4)$. The basis functions used are of degree 5. This example is carried out on a series of globally refined meshes

with grid size $1/10, 1/20, 1/40, 1/80$, respectively. As shown in Fig.(3.1), the trace error decreases 10^2 times ($\approx 2^6$, approximately) as the grid size decreases by half.

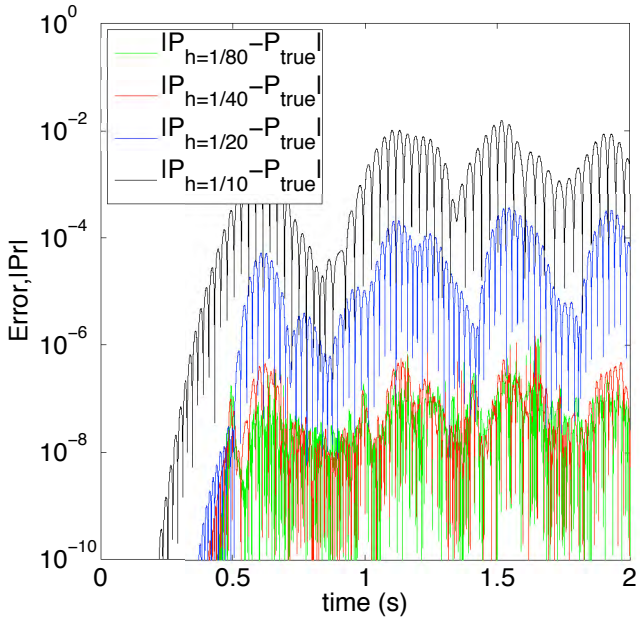

Figure 3.1: The point-wise trace error with respect to time of the point source wave experiment.

Plane Wave

In this example, a plane wave propagates in a two-layer medium with 90^o incident angle at the material interface. Though essentially this is a 1D problem, it provides a good example in 2D to test my DGTD implementation. The material interface is assumed to be a vertical line at $x = 0$. I use (ρ_l, c_l) and (ρ_r, c_r) to denote the density

and the wave speed for the materials at the two sides of the interface,

$$\rho_l = 2100\ kg/m^3,\ \ c_l = 2.3\ m/ms$$
$$\rho_r = 2300\ kg/m^3,\ \ c_r = 3.0\ m/ms$$

If v_z is 0 and (v_x, p) are smooth, then Eqs.(2.7) give me the analytic solutions for this case. Here I pick g as a Ricker's wavelet with central frequency $f_0 = 10$ Hz,

$$g(t) = (1 - 2(\pi f_0(t - t_0))^2)e^{-(\pi f_0(t-t_0))^2}.$$

The computation domain Ω is $[0, 1800\ m] \times [-15\ m, 15\ m]$ and the simulation time is 600 ms. I make the convergence test of DGTD methods with basis functions of different degrees on a series of globally refined interface-fitting meshes as indicated in Fig.(3.2). The estimated convergence rates shown in Tab.(3.1) agree with the optimal convergence rates

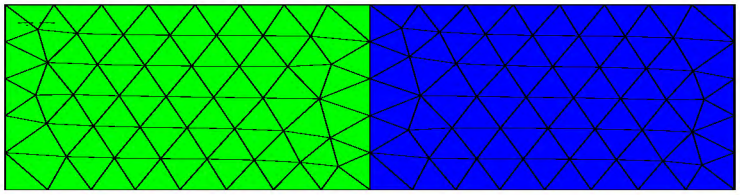

Figure 3.2: Illustration of the interface-fitting mesh. Different colors stand for different materials.

Table 3.1: Convergence tests for the plane wave case on the interface-fitting meshes. p, v_x, v_z are analytic solutions and $p_h, v_{x,h}, v_{z,h}$ are numerical solutions of the DGTD method. N indicates the order of basis functions in the DGTD method. The L^2 errors at $T = 600\ ms$ are measured for each field variable. R denotes the estimated convergence rate based on the L^2 error of the pressure.

h	N	$\|p_h(\cdot,T) - p(\cdot,T)\|_{L^2}$	$\|u_h(\cdot,T) - u(\cdot,T)\|_{L^2}$	$\|v_h(\cdot,T) - v(\cdot,T)\|_{L^2}$	R
10	1	0.7649	0.7762	0.1194	2.86
5	1	0.1053	0.1102	0.0307	2.74
2.5	1	0.0157	0.0177	0.0077	-
10	2	0.0084	0.0098	0.0044	3
5	2	0.0010	0.0012	5.54e-4	2.95
2.5	2	1.29e-4	1.50e-5	6.90e-5	-

3.3 INTERFACE ERROR

In this section, I illustrate the interface error in DGTD methods by the plane wave example when the triangular mesh misaligns with the material interface. One can see that the interface error is associated with the false representation of the model and therefore can not be eliminated by higher order schemes. Without modifying DGTD methods, I use the interface-fitting mesh and the local mesh refinement technique to reduce the interface error.

The model used here is the same as the one in the plane wave example. The trace recorded at $[500\ m, 0]$ as shown in Fig.(3.5) has two spikes corresponding to the direct wave and the reflected wave. The error in the direct wave is associated with the truncation error, while the error in the reflected wave is associated with the interface error. I use three sets of triangular meshes to test this example: the interface-fitting mesh, the mesh misaligned with the interface and the local refined mesh near the interface as shown in Fig.(3.2), Fig.(3.3) and Fig.(3.4), respectively.

Fig.(3.5) shows the traces at $[500\ m, 0]$ of the analytic and numerical solutions by DGTD with basis functions of degree 1 and 2 on the interface-fitting mesh. Both

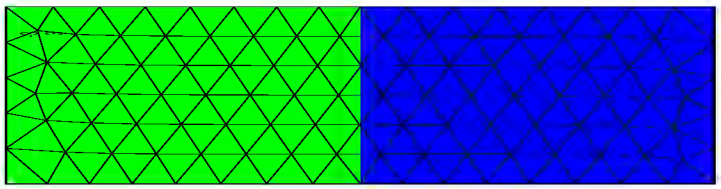

Figure 3.3: Illustration of the mesh misaligned with the interface.

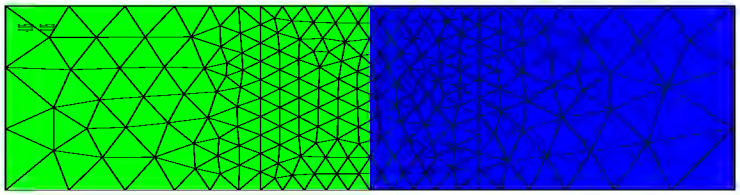

Figure 3.4: Illustration of the local refined mesh near the interface.

the direct wave and the reflected wave are resolved better by a high order scheme (basis functions of degree 2). Fig.(3.6) shows the trace at the same receiver of the analytic and numerical solution by DGTD with basis functions of degree 1, 2 and 4 on the interface-misaligned mesh. The direct wave is resolved accurately by a high order scheme. But the first-order error caused by the mesh misalignment dominates the error in the reflected wave. Fig.(3.7) illustrates the same simulation but on the local refined mesh. The time-shift effect in the reflected wave is much smaller than that on the interface-misaligned mesh.

3.4 COMPARISON OF DGTD AND FDTD

In the seismic community, people more care about the trace (seismogram) error at receivers (geophones) rather than the error measured over the entire computation domain. Hence the trace error at receivers is measured for the following numerical

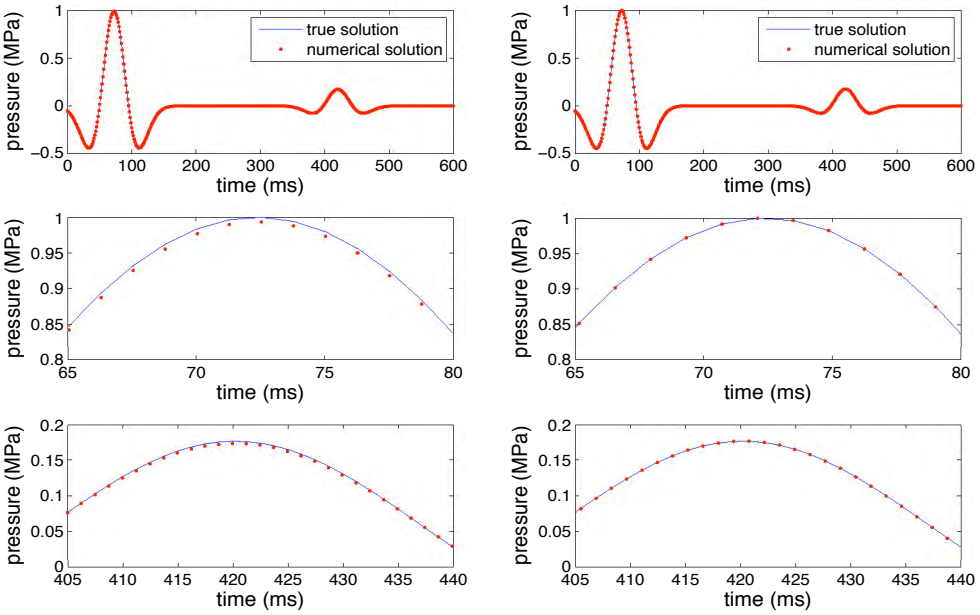

Figure 3.5: Traces at $[500\ m, 0]$ of the analytic and numerical solutions on the interface-fitting mesh as shown in Fig.(3.2). Basis functions of degree 1 and 2 are used to compute the trace plotted on the left and on the right, respectively. The top two plots are the entire traces from 0 ms to 600 ms. The middle two plots show the direct wave corresponding to the first spike of the entire trace. The two bottom plots show the reflected wave corresponding to the second spike of the entire trace.

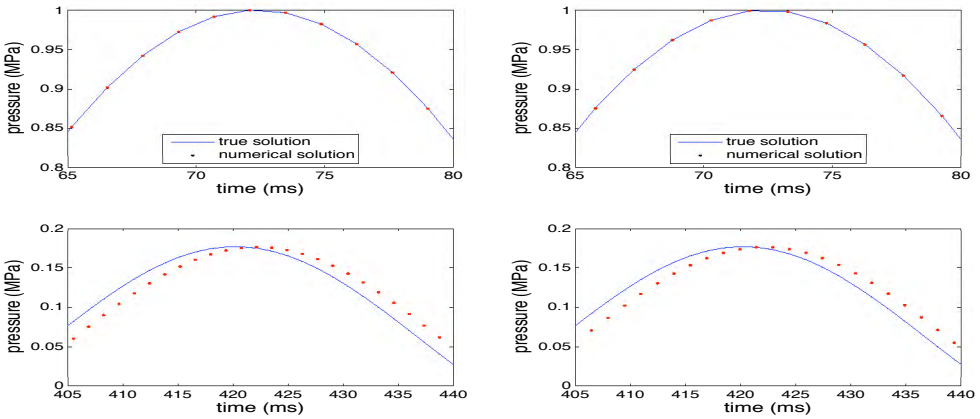

Figure 3.6: Traces at $[500\ m, 0]$ of the analytic and numerical solutions on the interface-misaligned mesh as shown in Fig.(3.3). Basis functions of degree 2 and 4 are used to compute the traces plotted on the left and right, respectively. The two plots at the top show the direct wave, while the ones at the bottom show the reflected wave.

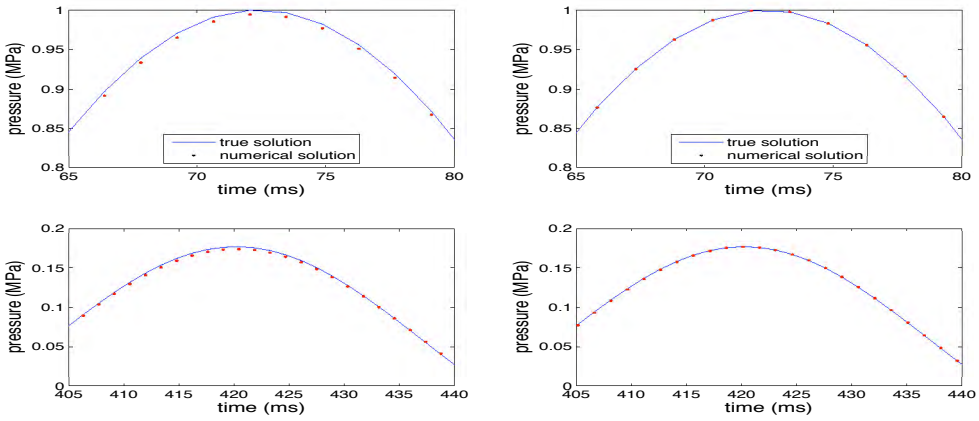

Figure 3.7: Traces at $[500\ m, 0]$ of the analytic and numerical solutions on the local refined mesh as shown in Fig.(3.4) . Basis functions of degree 1 and 2 are used to compute the traces plotted on the left and right, respectively. The plots at the top show the direct wave, while the ones at the bottom show the reflected wave.

experiments for the comparison involving DGTD and FDTD methods. Due to lack of analytic solutions for most of realistic models, the numerical error is estimated by Richardson extrapolation, that is, if assuming the numerical solution $D(h)$ differs from the analytic solution $\bar{D}$ by $E(h) = Ch^R + O(h^{R+1})$, then

$$E(h) \simeq \frac{D(2h) - D(h)}{2^R - 1}, \tag{3.2}$$

where R can be estimated by having $E(2h)$,

$$R \simeq \log_2 \frac{E(2h)}{E(h)}. \tag{3.3}$$

The programs for FDTD and DGTD methods are written in the ISO C language and use MPI and domain decomposition for parallelization. The numerical experiments were performed in single precision on a 2.66GHz Intel Core2 Quad Q9450 CPU. The operating system is Linux (2.6.18 kernel) and the compiler is GNU C compiler (version

4.1.2). For both examples, the numerical cost is measured by the total number of floating point operations (GFLOP) and the wall clock time.

Square-circle model

In this example, I assume a square domain of the size 1 km by 1 km with two different materials separated by a circle of radius 125 m at the center. As shown in Fig.(3.8), the parameters for the simulation are defined as,

- inside the circle: $\rho = 1500\ kg/m^3, c = 2000\ m/s$;
- outside the circle: $\rho = 1000\ kg/m^3, c = 1000\ m/s$;
- a point source at $(0.5\text{ km}, 0.25\text{ m})$ with a source pulse $w(t)$ defined in Eq.(3.1). The central frequency is 10 Hz;
- 41 geophones (receivers) are put at the depth 0.75 km with offset from 0.1 km to 0.9 km at spatial interval of 20 m. The time span of the simulation is $[0, 2s]$, and all the traces are sampled at temporal interval of 5 ms.

The grid size in FDTD is known because uniform grids are used. When it comes to DGTD, the grid sizes vary element by element. Therefore, as far as DGTD is concerned, I give the grid size range of the triangular mesh. I use 2-4 staggered-grid FDTD on 10 m, 5 m and 2.5 m grids and estimate the relative root square mean (RMS) error and the convergence rate at each receiver. As shown in Fig.(3.9), the estimated convergence rates of 2-4 staggered-grid FDTD are around order 1.4 at certain receivers right beneath the circular region. 2-4 staggered-grid FDTD on the 2.5 m grid achieves 3% RMS error and takes 33.2 GFLOP. The wall clock time is 19 sec on a single core. Fig.(3.10) illustrates the RMS errors and estimated convergence

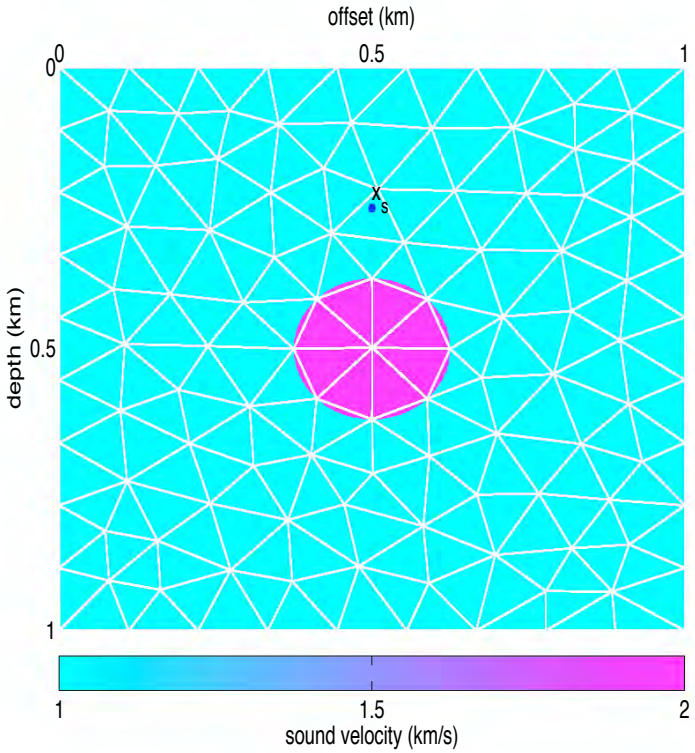

Figure 3.8: Sound velocity field and an interface-fitting mesh for Square-Circle Model. Source location is $\mathbf{x}_s = (0.5 \text{ km}, 0.25 \text{ km})$

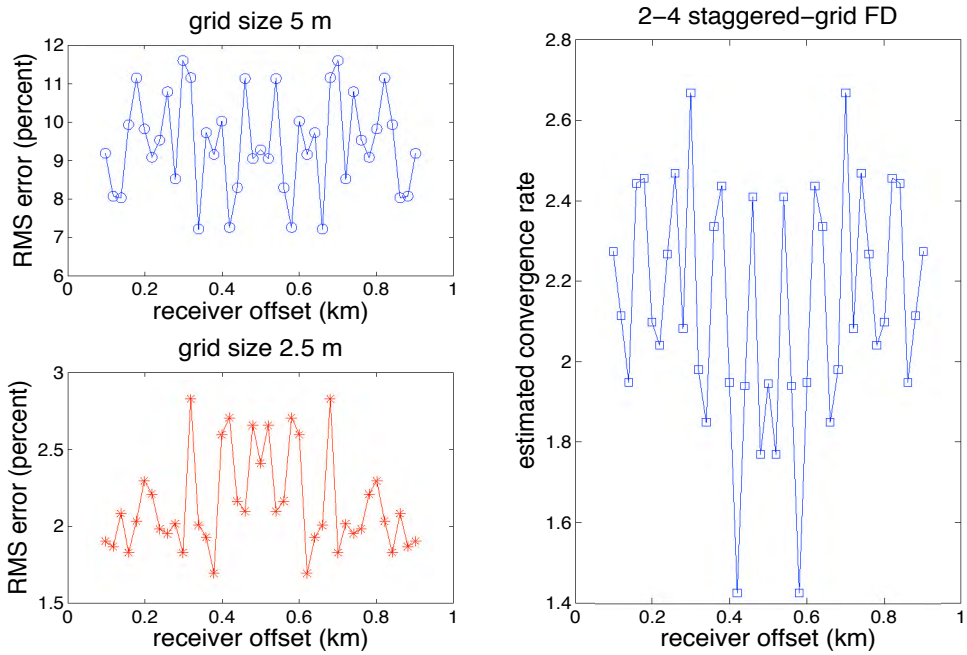

Figure 3.9: The RMS errors and estimated convergence rates by Richardson extrapolation for 2-4 staggered-grid FDTD on the square-circle model. All the traces are sampled at temporal interval of 5 ms.

rates of DGTD method with basis functions of degree 4. The DGTD method achieves 2% RMS error on a mesh with grid size range $6 \sim 14$ m and overall 2nd convergence rate. But the computation cost of DGTD method is 2465 GFLOP and the wall clock time is 760 sec on a single core. The first-order interface error in FDTD method is well resolved with less computation cost in this case due to the simplicity of the model structure. However, it'll be seen in the next example that the interface error in FDTD method ultimately meddles the RMS error as time goes by and one has to spend more computation cost achieving the same accuracy when the model becomes more complex.

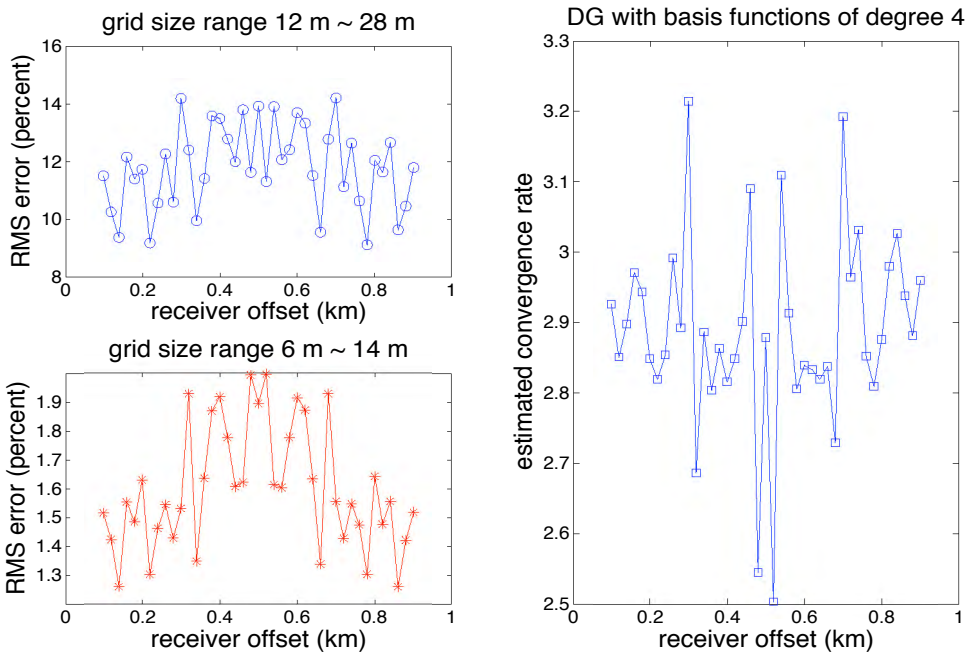

Figure 3.10: The RMS errors and estimated convergence rates by Richardson extrapolation for DGTD with basis functions of degree 4 on the square-circle model. All the traces are sampled at temporal interval of 5 ms.

2D Dome Model

In this experiment, the 2D dome model as found in Symes and Vdovina (2009) is set up for the comparison of DGTD and FDTD. Fig.(3.11) shows the material wave

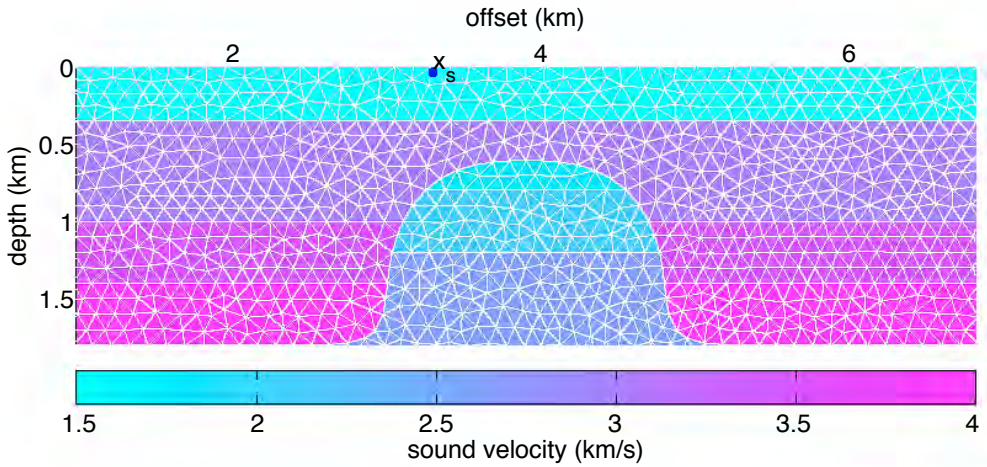

Figure 3.11: Sound velocity field and an interface-fitting mesh for 2D Dome Structure. Source location is $\mathbf{x}_s = (3.3 \text{ km}, 0.04 \text{ km})$.

speed of this model together with an interface-fitting mesh. The computation domain is a $[0, 7800 \text{ m}] \times [0, 1800 \text{ m}]$ rectangle, and the dome interface is located at the center. The wave propagation is forced by a point source at $\mathbf{x}_s = (3300 \text{ m}, 40 \text{ m})$. The source pulse $w(t)$ is defined in Eq.(3.1) with central frequency 15 Hz. This wavelet has significant energy at 30 Hz or a wavelength of 50 m. The time span of the simulation is $[0, 3000\ ms]$. PML layers are allocated on the left, right and at the bottom of the computation domain to absorb the outgoing waves, and the free surface boundary condition is applied at the top boundary. I compare 4th order Runge-Kutta scheme in time, nodal DG method with basis functions of degree 2 and 2-4 staggered-grid Taylor series stencil FDTD method on the 2D dome structure. According to Alford et al. (1974), 5 grid points per wavelength is marginal for the 4th order scheme in

FDTD. The numerical simulations by FDTD are tested on 5 m, 2.5 m, 1.25 m and 0.625 m grids. For DGTD, I use three sets of interface-fitting meshes generated by the global refinement process. The grid size range are $10.66 \sim 29.26$ m, $5.34 \sim 14.62$ m and $2.66 \sim 7.32$ m respectively. Fig.(3.12) shows the pressure trace by the 2-4

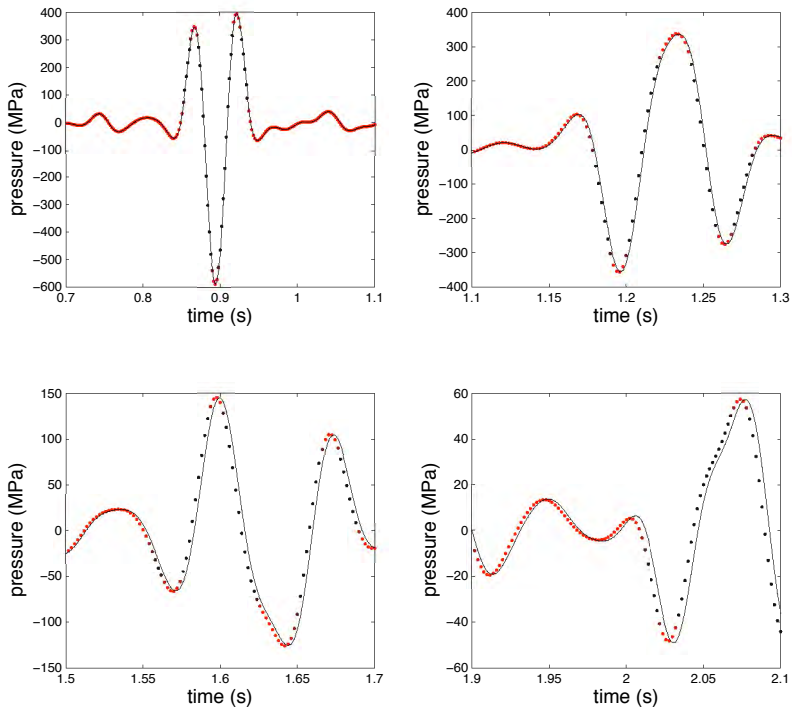

Figure 3.12: Pressure traces by the 2-4 staggered-grid FDTD on the 2D dome model at different time windows: upper-left: 0.7-1.1 s, upper-right: 1.1-1.3 s, lower-left: 1.5-1.7 s, lower-right: 1.9-2.1 s. $h = 2.5$ m (red dots) and $h = 1.25$ m (black line).

staggered-grid FDTD method at different time windows. As time goes by, the time shift effect caused by the interface error becomes more and more strong.

Tab.(3.2) lists the RMS error, computation cost (# GFLOP) of FDTD and DGTD at the receiver (2300 m, 20 m) at different time windows as well as the timing for each simulation. Both traces by DGTD and FDTD methods are sampled at tem-

poral interval of 2 ms. According to Richardson extrapolation, it is easily inferred that FDTD acts as a **first order** method while DGTD on interface-fitting meshes converges at **second order** convergence rate.

Table 3.2: RMS errors at different time windows, computation cost (# of GFLOP) and the wall clock time (second) on a single core. Rows 4-7 indicate the relative RMS errors at time windows: 0.7-1.1 s, 1.1-1.3 s, 1.5-1.7 s 1.9-2.1 s, respectively. Both traces by DGTD and FDTD methods are sampled at the same temporal interval of 2 ms.

	2-4 staggered-grid FDTD				DGTD ($N = 2$)	
grid size	2.5 m	1.25 m	0.625 m	0.3125 m	$5 \sim 15$ m	$2.7 \sim 7.3$ m
# of elements	NA				210176	840704
0.7-1.1 s	6.61%	4.64%	1.65%	0.82%	6.11%	0.31%
1.1-1.3 s	22.35%	12.30%	5.54%	2.76%	5.31%	0.60%
1.5-1.7 s	37.75%	20.06%	9.45%	4.70%	6.72%	0.79%
1.9-2.1 s	52.35%	28.64%	13.92%	6.91%	7.23%	1.15%
# of GFLOP	1286.3	1.03e+4	8.22e+4	6.57e+5	1.29e+4	1.03e+5
time	542 s	4125 s	32778 s	261991 s	6457 s	52401 s

Chapter 4

Curvilinear Discontinuous Galerkin Method

4.1 INTRODUCTION

As seen in the previous chapter, DGTD methods achieve a suboptimal 2nd order convergence rate when curved material interfaces are present. The incomplete representation to the model by simplices (triangles in 2D) brings in errors dominating the numerical results. In this chapter, I discuss DG methods on curvilinear elements. The curvilinear elements fit accurately with the material interface or boundaries by high order polynomials, and are hence able to complement the accuracy of the DG solver.

This chapter is organized as follows. First I talk about the procedure of forming curvilinear elements according to the interface geometry. Then the low-storage curvilinear DGTD method is formulated for the pressure-velocity formulation of AWEs. At last, the numerical results on square-circle model and 2D dome model are provided

to demonstrate the advantage of using curvilinear DG methods.

4.2 CURVILINEAR ELEMENT

By using the straight sided triangular mesh, we can not expect the edges in such mesh conform exactly with the curved material interfaces or boundaries. Such mesh fits the interfaces with piecewise linear polynomials. For example any point $\mathbf{x}$ in $\mathcal{T}_k$ with vertices $\mathbf{x}^{k,1}, \mathbf{x}^{k,2}, \mathbf{x}^{k,3}$ is the image of a point (r, s) in $\hat{\mathcal{D}} = \{(r, s) | -1 \leq r, s; r + s \leq 0\}$ under the following linear affine transform,

$$\mathbf{x} = -\frac{(r+s)}{2}\mathbf{x}^{k,1} + \frac{(1+r)}{2}\mathbf{x}^{k,2} + \frac{(1+s)}{2}\mathbf{x}^{k,3}. \tag{4.1}$$

Instead of the above linear coordinate transform, we may use an isoparametric transform $\mathbf{x} = \sum_{j=1}^{N_p} \mathbf{x}_j^k l_j(r, s)$ ($\{l_j\}_j$ are interpolating Lagrange polynomials on $\hat{\mathcal{D}}$) mapping the reference triangle $\hat{\mathcal{D}}$ to a curvilinear element $\mathcal{D}_k$ such that $\mathcal{D}_k$ fits more precisely with the interfaces or boundaries.

To form such curvilinear elements, I follow the steps in Hesthaven and Warburton (2008),

- identify element edges that need to be curved,
- reallocate the vertices and facial interpolating points on the curved material interfaces or boundaries,
- blend the face deformation of each curved face into the interior interpolating points through Gordon-Hall blending of face node deformation discussed in Gordon and Hall (1973).

After identifying element edges that need to be curved, various approaches can be used to push the vertices and facial interpolating points onto the curved interfaces or boundaries. For example, in the square-circle model the curved interface is circular. A linear distribution of polar angles according to the circle center and radius is created for the nodes that need to be moved. In the 2D dome model, the curved interface is a dome-shape. The nodes are reallocated at the intersection of the dome curve and the line starting at the original node location and along the normal direction of the corresponding edge. Then I use Gordon-Hall blending of face node deformation to blend the edge deformation into the interior nodes. Fig.(4.1) illustrates the process of forming two curvilinear elements conforming with a circular interface. The vertices $(1, 0)$ and $(1/2, \sqrt{3}/2)$ of the original triangle is on the interface but the facial interpolating points between them are not. I first move those facial interpolating points onto the circular interface according to the distribution of their polar angles. Then the deformation is blended into the interior interpolating points.

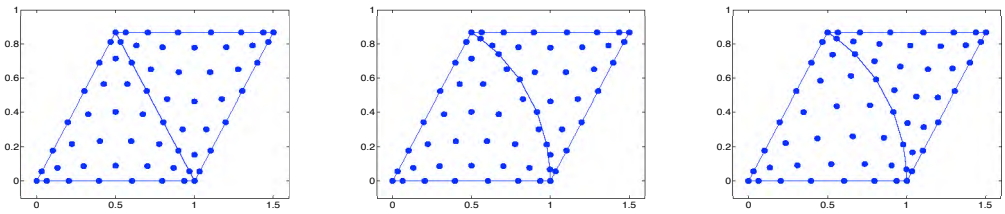

Figure 4.1: Left: original straight sided triangular elements. Middle: the facial interpolating points between $(1, 0)$ and $(1/2, \sqrt{3}/2)$ are moved to the circular interface. Right: the facial deformation is blended into the interior interpolating points.

4.3 CURVILINEAR DG FORMULATION

The semi-discrete symmetric DG variational equations for the pressure-velocity formulation of AWEs in 2D demand to find $(\mathbf{v}, p) \in (V_h)^3$ such that in the k'th curvilinear

element $\mathcal{D}_k$ ($\mathcal{D}_k = \mathcal{T}_k$ if the element is straight sided),

$$\begin{aligned}
\rho_k\Big(\phi, \frac{\partial \mathbf{v}}{\partial t}\Big)_{\mathcal{D}_k} - \Big(\nabla\phi, p\Big)_{\mathcal{D}_k} &= -\Big(\phi, \mathbf{n}p^*\Big)_{\partial\mathcal{D}_k}, \\
\frac{1}{\kappa_k}\Big(\psi, \frac{\partial p}{\partial t}\Big)_{\mathcal{D}_k} + \Big(\psi, \nabla\cdot\mathbf{v}\Big)_{\mathcal{D}_k} &= -\Big(\psi, \mathbf{n}\cdot(\mathbf{v}^* - \mathbf{v}^-)\Big)_{\partial\mathcal{D}_k},
\end{aligned} \tag{4.2}$$

for all $\phi, \psi \in V_h$. The '-' indicates the boundary trace of the solution within the k'th element. $\mathbf{v}^*, p^*$ are the numerical flux terms. The reason of starting with the symmetric variational equations is to guarantee numerical stability in the curvilinear DG formulation. Instead of expressing the solution as a linear combination of basis functions on $\mathcal{D}_k$ as in Eq.(2.15), I use the basis functions $\{l_j(r,s)\}_{j=1}^{N_p}$ on the reference element $\hat{\mathcal{D}}$ to do so,

$$\begin{aligned}
\mathbf{v}_h(\mathbf{x}(r,s),t)|_{\mathcal{D}_k} &= \sum_{j=1}^{N_p} \mathbf{v}_j^k(t) l_j(r,s), \\
p_h(\mathbf{x}(r,s),t)|_{\mathcal{D}_k} &= \sum_{j=1}^{N_p} p_j^k(t) l_j(r,s).
\end{aligned} \tag{4.3}$$

Then substituting these into the volume inner-product terms in the variational formulation Eqs.(4.2) and using the Einstein notation yields,

$$\begin{aligned}
\rho_k(l_i, l_j)_{\mathcal{D}_k}\frac{\partial \mathbf{v}_j^k}{\partial t} &= (\nabla_{x,z} l_i, l_j)_{\mathcal{D}_k}\ p_j^k - (l_i, \mathbf{n}p^*)_{\partial\mathcal{D}_k}, \\
\frac{1}{\kappa_k}(l_i, l_j)_{\mathcal{D}_k}\frac{\partial p_j^k}{\partial t} &= -(l_i, \nabla_{x,z} l_j)_{\mathcal{D}_k}\cdot\mathbf{v}_j^k - (l_i, \mathbf{n}\cdot(\mathbf{v}^* - \mathbf{v}^-))_{\partial\mathcal{D}_k},
\end{aligned} \tag{4.4}$$

The mass matrix M^k for the k'th element is given by,

$$M_{ij}^k = \int_{\mathcal{D}_k} l_i(r,s) l_j(r,s)\,\mathrm{d}x\,\mathrm{d}z = \int_{\hat{\mathcal{D}}} l_i(r,s) l_j(r,s) J^k(r,s)\,\mathrm{d}r\,\mathrm{d}s, \tag{4.5}$$

and the Jacobian $J^k(r,s)$ for the k'th element $\mathcal{D}_k$ by

$$J^k(r,s) = \left|\frac{\partial \mathbf{x}}{\partial r} \times \frac{\partial \mathbf{x}}{\partial s}\right| \tag{4.6}$$

As mentioned before when $\mathcal{D}_k$ is a straight sided triangular element Eq.(4.1) shows that the Jacobian J^k will be a constant within $\mathcal{D}_k$, independent of (r,s). In this case the mass matrix for $\mathcal{D}_k$ can be expressed as a scalar J^k multiple of the mass matrix on the reference element $\hat{\mathcal{D}}$. Because only the mass matrix on the reference element need to be stored, the computational storage of DGTD scales as CN_pK for a constant C independent of N or K.

Now a mesh is assumed to contains a subset of $K_c \leq K$ curvilinear elements that are deformed to conform with the curved interfaces or boundaries. An isoparametric transform $\mathbf{x}(r,s) = \sum_{j=1}^{N_p} \mathbf{x}_j^k l_j(r,s)$ is used to map $\hat{\mathcal{D}}$ to $\mathcal{D}_k$. Since the transform is an N'th order polynomial, the Jacobian J^k is no longer a constant on $\mathcal{D}_k$. Given that computing the mass matrix for each element on the fly during simulations is prohibitively expensive, it is common to precompute them before time stepping commences. This additional storage requirement scales as $K_c N_p^2$. If K_c is even a modest fraction of the total number of elements K then this can be the dominant storage cost. In the following the strategy to reduce or remove this storage overhead is discussed.

Weighting the variational spaces

One strategy for templating curvilinear elements is to modify test and trial spaces by the weighting approximation space,

$$V_h^J = \bigoplus_{k=1}^{K} \text{span}\Big\{ \frac{l_j(r,s)|_{\mathcal{D}_k}}{\sqrt{J^k(r,s)}} \Big\}_j. \tag{4.7}$$

The Jacobian matrix from the mass matrix can be eliminated by making the substitution,

$$\begin{aligned} \mathbf{v}_h(\mathbf{x}(r,s),t)|_{\mathcal{D}_k} &= \sum_{j=1}^{N_p} \tilde{\mathbf{v}}_j^k(t) \frac{l_j(r,s)}{\sqrt{J^k(r,s)}}, \\ p_h(\mathbf{x}(r,s),t)|_{\mathcal{D}_k} &= \sum_{j=1}^{N_p} \tilde{p}_j^k(t) \frac{l_j(r,s)}{\sqrt{J^k(r,s)}}. \end{aligned} \tag{4.8}$$

In words the variational space is replaced with polynomials weighted by the reciprocal of the square root of the transform Jacobian specific to each element. Equivalently $\sqrt{J}\mathbf{v}$ and $\sqrt{J}p$ are approximated instead of $\mathbf{v}$ and p in the regular DGTD formulation.

The mass matrix on a curvilinear element then becomes,

$$M_{ij}^k = \int_{\hat{\mathcal{D}}} \frac{l_i(r,s)}{\sqrt{J^k}} \frac{l_j(r,s)}{\sqrt{J^k}} J^k(r,s) dr ds = \int_{\hat{\mathcal{D}}} l_i(r,s) l_j(r,s) dr ds = M_{ij} \tag{4.9}$$

With this choice of test and trial spaces the DGTD variational formulation for AWE is,

$$\begin{aligned} \rho_k M_{ij} \frac{\partial \tilde{\mathbf{v}}_j^k}{\partial t} &= \Big(\nabla_{x,z} \frac{l_i}{\sqrt{J^k}}, \frac{l_j}{\sqrt{J^k}} \Big)_{\mathcal{D}_k} \tilde{p}_j^k - \Big(\frac{l_i}{\sqrt{J^k}}, \mathbf{n} p^* \Big)_{\partial \mathcal{D}_k}, \\ \frac{1}{\kappa_k} M_{ij} \frac{\partial \tilde{p}_j^k}{\partial t} &= -\Big(\frac{l_i}{\sqrt{J^k}}, \nabla_{x,z} \frac{l_j}{\sqrt{J^k}} \Big)_{\mathcal{D}_k} \cdot \tilde{\mathbf{v}}_j^k - \Big(\frac{l_i}{\sqrt{J^k}}, \mathbf{n} \cdot (\mathbf{v}^* - \mathbf{v}^-) \Big)_{\partial \mathcal{D}_k}. \end{aligned} \tag{4.10}$$

The storage requirement for the mass matrix is removed with a modified approximation space. Consequently the right hand side residuals are more difficult to evaluate. In the original variational equation all the integrands are polynomial, but now in the new form the integrands are rational functions. The form can be slightly simplified to,

$$\begin{aligned}
\rho_k M_{ij} \frac{\partial \tilde{\mathbf{v}}_j^k}{\partial t} &= (\nabla_{x,z} l_i, l_j)_{\hat{\mathcal{D}}}\ \tilde{p}_j^k - \Big(l_i, \frac{l_j}{2} \nabla_{x,z} \log(J^k)\Big)_{\hat{\mathcal{D}}}\ \tilde{p}_j^k - \Big(\frac{l_i}{\sqrt{J^k}}, \mathbf{n} p^*\Big)_{\partial \mathcal{D}_k} \\
\frac{1}{\kappa_k} M_{ij} \frac{\partial \tilde{p}_j^k}{\partial t} &= -\ (l_i, \nabla_{x,z} l_j)_{\hat{\mathcal{D}}} \cdot \tilde{\mathbf{v}}_j^k + \Big(l_i, \frac{l_j}{2} \nabla_{x,z} \log(J^k)\Big)_{\hat{\mathcal{D}}} \cdot \tilde{\mathbf{v}}_j^k \\
&\quad - \Big(\frac{l_i}{\sqrt{J^k}}, \mathbf{n} \cdot (\mathbf{v}^* - \mathbf{v}^-)\Big)_{\partial \mathcal{D}_k},
\end{aligned} \tag{4.11}$$

which is derived from Eq.(4.10) by expanding $\Big(\frac{l_i}{\sqrt{J^k}}, \nabla_{x,z} \frac{l_j}{\sqrt{J^k}}\Big)_{\mathcal{D}_k}$ as

$$\begin{aligned}
\Big(\frac{l_i}{\sqrt{J^k}}, \nabla_{x,z} \frac{l_j}{\sqrt{J^k}}\Big)_{\mathcal{D}_k} &= \int_{\hat{\mathcal{D}}} \frac{l_i(r,s)}{\sqrt{J^k}} \nabla_{x,z} \frac{l_j(r,s)}{\sqrt{J^k}} J^k(r,s) dr ds \\
&= (l_i, \nabla_{x,z} l_j)_{\hat{\mathcal{D}}} - \Big(l_i, \frac{l_j}{2} \nabla_{x,z} \log(J^k)\Big)_{\hat{\mathcal{D}}}
\end{aligned} \tag{4.12}$$

Notice that

$$\nabla_{x,z} l_j = \begin{pmatrix} \dfrac{\partial r}{\partial x}\dfrac{\partial l_j}{\partial r} + \dfrac{\partial s}{\partial x}\dfrac{\partial l_j}{\partial s} \\ \\ \dfrac{\partial r}{\partial z}\dfrac{\partial l_j}{\partial r} + \dfrac{\partial s}{\partial z}\dfrac{\partial l_j}{\partial s} \end{pmatrix},$$

and $(\frac{\partial r}{\partial x}, \frac{\partial s}{\partial x}, \frac{\partial r}{\partial z}, \frac{\partial s}{\partial z})$ have different value for each element when using curvilinear elements. Hence though $(\nabla_{x,z} l_i, l_j)_{\hat{\mathcal{D}}}$ and $(l_i, \nabla_{x,z} l_j)_{\hat{\mathcal{D}}}$ are integrations on the reference element, unlike the regular DG method we have to compute these integrations with cubature rule. The cubature rule is also applied to the additional low order correction

term, for example,

$$\Big(l_i, \frac{l_j}{2}\nabla_{x,z}\log(J^k)\Big)_{\hat{\mathcal{D}}} = \sum_{n=1}^{N_c} \omega_i^c l_i(r_n^c, s_n^c) l_j(r_n^c, s_n^c) \nabla_{x,z}\log\big(J^k(\mathbf{x}(r_n^c, s_n^c))\big), \qquad (4.13)$$

where $\{(r_n^c, s_n^c)\}_{n=1}^{N_c}$, $\{\omega_n^c\}_{n=1}^{N_c}$ are the cubature nodes and weights on $\hat{\mathcal{D}}$. For the weighted boundary integration, we may first transform the edges to the reference interval $I = [-1, 1]$ and then apply the quadrature rule on I.

4.4 NUMERICAL RESULTS

I apply the curvilinear DG method to the same square-circle model and compute the numerical results and the RMS errors on three meshes with grid size range $56 \sim 114$ m, $25 \sim 56$ m and $12 \sim 28$ m respectively. The basis functions are polynomials of degree 8. Fig.(4.2) displays the nodal distribution of the curvilinear element of degree 8 for the square-circle model near the circular region. Fig.(4.3) shows the

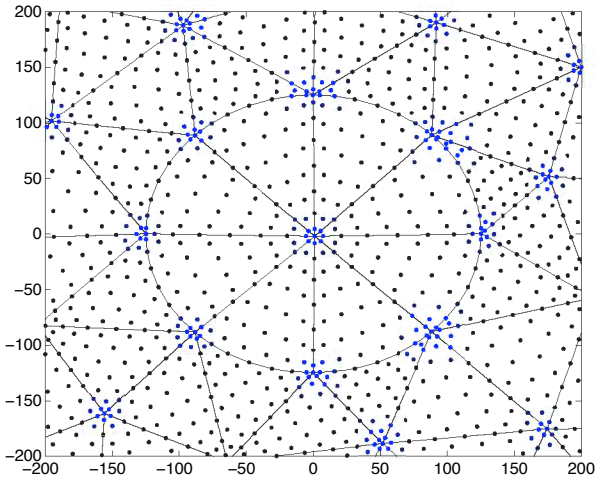

Figure 4.2: Illustration of the nodal distributions of the curvilinear element of degree 8 near the circular region of the square-circle model.

RMS errors and estimated convergence rates for each receiver. The curvilinear DG method achieves the optimal convergence rate for this example.

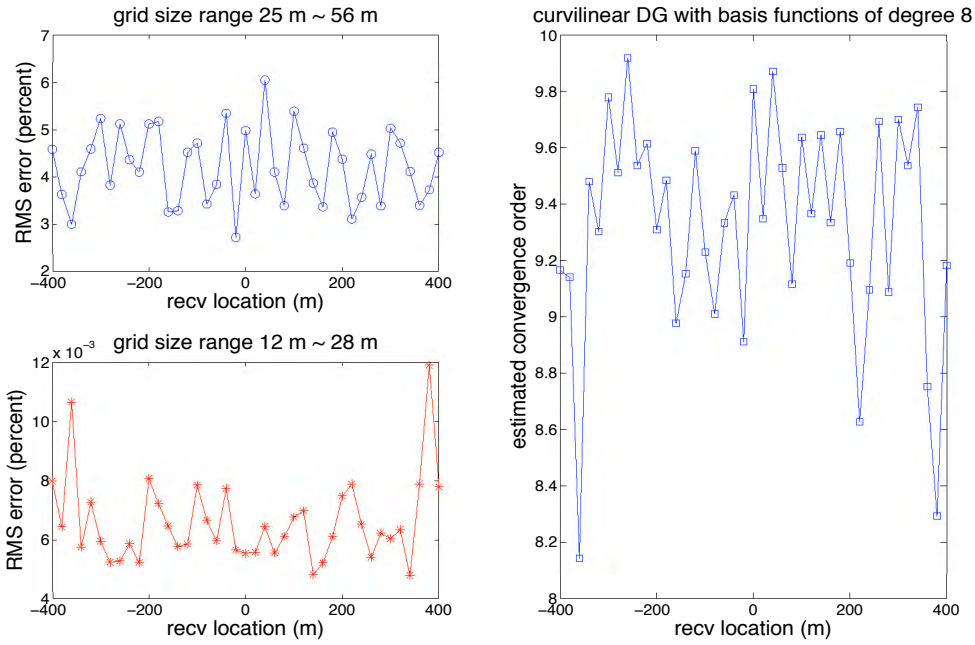

Figure 4.3: The RMS errors and estimated convergence rates for each receiver when using the curvilinear DG method with basis functions of degree 8 on the square-circle model. All the traces are sampled at temporal interval of 5 ms.

I also test this method on the 2D dome model. This time I use 301 geophones (receivers) at the depth 20 m with offset from 100 m to 6100 m at interval 20 m. Three meshes for the 2D dome model are used with grid size range $21 \sim 58$ m, $10.66 \sim 29.26$ m $5.34 \sim 14.62$ m, respectively. The basis functions are polynomials of degree 5.

Fig.(4.4) shows the RMS errors and estimated convergence rates of each trace when the curvilinear DG method is applied to the 2D dome model. The RMS errors on the mesh with grid size range $21 \sim 58$ m are relatively large at the far left receivers and the receivers near the source location. PML might be responsible for the large errors at the far left receivers. When using the mesh with grid size range $21 \sim 58$

m, it's very possible that some receivers near (3300 m, 40 m) are in the same triangle with the source. The approximation error of the Dirac delta function at the beginning of the simulation may affect the numerical accuracy of traces at those receivers.

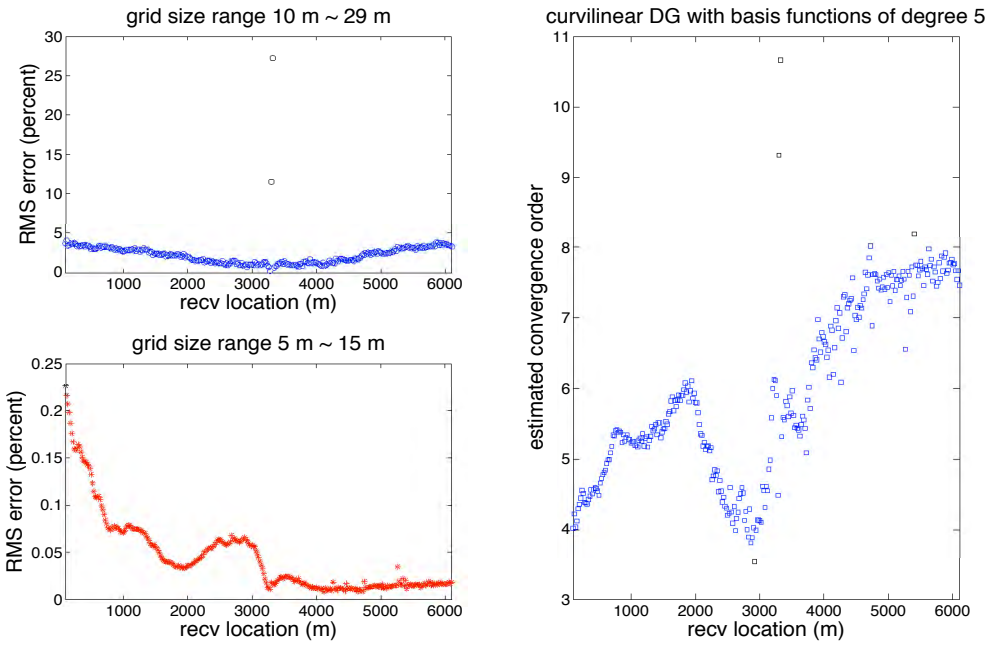

Figure 4.4: The RMS errors and estimated convergence rates for each receiver when using the curvilinear DG method with basis functions of degree 5 on the 2D dome model. All the traces are sampled at temporal interval of 2 ms.

Chapter 5

Discussion and Conclusion

Due to the very important applications, various methods have been developed to simulate the seismic wave propagation. Because of the relatively easy implementation and the desirable balance between the computation cost and the numerical accuracy, FDTD methods on uniform Cartesian grids have become an industry standard in seismic community. DGTD methods have been applied to a wide range of hyperbolic problems. Their successes in those problems encourage me to apply DGTD methods to seismic wave simulation.

The interface error due to the heterogeneity of the model is inevitable in staggered-grid FDTD methods, because several grids are employed. As shown in numerical examples, this error as a time shift effect eventually reduces the convergence rate of FDTD methods to 1st order. DGTD methods somehow remedy the interface error by using the interface-fitting mesh and achieve 2nd order convergence rate when the curved material interfaces are presented. Based on the results of 2D dome model in Tab.3.2, by extrapolation the DGTD method with basis functions of degree 2 on a piecewise linear interface-fitting mesh pays 2.24e+4 GFLOP for 5% RMS error,

while the 2-4 staggered-grid FDTD method has to use 0.23 m grid and pays 1.65e+6 GFLOP for the same accuracy. As the simulation time increases, unbearable small grid size in FDTD methods is required to achieve a given level of accuracy.

More interestingly, if the precise geometry information is provided, we can fit the material interfaces and/or the boundaries with the curvilinear elements, which complement the accuracy of the high order DG solver. The curvilinear DGTD method hence converges at the optimal convergence rate in the numerical experiments with a modest increasing storage in our low-storage curvilinear DG method.

The interface-fitting mesh generation is nontrivial, especially when the structure of the model is complicated and many scales of the materials coexist. Unfortunately, the geological models usually have these features. So I propose the local mesh refinement technique as an alternative for DGTD methods to reduce the interface error as shown in the numerical example. This technique handles different models under the same procedure,

1. start with an initial triangulation $\{\mathcal{T}_k\}_k$, $\Omega = \bigcup_k \mathcal{T}_k$
2. compute the material contrast indicator $\mathcal{I}_k$ on $\mathcal{T}_k$
3. if $\mathcal{I}_k >$ threshold and the grid size of $\mathcal{T}_k > h_t$, refine $\mathcal{T}_k$
4. assemble the new mesh $\{\tilde{\mathcal{T}}_{\tilde{k}}\}_{\tilde{k}}$ that satisfies $\Omega = \bigcup_{\tilde{k}} \tilde{\mathcal{T}}_{\tilde{k}}$

A very coarse mesh is generated first. Then the mesh is locally refined according to the material contrast indicator so as to decrease the element size near the interface. Since the spatial step is determined by the slowest velocity, different element sizes should be applied to different materials. This can be done by adding the sound velocity as a weight when computing the indicators. In this way an optimal mesh for

a given model can be generated through the local mesh refinement process. The only problem is that small elements can lead to small time step, which can increase the overall computation cost.

In summary, this thesis formulates and implements regular and curvilinear DGTD methods for the acoustic wave equations (pressure-velocity formulation) in heterogeneous media. The regular DGTD method achieves as much as 2nd order convergence rate in both square-circle model and 2D dome model while the curvilinear DGTD method performs the optimal convergence rate. The interface error in the square-circle model is well resolved by FDTD with less computation cost. But when the model becomes more complex like the 2D dome model and the simulation time becomes longer, the regular DGTD method on a piecewise linear interface-fitting mesh is more efficient.

BIBLIOGRAPHY

Abarbanel, S. and Gottlieb, D., 1998, On the construction and analysis of absorbing layers in CEM: Appl. Numer. Math., **27**, 331–340.

Ainsworth, M., 2004, Dispersive and dissipative behavior of high-order discontinuous Galerkin finite element mehtods: Journal of Computational Physics, **198**, 106–130.

Alford, R. M., Kelly, K. R., and Boore, D. M., 1974, Accuracy of finite-difference modeling of the acoustic wave equation: Geophysics, **39**, 834–842.

Alterman, Z. and Karal, F. C., 1968, Propagation of elastic waves in layered media by finite difference methods: Bull. Seismol. Soc. Am., **58**, 367–398.

Atkins, H. and Shu, C.-W., 1998, Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations: AIAA J., **36**, 775–782.

Bassi, F. and Rebay, S., 1997, A high-order discontinuous Galerkin finite element method solution of 2d euler equations: Journal of Computational Physics, **138**, 251–285.

Bayliss, A., LeMesurier, K. E., and Turkel, B. J., 1986, A fourth-order accurate finite-difference scheme for the computation of elastic waves: Bull. Seismol. Soc. Am., **76**, 1115–1132.

Bedford, A. and Drumheller, D. S., 1994, Introduction to elastic wave propagation: Wiley.

Berenger, J. P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: Journal of Computational Physics, **114**, 185–200.

Boore, D. M., 1970, Love waves in non-uniform wave guides: finite difference calculations: Geophys. Res., **75**, 1512–1527.

———, 1972, Finite difference methods for seismic wave propagation in heterogeneous materials: Chapter 1 in Methods in Computational Physics, II.

Brown, D. L., 1984, A note on the numerical solution of the wave equation with piecewise smooth coefficients: Mathematics of Computation, **42**, 369–391.

Carpenter, M. H. and Kennedy, C., 1994, Fourth-order 2n-storage Runge-Kutta schemes: NASA Report TM 109112, NASA Langley Research Center.

Chavent, G. and Salzano, G., 1982, A finite element method for the 1d water flooding problem with gravity: Journal of Computational Physics, **45**, 307–344.

Cockburn, B., Karniadakis, G. E., and Shu, C.-W., 1999, The development of discontinuous Galerkin methods: http://purl.umn.edu/3384.

Cockburn, B., Li, F., and Shu, C.-W., 2004, Locally divergence-free discontinuous Galerkin methods for the Maxwell equations: Journal of Computational Physics, **194**, 588–610.

Cockburn, B. and Shu, C.-W., 1989, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework: Math. Comp., **52**, 411–435.

Cohen, G., Ferrieres, X., and Pernet, S., 2006, A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell's equations in the time-domain: Journal of Computational Physics, **217**, 494–500.

Cohen, G. C., 2002, Higher order numerical methods for transient wave equations: Springer-Verlag, New York.

Dablain, M. A., 1986, The application of high-order differencing to the scalar wave

equation: Geophysics, **51**, 54–66.

Giraldo, F. X., Hesthaven, J. S., and Warburton, T., 2002, Nodal high-order discontinuous Galerkin method for the spherical shallow water equations: Journal of Computational Physics, **181**, 499–525.

Gordon, W. J. and Hall, C. A., 1973, Transfinite element methods: blending-function interpolation over arbitrary curved element domains: Numerische Mathematik, **21**, 109–129.

Hesthaven, J. S. and Warburton, T., 2002, Nodal high-order methods on unstructured grids i. time-domain solution of Maxwell's equations: Journal of Computational Physics, **181**, 186–221.

———, 2008, Nodal discontinuous Galerkin methods: algorithms, analysis and applications: Springer.

Johnson, C. and Pitkaranta, J., 1986, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation: Math. Comp., **46**, 1–26.

Käser, M. and Dumbser, M., 2006, An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes I: The two-dimensional isotropic case with external source terms: Geophys. J. Int., **166**, 855–877.

Kelly, K. R., Ward, R. W., Treitel, S., and Alford, R. M., 1976, Synthetic seismograms: a finite-difference approach: Geophysics, **41**, 2–27.

Lesaint, P. and Raviart, P. A., 1974, On a finite element method for solving the neutron transport equation: In mathematical aspects of finite elements in partial differential equations, Academic Press, New York, 89–145.

Leveque, R. J., 2002, Finit volume methods for hyperbolic problems: Cambridge University Press.

Madariaga, R., 1976, Dynamics of an expanding circular fault: Bull. Seismol. Sco. Am., **67**, 163–182.

Mitchell, A. R. and Griffiths, D. F., 1994, The finite difference method in partial differential equations: Wiley & Sons, New York.

Reed, W. H. and Hill, T. R., 1973, Triangular mesh methods for the neutron transport equation: Los Alamos Scientific Laboratory Report, **LA-UR-73-479**.

Regone, C., 2007, Using 3d finite-difference modeling to design wide azimuth surveys for improved subsalt imaging: Geophysics, **72**, SM231–SM239.

Richtmyer, R. and Morton, K., 1967, Difference methods for initial-value problems: Wiley & Sons, New York.

Sjögren, M., 2002, Comparison of spectral element and finit difference methods for electromagnetic wave propagation over a material discontinuity: FOI-R-0489-SE, Swedish Defence Research Agency.

Symes, W. W., 2003, Mathematics of seismic imaging: http://www.trip.caam.rice.edu/downloads/summer/sc_im.pdf.

Symes, W. W. and Vdovina, T., 2009, Interface error analysis for numerical wave propagation: Comput. Geosci., **13**, 363–371.

Terentyev, I., Symes, W. W., and Vdovina, T., 2008, Iwave: The Rice Inversion Project, Rice University.

Toro, E. F., 1997, Riemann solvers and numerical methods for fluid dynamics: Springer.

Virieux, J., 1984, SH-wave propagation in heterogeneous media: velocity-stress finite-difference method: Geophysics, **49**, 1933–1957.

———, 1986, P-sv wave propagation in heterogeneous media: velocity-stress finite-difference method: Geophysics, **51**, 889–901.

Warburton, T., 1999, Application of the discontinuous Galerkin method to Maxwell's equations using unstructured polymorphic hp-finite elements: Lecture notes in computational science and engineering **11**, Springer-Verlag, Berlin, 451–458.

———, 2006, An explicit construction of interpolation nodes on the simplex: Journal of Engineering Mathematics, **56**, 247–262.

Yee, K. S., 1966, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media: IEEE transactions on antennas and propagation, **AP-14**, 302–307.