2014

Numerical superposition of Gaussian beams over propagating domain for high frequency waves and high-order invariant-preserving methods for dispersive waves

Nattapol Ploymaklam
Iowa State University

Numerical superposition of Gaussian beams over propagating domain for high frequency waves and high-order invariant-preserving methods for dispersive waves

by

Nattapol Ploymaklam

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Applied Mathematics

Program of Study Committee:

Hailiang Liu, Major Professor

Lisheng Steven Hou

Songting Luo

Sung Yell Song

Zhijun Wu

Iowa State University

Ames, Iowa

2014

DEDICATION

I would like to dedicate this thesis to my mother Wipha Bunsong without whose sacrifice and support I would not have been able to come this far. I would also like to thank my family and friends for their loving guidance and moral support during the process of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis.

First and foremost, I would like to express my deepest gratitude to Professor Hailiang Liu, my academic advisor, for his guidance, patience, and support throughout my graduate study. His keen academic insight has always been inspiring me throughout the completion of this dissertation and beyond. His deep life lessons have made me a much better person than before. Cooperating with his other students, Dr. Zhongming Wang and Dr. Hui Yu, has also contributed to the progress of my research greatly.

I am also very grateful to my committee members for their efforts and contributions to this work: Professor L. Steven Hou, Professor Songting Luo, Professor Sung Yell Song, and Professor Zhijun Wu.

I also want to thank all the faculty members, colleagues, friends, and staffs in the Department of Mathematics at Iowa State University for their supports and assistances.

Last but not least, I would like to thank the government of my home country, Thailand, for supporting my study in the United States. Special thanks go to the Institute for the Promotion of Teaching Science and Technology (IPST) which awarded me the Development and Promotion of Science and Technology Talents (DPST) scholarship.

ABSTRACT

This thesis is devoted to efficient numerical methods and their implementations for two classes of wave equations. The first class is linear wave equations in very high frequency regime, for which one has to use some asymptotic approach to address the computational challenges. We focus on the use of the Gaussian beam superposition to compute the semi–classical limit of the Schrödinger equation. The second class is dispersive wave equations arising in modeling water waves. For the Whitham equation, so-called the Burgers–Poisson equation, we design, analyze, and implement local discontinuous Galerkin methods to compute the energy conservative solutions with high-order of accuracy.

Our Gaussian beam (GB) approach is based on the domain-propagation GB superposition algorithm introduced by Liu and Ralston [Multiscale Model. Simul., 8(2), 2010, 622–644]. We construct an efficient numerical realization of the domain propagation-based Gaussian beam superposition for solving the Schrödinger equation. The method consists of several significant steps: a semi-Lagrangian tracking of the Hamiltonian trajectory using the level set representation, a fast search algorithm for the effective indices associated with the non-trivial grid points that contribute to the approximation, an accurate approximation of the delta function evaluated on the Hamiltonian manifold, as well as efficient computation of Gaussian beam components over the effective grid points. Numerical examples in one and two dimensions demonstrate the efficiency and accuracy of the proposed algorithms.

For the Burgers–Poisson equation, we design, analyze and test a class of local discontinuous Galerkin methods. This model, proposed by Whitham [Linear and Nonlinear Waves, John Wiley & Sons, New York, 1974] as a simplified model for shallow water waves, admits conservation of both momentum and energy as two invariants. The proposed numerical method is high order accurate and preserves two invariants, hence producing solutions with satisfying long time behavior. The L^2-stability of the scheme for general solutions is a consequence of the energy

preserving property. The optimal order of accuracy for polynomial elements of even degree is proven. A series of numerical tests is provided to illustrate both accuracy and capability of the method.

CHAPTER 1. INTRODUCTION

This thesis features two main objectives: (i) recovery of high frequency wave fields, and (ii) accurate computation of dispersive waves. We use the Gaussian beam method to recover the highly oscillatory wave fields, and we use the local discontinuous Galerkin method to approximate the dispersive waves so that some invariants are preserved at the discrete level.

Computing high frequency wave fields using Gaussian beams (GB) has received much attention because of its validity at caustics. Despite this much attention, many problems remain to be solved. The domain propagation based Gaussian beam superposition developed by Liu and Ralston (2010) provides a quantitative recovery. Our main task is to numerically implement this recovery theory, with a focus on GB solutions to the Schrödinger equation. In doing so, we propose a search algorithm that captures an interface within a moving domain, modify an existing approximation of the delta function to fit into our problem, and adapt the semi-Lagrangian method to approximate the GB components and the level set function representing the moving interface.

We also design a local discontinuous Galerkin (LDG) method for the Burgers-Poisson equation. The scheme conserves the momentum and energy of the numerical solution and is proven to be of optimal order for polynomials with even degree. In the error estimation, we introduce a global projection and derive some of its properties to use as an estimation tool.

1.1 Organization

This thesis is organized as follows. Chapters 2-4 are devoted to the recovery of high frequency wave fileds. In Chapter 2, we introduce a new search algorithm to trace the moving domain in order to reduce the computational cost. We present examples in two dimensional

plane to illustrate how the algorithm performs. In Chapter 3, we adapt an approximation of the delta function so that it can be effectively used to numerically recover the high frequency wave fields. In Chapter 4, we apply our algorithm to computing the semi-classical limit of the Schrödinger equation, and discuss the approximation of the GB components and the level set representation. Examples in one and two dimensions are presented to verify the accuracy of the numerical implementation.

Chapters 5-6 are devoted to the LDG method for solving the Burgers–Poisson equation. In Chapter 5, we formulate our LDG method so that the method has some desired properties through choices of numerical fluxes. We then show that the LDG method for solving the Poisson equation (1.4b) is well defined and stable, and the method is shown to conserve both momentum and energy for the conservative numerical fluxes. In Chapter 6, we obtain the optimal order of error between the numerical solution and smooth solutions for the conservative scheme when using polynomial elements of even degree. Then, we present numerical examples to illustrate the capacity of the LDG scheme to preserve two invariants after long-time simulation.

We use the rest of this chapter to discuss the general background for the recovery of high frequency wave fields for the Schrödinger equation and the LDG method for the Burgers–Poisson equation.

1.2 Recovery of high frequency wave fields

We consider the linear equation

$$-i\epsilon\partial_t\psi + H(x, -i\epsilon\partial_x)\psi \quad = \quad 0, \quad (t,x) \in \mathbb{R}\times\mathbb{R}^n, \tag{1.1}$$

subject to the highly oscillatory initial data

$$\psi(0,x) \quad = \quad A_{in}(x)e^{iS_{in}(x)/\epsilon}, \tag{1.2}$$

where $A_{in} \in C_0^\infty(\mathbb{R}^n)$ and $S_{in} \in C^\infty(\mathbb{R}^n)$. The example we will focus on is the Schrödinger equation with $H(x,p) = V(x) + \frac{|p|^2}{2}$. The small parameter ϵ represents the fast space and time scale introduced in the equation, as well as the typical wavelength of oscillations of the initial data. Propagation of oscillations of wavelength $O(\epsilon)$ causes mathematical and numerical

challenges in solving the problem. To solve (1.1) directly using finite difference method, one needs to use a mesh size and time step that are of order $O(\epsilon)$ (see Markowich et al. (1999, 2002)) which is expensive when ϵ is small. The classical remedy for this is the geometric optic method which looks for an asymptotic solution of the form

$$\psi^\epsilon(t,x) = A(t,x)e^{\frac{i}{\epsilon}\phi(t,x)}. \tag{1.3}$$

One can plug-in this ansatz into (1.1) and get the Hamilton–Jacobian equation for the phase ϕ and the transport equation for A,

$$\begin{aligned}
&\partial_t\phi + H(x,\nabla_x\phi) = 0, \quad x \in \mathbb{R}^n, t > 0,\\
&\partial_t|A|^2 + \nabla\cdot(\nabla\phi|A|^2) = 0,
\end{aligned}$$

which develops kink singularity in ϕ at finite time, where $|A|$ becomes unbounded, therefore unphysical. The remedy for this is the Gaussian beam method which assumes that the phase ϕ can be complex away from a central ray determined by the geometrical optics approach (Ralston (1982)). The superposition of Gaussian beams of the form (1.3) yields an asymptotic solution at any time including at the formulation of caustic. However, the summation of the beams takes place on the whole Cartesian plane. Liu and Ralston (2010) develops the superposition formulation where the asymptotic solution is taken from the summation of the Gaussian beams over a propagating domain which evolves along a Hamiltonian flow. This resolves the problem on where to take the summation.

In this thesis, we develop a numerical method to implement the superposition of the Gaussian beams introduced by Liu and Ralston (2010). The main difficulty of this approach is how to trace the propagating domain. We develop the search algorithm that captures the propagating domain so the calculation can be done on a reduced number of grids. The search algorithm can be generalized to other situations where you have a region that moves along a given velocity as time goes by. It can also be generalized to a higher-dimensional setting.

Other difficulty for the superposition formulation is the approximation of the delta function in multidimensional setting. We develop a new approach for delta approximation in one dimensional setting, then adapt the existing delta approximation for the surface integral in

two dimensional setting into our simulation. Another task is to efficiently compute the GB component of the ansatz (1.3) at a given grid point at any time τ, for which we use the semi-Lagrangian method.

The approach for the superposition of Gaussian beams we introduce here can be implemented for the problems in higher dimension. We illustrate the results in one and two dimensions as examples.

1.3 Local Discontinuous Galerkin method for Burger-Poisson equation

We are interested in numerical approximations to the Burgers-Poisson (BP) equation of the form

$$u_t + \left(\frac{u^2}{2} - \phi\right)_x = 0, \tag{1.4a}$$

$$\phi_{xx} - \phi = u. \tag{1.4b}$$

The subscript t (or x, respectively) denotes the differentiation with respect to time variable t (or spatial variable x), where u and ϕ depend on $(t, x) \in (0, \infty) \times \mathbb{R}$. System (1.4) can be rewritten as a nonlocal equation

$$u_t + \left(\frac{u^2}{2} + G * u\right)_x = 0 \tag{1.5}$$

with the kernel $G(x) = \frac{1}{2}e^{-|x|}$. This nonlocal model was found as a simplified shallow water model by Whitham Whitham (1974) to approximate the model with a singular kernel

$$G(x) = \frac{1}{2\pi}\int_{\mathbb{R}} \left(\frac{\tanh k}{k}\right)^{1/2} e^{ikx} dk.$$

For (1.5) with initial data $u_0 \in BV(\mathbb{R})$, it is shown in Fellner and Schmeiser (2004) that there exists a unique global weak solution $u \in L^\infty_{\text{loc}}([0, \infty); BV(\mathbb{R}))$. For smooth initial data $u_0 \in H^s(\mathbb{R})$ with $s > 3/2$, there exists a unique smooth solution $u \in L^\infty((0, T); H^s(\mathbb{R})) \cap C((0, T); H^{s-1}(\mathbb{R}))$, at least for some finite time T. Furthermore, analysis of traveling waves in Fellner and Schmeiser (2004) shows that there are three generic cases of wave patterns, including solitary waves, peaked periodic waves, and shock waves, and the set of pairs $(u_{-\infty}, u_\infty)$ can be connected by a shock wave only when $u_\infty - u_{-\infty} \geq 2$.

In this thesis , we develop a local discontinuous Galerkin (LDG) method to solve this nonlinear BP equation with initial data $u_0(x)$, posed on a bounded domain $[0, L]$, with periodic boundary conditions. For other type of boundary conditions, the method can be modified to incorporate the specified boundary condition through suitable boundary numerical fluxes, while still using the conserved numerical fluxes for other cell interfaces.

Our proposed scheme is high order accurate, and preserves two invariants of momentum and energy, hence producing solutions with satisfying long time behavior. The L^2-stability of the scheme for general solutions is a consequence of the energy preserving property.

In the context of water waves, one of the best known local models is probably the Korteweg de Vries (KdV)-equation,

$$u_t + uu_x + u_{xxx} = 0.$$

This equation possesses soliton solutions' coherent structures that interact nonlinearly among themselves, then reemerge, retaining their identity and showing particle-like scattering behavior. In shallow water wave theory, the nonlinear shallow water equations which neglect dispersion altogether lead to the finite time wave breaking. On the other hand the third order derivative term in the KdV equation will prevent this ever happening in its solutions. In reality, some water waves appear to break, if the wave height is above certain threshold. Therefore in Whitham (1974), an intriguing question was raised: what kind of mathematical equation can describe waves with breaking? He suggested equation (1.5) with the above two kernels; many competing models have since been suggested to capture one aspect or another of the classical water-wave problem, see e.g., Fuchssteiner and Fokas (1982); Camassa and Holm (1993); Camassa et al. (1994); Johnson (2002); Degasperis et al. (2002); Holm and Staley (2003); Liu and Yin (2006); Constantin and Lannes (2009); Liu and Yin (2010).

One common feature of these models is the associated global invariants, infinitely many or finitely many. The BP equation preserves both the momentum and the energy, that is, it has the following two global invariants

$$\int u(0,x)\,dx = \int u(t,x)\,dx =: E_1(t), \tag{1.6a}$$

$$\int u^2(0,x)\,dx = \int u^2(t,x)\,dx =: E_2(t). \tag{1.6b}$$

It is desirable to design stable and high order accurate numerical schemes which preserve two invariants for solving the BP equation. It is believed that numerical methods preserving more invariants are advantageous: besides the high accuracy of numerical solutions, an invariant preserving scheme can preserve good stability properties after long-time numerical integration. Much more effort has been devoted in this topic for different integrable PDEs recently, e.g., Furihata and Mori (1998); McLachlan et al. (1999); Matsuo (2008); Celledoni et al. (2012).

The goal of this thesis is to develop a discontinuous Galerkin (DG) method to preserve both momentum and energy at the discrete setting. The DG method is a class of finite element methods using completely discontinuous piecewise-polynomials for the numerical solution and the test functions. It was first designed and has been successful for solving first order PDEs such as nonlinear conservation laws, Reed and Hill (1973); Cockburn et al. (1989); Cockburn and Shu (1989); Cockburn et al. (1990); Cockburn and Shu (1998b). The local discontinuous Galerkin (LDG) method is an extension of the DG method for solving higher order PDEs. It was first designed for convection-diffusion equations in Cockburn and Shu (1998a), and has been extended to other higher order wave equations, including the KdV equation, Yan and Shu (2003); Xu and Shu (2005); Liu and Yan (2006); Xu and Shu (2007) and the Camassa-Holm equation, Xu and Shu (2008), see also the recent review paper by Xu and Shu (2010) on the LDG methods for higher order PDEs. The idea of the LDG method is to rewrite higher order equations into a first order system, and then apply the DG method on the system. In contrast, the direct discontinuous Galerkin (DDG) methods, proposed in Liu and Yan (2009, 2010) primarily for diffusion equations, aimed at directly solving higher order PDEs by the DG discretization, see e.g., Bona et al. (2013); Yi et al. (2013) for energy preserving DG methods for KdV type equations, and Liu et al. (2014) for the Degasperis-Procesi equation. The DDG method, as another class of DG methods for higher order partial differential equations, is to directly force the weak solution formulation of the PDE into the DG function space for both the numerical solution and test functions. Unlike the traditional LDG method, the DDG method does not rewrite the original equation into a larger first order system. The main novelty in the DDG schemes proposed in Liu and Yan (2009, 2010) lies in numerical flux choices for the solution gradient, which involves higher order derivatives evaluated crossing cell interfaces.

In this thesis we propose an LDG method based on formulation (1.4), for which the second equation was rewritten into a first order system for applying the LDG discretization. In the algorithm we update the solution in two steps: (1) given u, obtain ϕ by solving (1.4b) with the LDG method; (2) with the obtained ϕ, update u by solving (1.4a) with a standard DG method using a conservative numerical flux so that the resulting scheme preserves two integrals E_1 and E_2 in smooth region.

As for error estimates, we define a global projection dictated by the selected numerical flux and obtain the needed projection error, following the strategy of error estimates carried out in Liu (2014) for the DDG method to solve convection-diffusion equations. Through careful estimates using this global projection, we obtain the optimal order of accuracy for polynomial elements of even degree. This is confirmed by the numerical tests with $k = 2, 4$. Numerical tests also show that for k odd, only k-th order of accuracy is observed. Such an optimal error estimate only for k =even was also shown in Bona et al. (2013), and numerically observed in Bona et al. (2013); Yi et al. (2013) for KdV type equations. The main feature of the scheme presented in this work is its capability to produce wave solutions with satisfying long time behavior.

We want to point out that our estimates apply only for smooth solutions. However, for some initial configuration, the BP equation may admit discontinuous solutions at finite time, and beyond that time weak solutions need to be considered. The question is that in what sense our high order LDG methods mean for weak solutions in large times. Some rigorous $C_t^0(L_x^1)$ estimate would be desirable to understand this issue. A recent example of this type of estimates can be found in Amadori and Gosse (2013) for well-balance schemes on non-resonant scalar balance laws.

CHAPTER 2. INTERFACE TRACKING

2.1 Introduction

2.1.1 General background

The domain propagation based Gaussian beam superposition is of the following form

$$\psi(t,y) = \int_{\Omega(t)} \Psi(t,y,X)\delta(w(t,X))dX,$$

where Ψ is obtained from solving the ODEs for GB components, w is solved from the level set equation which is used to capture the interface, and $\Omega(t)$ is the moving domain driven by the Hamiltonian. In order to implement this recovery formula, there is a need to track $\Omega(t)$ and $w(t,X)$. In this chapter, we discuss a new search algorithm that efficiently captures the effective computational cells required for the above recovery.

There are two main approaches for the interface tracking in physical space: Lagrangian approach which tracks the particles that marks the interface, and the Eulerian method which uses a level set function to represent the evolving interface.

The Lagrangian approach was first used by Peskin (1977) to approximate the flow of blood in heart using the immersed boundary method. The muscular heart wall was considered a moving immersed boundary which interacts with fluid. When using particle markers to capture the interface, one needs a reconstruction technique to recover the interface from the set of finite markers.

On the other hand, the Eulerian approach was introduced by Osher and Sethian (1988) to approximate equations of propagating fronts whose speeds depend on the local curvatures. The moving interface is described by the zero level set of a continuous function that changes depending on the motion of the interface. Because the interface is embedded in the zero level

set of a function, the dimension of the computational domain increases by one. The trade-off is that the topological merging and breaking are well defined and easily performed.

Interface tracking in phase space has not been well developed. In this chapter, we propose a new search algorithm which utilizes a combination of both approaches carried out in phase space. The algorithm uses one particle marker to find the interface. It then searches for the rest of the interface using the level set functions representing the interface.

2.1.2 Problem formulation

Let $X = (x, p) \in \mathbb{R}^2$. Given a C^1 function $S_{in}(x)$ and a simply-connect, bounded region $\Omega(0) \subset \mathbb{R}^2$, we consider level set functions $\psi(t, X), w(t, X)$ satisfying

$$\begin{aligned} \mathcal{L}[\psi] &= 0, \\ \psi(0, X) &= \begin{cases} 0 & \text{if } X \in \partial\Omega(0) \\ d(X; \Omega(0)) & \text{otherwise,} \end{cases} \end{aligned}$$

and

$$\begin{aligned} \mathcal{L}[w] &= 0, \\ w(0, X) &= p - \partial_x S_{in}(x), \end{aligned}$$

where

$$\mathcal{L} := \partial_t + p\partial_x - V'(x)\partial_p, \quad \text{for some } V(x) \in C^1.$$

Here, $d(X; A)$ is the distance function for a set $A \subset \mathbb{R}^2$ defined so that $d(X) < 0$ if $X \in A$ and $d(X) > 0$ if $X \in A^c$, where A^c is the complement of A.

Our goal, at a fixed time $t = T$, is to find the effective index set $\mathcal{G} = \mathcal{G}(T)$, which is defined by

$$\{X \in \mathbb{R}^2 \,|\, \psi(t, X) \le 0, w(t, X) = 0\} \subseteq \bigcup_{(j,k)\in\mathcal{G}} I_{j,k},$$

where $I_{j,k} := [x_j, x_{j+1}] \times [p_k, p_{k+1}]$ with $x_j := \Delta x \cdot j$ and $p_k := \Delta p \cdot k$. In other words, define the sets Γ and Ω as

$$\Gamma(t) := \{X \in \mathbb{R}^2 \,|\, w(t, X) = 0\},$$
$$\Omega(t) := \{X \in \mathbb{R}^2 \,|\, \psi(t, X) \leq 0\}.$$

Then, the effective index set $\mathcal{G}(T)$ collects all the grid points near $\Gamma(T)$ that are inside of $\Omega(T)$.

2.2 Algorithm

The search algorithm discussed here is for $X \in \mathbb{R}^2$, but it can be readily generalized to higher dimension (see, for example, section 2.3). The idea for the search goes as follows: (1) Locate one cell that contains the interface Γ inside of Ω. (2) Find its neighboring cells that also contain the interface. (3) Find neighboring cells of the neighboring cells that contains the interface (that haven't been found previously). The process continues until we can no longer find any more new neighboring cells containing the interface. The termination of the process is possible because the domain Ω is bounded. We explain the technique algorithmically below.

2.2.1 Search algorithm

We start with $[a, b]$ as an input. Here, $\Omega(0) = [a, b] \times S'_{in}([a, b])$. The output will be the effective index set $\mathcal{G}$. In each step of the search, we explore all neighboring cells of the cell of interest (that is not already in $\mathcal{G}$) to see if any of them intersects with Γ and is in Ω. If there is one, we move on to explore that cell (and call it the cell of interest). If there are more than one of such cells, we choose any one of them to explore and save the others in the waiting-list set Wl so that we can come back to explore it later.

Algorithm 2.2.1

1. Find first cell to begin the search by locating any cell that intersects Γ. For instance, we take $\mathcal{X}^0 = (x^0, p^0)$ where $x^0 = (a + b)/2$ and $p^0 = \partial_x S_{in}(x^0)$. Then, we use a cell that contains $\mathcal{X}^N = \Theta^N(\mathcal{X}^0)$, where Θ is a one-step ODE solver, as starting cell. We add that cell to $\mathcal{G}$ and call it the cell of interest.

2. Explore all neighboring cells of the cell of interest. Identify the neighbors that intersect with the interface that are not already in $\mathcal{G}$. Call them the set of neighborhood Nbh and add them to $\mathcal{G}$. (It is possible that Nbh is empty.) Note: this step requires algorithm 2.2.2 below.

3. If Nbh contains exactly one element, it is a new cell of interest. If Nbh contains more than one element, we take one of them as a new cell of interest and add the others to the waiting-list set Wl. If Nbh is empty, we take one element from Wl as a new cell of interest.

4. Repeat steps 2-3 until both Nbh and Wl are empty. (i.e. no more cell to explore.)

2.2.2 Find neighboring cells

Suppose we are at the cell of interest $I_{j,k}$. (Here, $I_{j,k} := [x_j, x_{j+1}] \times [y_k, y_{k+1}]$ with $x_j := \Delta x \cdot j$, $y_k := \Delta y \cdot k$.) We find the neighboring cells that intersect Γ as follows:

Algorithm 2.2.2

1. Identify all cells that are adjacent to the cell of interest. In this case, they are labeled as $I_{j\pm1,k\pm1}$, $I_{j,k\pm1}$, and $I_{j\pm1,k}$.

2. Go through each adjacent cell to check if any of them intersects with Γ (using algorithm 2.2.3) and overlaps with Ω (using algorithm 2.2.4). If so, put it in the candidate set C.

3. Check if each member of C is already in $\mathcal{G}$. If so, remove it from C.

4. Set the resulting C to be Nbh, which is the output of the algorithm.

2.2.3 Check intersection

We need to determine if a cell $I_{j,k}$ intersects Γ. If it does and overlaps with Ω, it is an effective cell, i.e. if $I_{j,k} \cap \Gamma \neq \emptyset$ and $I_{j,k} \cap \Omega \neq \emptyset$, then $(j,k) \in \mathcal{G}$. The algorithm below checks if a given cell $I_{j,k}$ intersects Γ.

Algorithm 2.2.3

1. Compute $w_{j',k'}$, $(j',k') \in \mathcal{J} := \{(i,j),(i+1,j),(i,j+1),(i+1,j+1)\}$.
2. If $\left| \sum_{(j',k')\in\mathcal{J}} \text{sign}(w_{j',k'}) \right| \neq 4$, then the cell $I_{j,k}$ intersects with Γ.

2.2.4 Check domain overlap

We also need to determine if part of a cell $I_{j,k}$ is inside Ω. The algorithm below checks if a given cell $I_{j,k}$ overlaps with Ω.

Algorithm 2.2.4

1. Compute $\psi_{j',k'}$, $(j',k') \in \mathcal{J} := \{(i,j),(i+1,j),(i,j+1),(i+1,j+1)\}$.
2. If $\sum_{(j',k')\in\mathcal{J}} \text{sign}(\psi_{j',k'}) < 4$, then the cell $I_{j,k}$ overlaps with Ω.

2.3 Extension to high dimensions

We can generalize the algorithm above to $X = (x,p) \in \mathbb{R}^{2d}$ directly. To illustrate the main ideas, we take $d = 2$. The problem formulation in four dimensional setting, at a fixed time $t = T$, is to locate $\Gamma(T)$ inside of $\Omega(T)$ where

$$\Gamma(t) := \{X \in \mathbb{R}^4 \,|\, w(t,X) = (0,0)^T\},$$
$$\Omega(t) := \{X \in \mathbb{R}^4 \,|\, \psi(t,X) \leq 0\}.$$

Here, the level set functions are defined similarly to the two dimensional case:

$$\mathcal{L}[\psi] = 0, \quad \psi(0,X) = \begin{cases} 0 & \text{if } X \in \partial\Omega(0) \\ d(X;\Omega(0)) & \text{otherwise,} \end{cases}$$
$$\mathcal{L}[w] = 0, \quad w_i(0,X) = p - \nabla_x S_{in}(x),$$

where

$$\mathcal{L} := \partial_t + p \cdot \nabla_x - \nabla_x V(x) \cdot \nabla_p.$$

When implementing the algorithms 2.2.1-2.2.4, the only differences to keep in mind are:

1. The input for the search algorithm is now $D_0 := [a,b]\times[c,d]$. Here, $\Omega(0) = D_0 \times \nabla_x S_{in}(x)$.

2. To begin the search in algorithm 2.2.1, we choose the starting cell from the midpoint $((a+b)/2, (c+d)/2)$.

3. When we go through all the neighboring cells in algorithm 2.2.2, there are $80 = 3^4 - 1$ neighboring cells altogether.

4. In algorithm 2.2.3, we need to check if $\left|\sum_{(J')\in\mathcal{J}} \text{sign}(w^i_{J'})\right| \neq 16$ for $i = 1, 2$. (This is because there are $16 = 2^4$ corner points in each cell.) Here, J' is the 4-dimensional index of the neighboring cells, and $w = (w^1, w^2)$.

5. In algorithm 2.2.4, we need to check if $\sum_{(J')\in\mathcal{J}} \text{sign}(\psi_{J'}) < 16$.

2.4 Illustration of the search

We illustrate how the search algorithm works by showing the first three steps of the search in two simple cases. In the 2D example (figure 2.1), we assume that the interface is a straight line joining $(2,2)$ and $(6,6)$ and that $h = 1$. In the 4D example (figure 2.2), we assume that the projection of the interface onto two axes is a rectangle $[2,6]\times[2,6]$ and that $h = 1$. In both cases, we choose the upper left cell as a cell of interest whenever there are more than one cell to choose from.

2.5 Numerical examples

We use the search algorithm above to trace the propagating domain $\Omega(t)$ when approximating

$$\psi^\epsilon(t,y) := \int_{\Omega(t)} \Psi_{PGB}(t,y,X)\delta(w(t,X))dX, \tag{2.1}$$

which is an asymptotic solution to the Shrödinger equation

$$i\epsilon\partial_t\psi = \frac{\epsilon^2}{2}\Delta\psi - V(y)\psi, \tag{2.2}$$

$$\psi(0,y) = A_{in}(y)\exp(\frac{i}{\epsilon}S_{in}(y)). \tag{2.3}$$

Using the search algorithm above, we can narrow down the computational area for computing ψ^ϵ. Instead of performing a calculation on a large rectangular area on X-plane, we can focus the calculation near the zero set of w inside of $\Omega(T)$ only. We present the interfaces for various examples below. The initial conditions (2.3) for all the examples below have amplitude $A_{in}(x) = e^{-25(x-0.5)^2}$. We take the input $[a, b]$ to be the support of $A_{in}(x)$.

2.5.1 Example 1.

The quadratic potential $V(x) = x^2/2$ with quadratic phase $S_{in}(x) = x^2/2$. Using the search algorithm, we capture the interface $\Gamma(T)$ and the domain $\Omega(T)$ at the initial time and at the time $T = 0.8$, where caustic occurs, in Figure 2.3 below.

2.5.2 Example 2.

The quadratic potential $V(x) = x^2/2$ with non-quadratic phase $S_{in}(x) = -\frac{1}{5}\log\left[2\cosh\left(5(x-0.5)\right)\right]$. The initial interface and the moving interface at times $T = 0, 0.5$ are in Figure 2.4 below.

2.5.3 Example 3.

The lazy potential $V(x) = x^4/12$ with quadratic phase $S_{in}(x) = x^2/2$. The initial interface and the moving interface at times $T = 0, 1$ are in Figure 2.5 below.

2.5.4 Example 4.

The lazy potential $V(x) = x^4/12$ with non-quadratic phase $S_{in}(x) = -\frac{1}{5}\log\left[2\cosh\left(5(x-0.5)\right)\right]$. The initial interface and the moving interface at times $T = 0, 0.5$ are in Figure 2.6 below.

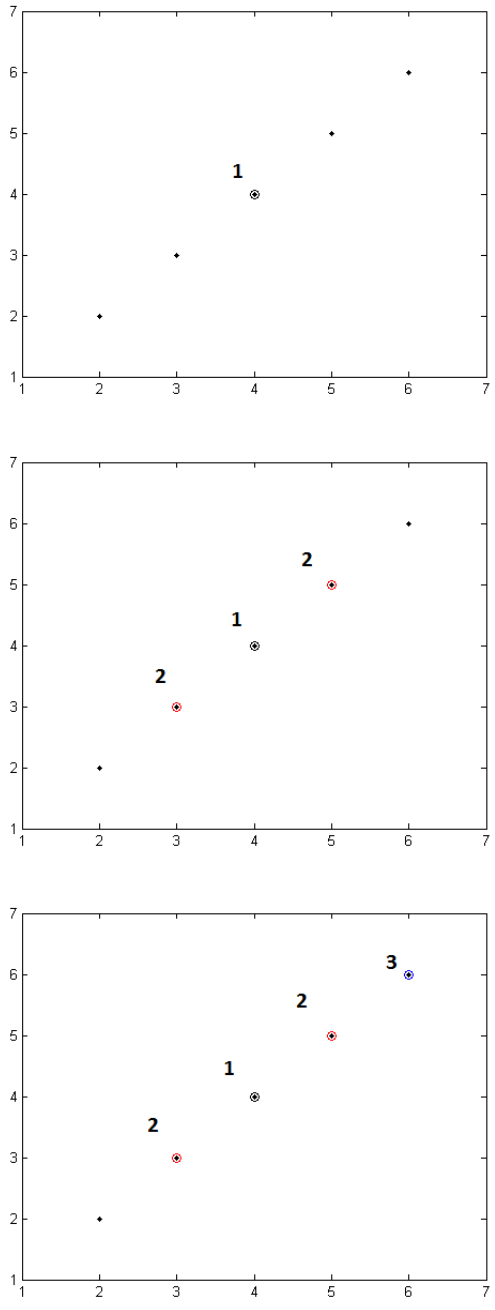

Figure 2.1 Search process in 2D phase space

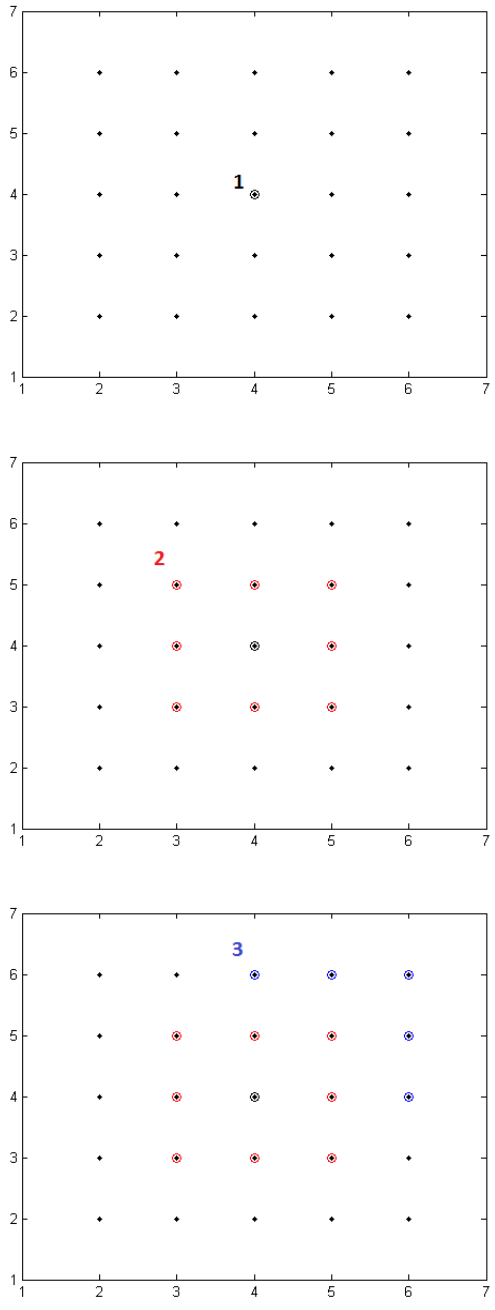

Figure 2.2 Search process in 4D phase space

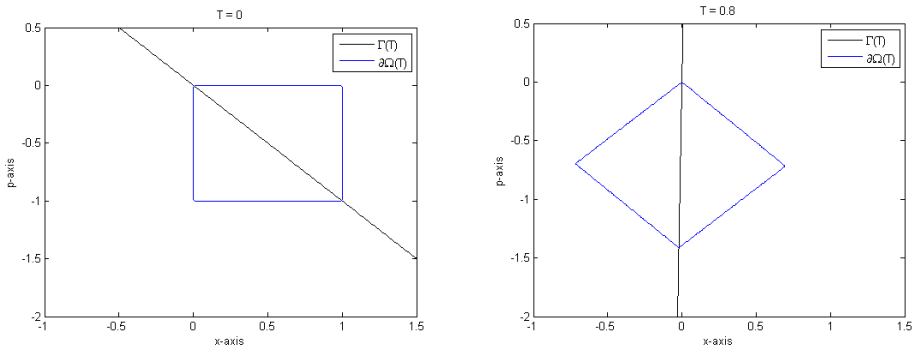

Figure 2.3 Example 1: the zero level set $\Gamma(T)$ and the domain $\Omega(T)$.

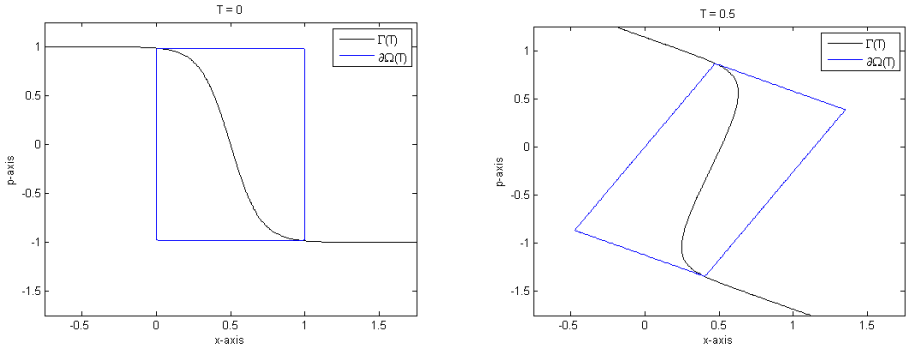

Figure 2.4 Example 2: the zero level set $\Gamma(T)$ and the domain $\Omega(T)$.

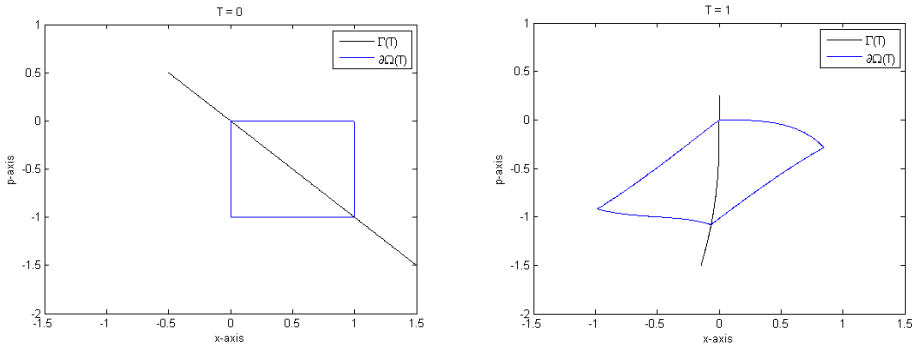

Figure 2.5 Example 3: the zero level set $\Gamma(T)$ and the domain $\Omega(T)$.

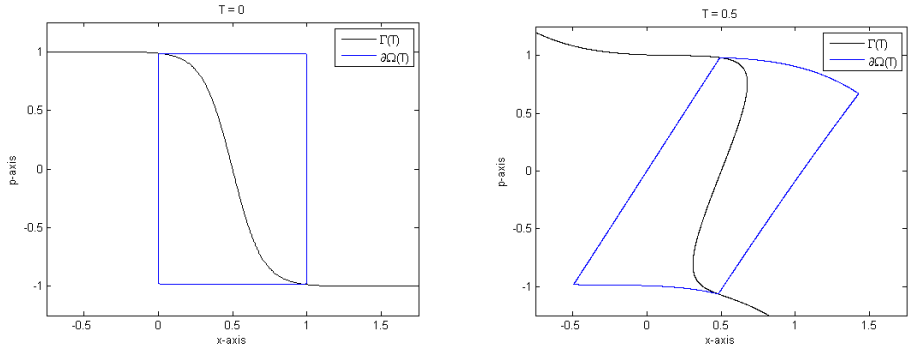

Figure 2.6 Example 4: the zero level set $\Gamma(T)$ and the domain $\Omega(T)$.

CHAPTER 3. DELTA APPROXIMATION

In order to compute ψ^ϵ in (2.1) at any given time, one needs to approximate the integral involving the delta function of the form

$$\int_\Omega \Psi(X)\delta(w(X))dX.$$

We discuss the delta approximation in this section starting from one dimensional case to higher.

3.1 One dimensional cases

In this section, we discuss a method to approximate the integral involving the dirac delta function of the form

$$I = \int_a^b \alpha(x)\delta(\beta(x))dx, \tag{3.1}$$

where $\beta(x)$ is a smooth function having only one zero, c, in (a, b) with $\beta'(c) > 0$, and $\alpha(x)$ is a sufficiently smooth function. Let h represent the mesh size, we use the quadrature rule

$$I_h = h\sum_i \alpha(x_i)\delta_{\omega_i}(\beta(x_i)), \tag{3.2}$$

to approximate I. Here, the discretized delta function is given by $\delta_\omega(x) = \frac{1}{\omega}\phi\left(\frac{x}{\omega}\right)$ for some function ϕ whose support is $[-1, 1]$. We are going to find appropriate choices of ω and ϕ that achieve the desired order of accuracy.

To design a discrete delta function, it is desirable to maintain the properties of the continuous delta function. One of the such properties is the moment condition. We wish to preserve this property in the discrete sense.

Definition 1. *A discrete delta function δ_{mh}, where $m \in \mathbb{R}^+$ is independent of h, satisfies the p^{th} order discrete moment condition if*

$$h \sum_{|x_i - c| \le mh} \delta_{mh}(x_i - c)(x_i - c)^r = \begin{cases} 1 & r = 0, \\ 0 & r = 1, \dots, p-1. \end{cases} \tag{3.3}$$

Now, let us denote

$$y := \beta(x), \quad A(y) := \alpha(\beta^{-1}(y)). \tag{3.4}$$

With notation $y_i := \beta(x_i)$, we define ω_i by

$$\frac{y_i}{\omega_i} = \frac{x_i - c}{mh}. \tag{3.5}$$

This gives

$$\omega_i = \frac{\beta(x_i)}{x_i - c} mh = g(x_i) \cdot mh, \tag{3.6}$$

where

$$g(x) := \int_0^1 \beta'(c + (x - c)\eta)\, d\eta. \tag{3.7}$$

Using (3.5), we can convert the discrete moment condition (3.3) to

$$h \sum_{|y_i| \le \omega_i} \left(\frac{mh}{\omega_i} \right)^{r-1} \delta_{\omega_i}(y_i) y_i^r = \begin{cases} 1 & r = 0, \\ 0 & r = 1, \dots, p-1. \end{cases} \tag{3.8}$$

Now we prove the accuracy of the formula (3.2) where the support size ω_i is given by (3.6). The following lemma is helpful for the proof.

Lemma 3.1.1. *For any positive integer p, there exist constants a_k, $k = 0, 1, \dots, p-1$, such that*

$$A(y_i) = \sum_{k=0}^{p-1} a_k \left(\frac{mh}{\omega_i} \right)^{k-1} y_i^k + R_i^p, \tag{3.9}$$

for all $|y_i| \le \omega_i$, where $R_i = O(h)$.

Proof. Define a function

$$B(x) := \frac{\alpha(x)}{g(x)}, \tag{3.10}$$

which is smooth in the neighborhood of c since $g(c) = \beta'(c) \neq 0$. Note that

$$A(y_i) = g(x_i)B(x_i) = \frac{\omega_i}{mh} B(x_i). \tag{3.11}$$

By Taylor series expansion,

$$B(x_i) = \sum_{k=0}^{p-1} \frac{B^{(k)}(c)}{k!}(x_i - c)^k + \frac{B^{(p)}}{p!}(\xi_i)(x_i - c)^p, \tag{3.12}$$

where ξ_i is between x_i and c. Hence,

$$\begin{aligned} A(y_i) &= \frac{\omega_i}{mh} \sum_{k=0}^{p-1} \frac{B^{(k)}(c)}{k!}(x_i - c)^k + \frac{\omega_i}{mh} \cdot \frac{B^{(p)}}{p!}(\xi_i)(x_i - c)^p, \\ &= \sum_{k=0}^{p-1} \frac{B^{(k)}(c)}{k!}(y_i)^k \left(\frac{\omega_i}{mh}\right)^{k-1} + \frac{B^{(p)}}{p!}(\xi_i)(y_i)^p, \quad \text{by (3.5)}. \end{aligned} \tag{3.13}$$

This leads to (3.9).

□

Theorem 3.1.2. *Let ω_i be given as in (3.6) and let δ_{mh} satisfy the p^{th} order discrete moment condition (3.3). Then, the approximation I_h has p^{th} order of accuracy, i.e.,*

$$|I - I_h| \leq C \cdot h^p, \tag{3.14}$$

where $C = \frac{\|B^{(p)}\|_\infty \|\phi\|_\infty}{p!} \left(2 + \frac{1}{m}\right) m^p$.

Proof. Note that the exact value of I is $\frac{\alpha(c)}{\beta'(c)}$, so

$$\begin{aligned} I_h - I &= h \sum_i \alpha(x_i)\delta_{\omega_i}(\beta(x_i)) - \frac{\alpha(c)}{\beta'(c)} \\ &= h \sum_{|y_i| \leq \omega_i} A(y_i)\delta_{\omega_i}(y_i) - \frac{\alpha(c)}{\beta'(c)} \cdot 1 \\ &= h \sum_{|y_i| \leq \omega_i} A(y_i)\delta_{\omega_i}(y_i) - \frac{\alpha(c)}{\beta'(c)} \cdot \left(h \sum_{|y_i| \leq \omega_i} \frac{\omega_i}{mh} \delta_{\omega_i}(y_i) \right) \\ &= h \sum_{|y_i| \leq \omega_i} \left[A(y_i) - \frac{\alpha(c)}{\beta'(c)} \cdot \frac{\omega_i}{mh} \right] \delta_{\omega_i}(y_i). \end{aligned} \tag{3.15}$$

With $A(y_i)$ given in (3.9) and $a_0 = B(c) = \frac{\alpha(c)}{\beta'(c)}$, we get

$$I_h - I = h \sum_{|y_i| \le \omega_i} \left[\sum_{k=1}^{p-1} a_k \left(\frac{mh}{\omega_i} \right)^{k-1} y_i^k + R_i^p \right] \delta_{\omega_i}(y_i). \tag{3.16}$$

Using the discrete moment condition (3.8), we get

$$\begin{aligned} I_h - I &= h \sum_{|y_i| \le \omega_i} R_i^p \delta_{\omega_i}(y_i) \\ &= h \sum_{|x_i - c| \le mh} \frac{B^{(p)}(\xi_i)}{p!} (x_i - c)^p \delta_{mh}(x_i - c). \end{aligned} \tag{3.17}$$

Now, let M_1 be maximum of $|B^{(p)}(x)/p!|$ on $[c - mh, c + mh]$, and let M_2 be maximum of $|\phi(x)|$ on $[-1, 1]$. Then, we have that

$$\begin{aligned} |I_h - I| &\le h \sum_{|x_i - c| \le mh} M_1 |x_i - c|^p \frac{1}{mh} M_2 \\ &\le h \sum_{|x_i - c| \le mh} M_1 (mh)^p \frac{1}{mh} M_2 \\ &\le h(2m+1) M_1 (mh)^p \frac{1}{mh} M_2 \\ &= 2 M_1 M_2 (mh)^p + M_1 M_2 m^{p-1} h^p \\ &= M_1 M_2 \left(2 + \frac{1}{m} \right) (mh)^p. \end{aligned} \tag{3.18}$$

Thus, the quadrature I_h is p^{th} order of accuracy as desired.

□

The result from above suggests that for the approximation to be p^{th} order of accuracy, the following requirements are sufficient:

1. The discrete δ_{mh} satisfies the p^{th} order discrete moment condition (See Engquist et al. (2005)).
2. The support size ω_i is an approximation of $\dfrac{\beta(x_i)}{x_i - c} mh$.

Following these guidelines, with $m = 1$, we use the support size

$$\omega_i = \frac{\beta(x_i) h}{h_c}, \tag{3.19}$$

where h_c is a third-order approximation of $x_i - c$. (See Smereka (2006).)

As for the discrete delta function, we use $\delta_\omega(x) = \phi(x/\omega)/\omega$ where

$$\phi(x) = \begin{cases} 1 - |x| & \text{if } |x| < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

The hat function ϕ above satisfies the second order discrete moment condition (3.3). (See Engquist et al. (2005).) Thus, using the support size (3.19) we expect to get the second order of accuracy.

We test δ_ω with support size (3.19) by computing (3.1) where $\alpha(x) = \log(x + e)$, $\beta(x) = x^2 + 2x$. This example was used in Wen (2008). The integral intervals $[a_l, b_l], 1 \le l \le 201$ are determined by selecting $a_l = -1.1 + 0.001 \cdot (l - 1)$ and $b_l = a_l + 1 + \sqrt{2}$. For each grid size, we compute the average of the error $|I - I_h|$ of all 201 trials. Table 3.1 shows the average error on each grid size the order of convergence.

N	20	40	80	160	320	640	1280	2560
error	4.6770e-3	1.7736e-3	2.9164e-4	5.7859e-5	1.3169e-5	7.0786e-6	1.1922e-6	2.6656e-7
order		1.3989	2.6044	2.3336	2.1354	0.8956	2.5699	2.1611

Table 3.1 Errors and orders of the δ_ω function.

3.2 Two dimensional cases

For the approximation of delta function in two dimensional space we adapt the approach from Smereka (2006), which allows us to approximate $\delta(w)$ using values of w at the points whose indices are in the set $\mathcal{G}$. Consider an integral of the form

$$I := \int_\Omega f(X)\delta(w(X))\, dX, \tag{3.21}$$

where Ω is a closed and bounded subset of $\mathbb{R}^2$, $f : \mathbb{R}^2 \to \mathbb{R}$ is a continuous function, and $w : \mathbb{R}^2 \to \mathbb{R}$ is a continuously differentiable function. Let Γ be an interface defined by $w^{-1}(0)$ inside Ω. With respect to the Minkowski content measure, we have the identity

$$I = \int_\Gamma \frac{f(X)}{|\nabla w|}\, dS. \tag{3.22}$$

Define $\tilde{f} = \frac{f}{|\nabla w|}$, then the surface integral of $\tilde{f}$ can be expressed as a volume integral

$$I = \int_{\Omega} \tilde{f}(X)\hat{\delta}(X)\, dX. \tag{3.23}$$

Here, $\hat{\delta}$ is a directional derivative of the Heaviside function in the normal direction. We next find an approximation of $|\nabla w|$ at X_{ij} using grid values $w_{ij} = w(X_{ij})$. One of such an approximation is

$$|\nabla_0 w_{ij}| = \sqrt{(D_x^0 w_{ij})^2 + (D_y^0 w_{ij})^2}, \tag{3.24}$$

where D_x^0 and D_p^0 are central difference in x and p directions, respectively. For sufficiently small mesh, the value of $|\nabla_0 w_{ij}|$ cannot be zero. By a simple estimate using Taylor expansion we have

$$\tilde{f}(X_{ij}) = \frac{f_{ij}}{|\nabla_0 w_{ij}|} + O(h^2). \tag{3.25}$$

We approximate (3.21) by

$$I_h = \sum_{ij} \tilde{f}_{ij} h^2 \tilde{\delta}_{ij}. \tag{3.26}$$

Here, $\tilde{\delta}$ is defined by $\tilde{\delta}(w_{i,j}) = \tilde{\delta}_{i,j}^{(+x)} + \tilde{\delta}_{i,j}^{(-x)} + \tilde{\delta}_{i,j}^{(+p)} + \tilde{\delta}_{i,j}^{(-p)}$ where

$$\tilde{\delta}_{i,j}^{(+x)} = \begin{cases} \frac{|w_{i+1,j} D_x^0 w_{i,j}|}{h^2 |D_x^+ w_{i,j}| |\nabla_0 w_{i,j}|} & \text{if } w_{i,j} w_{i+1,j} \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.27}$$

$$\tilde{\delta}_{i,j}^{(-x)} = \begin{cases} \frac{|w_{i-1,j} D_x^0 w_{i,j}|}{h^2 |D_x^- w_{i,j}| |\nabla_0 w_{i,j}|} & \text{if } w_{i-1,j} w_{i,j} \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.28}$$

$$\tilde{\delta}_{i,j}^{(+p)} = \begin{cases} \frac{|w_{i,j+1} D_p^0 w_{i,j}|}{h^2 |D_p^+ w_{i,j}| |\nabla_0 w_{i,j}|} & \text{if } w_{i,j} w_{i,j+1} \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.29}$$

$$\tilde{\delta}_{i,j}^{(-p)} = \begin{cases} \frac{|w_{i,j-1} D_p^0 w_{i,j}|}{h^2 |D_p^- w_{i,j}| |\nabla_0 w_{i,j}|} & \text{if } w_{i,j-1} w_{i,j} \le 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.30}$$

The above $\tilde{\delta}$ is first order of accuracy. One may use higher-order approximation of $\tilde{\delta}$ which can be found in Smereka (2006). If Γ is a closed curve strictly within Ω, then we have

$$|I - I_h| = O(h^q), \quad q = 1, 2.$$

If Γ is an open curve crossing Ω, we need to replace $\tilde{\delta}_{i,j}$ by $\frac{1}{2}\tilde{\delta}_{i,j}$ at the two points $X_{i,j} \in \partial\Omega \cap \Gamma$.

We run the simulation on the above approximation on the following examples.

Example 1 In this example, we approximate an integral involving δ function,

$$I = \int_{-1}^{1}\int_{-1}^{1} \delta(x+y)\,dx\,dy,$$

which has exact value = 2. The simulation is done on the region $[-1,1]^2$ with

$$\Delta x = 0.02, 0.01, \ldots, 0.000625.$$

The result is given in Table 3.2 below.

h	$q=1$			$q=2$		
	I_h	$\lvert I-I_h\rvert$	order	I_h	$\lvert I-I_h\rvert$	order
0.02	1.979999999752491	2.0000e-2		1.939999999951497	6.0000e-2	
0.01	1.989999999751243	1.0000e-2	1.0000	1.969999999950749	3.0000e-2	1.0000
0.005	1.994999999750630	5.0000e-3	1.0000	1.984999999950394	1.5000e-2	1.0000
0.0025	1.997499999750221	2.5000e-3	1.0000	1.992499999950113	7.5000e-3	1.0000
0.00125	1.998749999750160	1.2500e-3	1.0000	1.996249999950009	3.7500e-3	1.0000
0.000625	1.999374999749768	6.2500e-4	1.0000	1.998124999949939	1.8750e-3	1.0000

Table 3.2 Errors and orders of the $\tilde{\delta}$ function for Example 1.

Example 2 The line integral over an open curve,

$$I = \int_{y=\sin(\pi x), -1\le x\le 1} 3x^2 - y^2\,dS,$$

which can be approximated to 3.184207823201663. (We will use this as a reference.) The simulation is done on the region $[-1,1]^2$ with $\Delta x = 0.02,\ 0.01,\ \ldots,\ 0.000625$. The result is given in Table 3.3 below.

In Figure 3.1, we show the zero contour of the function $z(x,y) = y - \sin(\pi x)$ over the region $[-1,1]^2$.

Example 3 The integral involving δ function,

$$I = \int_{-1}^{1}\int_{0}^{1} \delta(y+\tanh(5(x-0.5)))\,dx\,dy,$$

h	$q=1$			$q=2$		
	I_h	$\lvert I-I_h\rvert$	order	I_h	$\lvert I-I_h\rvert$	order
0.02	3.027976901083197	1.5623e-1		2.684086765375988	5.0012e-1	
0.01	3.114770937560744	6.9437e-2	1.1699	2.945656940797150	2.3855e-1	1.0680
0.005	3.155587060384282	2.8621e-2	1.2786	3.075029643763094	1.0918e-1	1.1276
0.0025	3.174202221764498	1.0006e-2	1.5163	3.137570202779929	4.6638e-2	1.2271
0.00125	3.182330838777385	1.8770e-3	2.4143	3.166446782697136	1.7761e-2	1.3928
0.000625	3.185457975174278	1.2502e-3	0.5863	3.179354999864561	4.8528e-3	1.8718

Table 3.3 Errors and orders of the $\tilde{\delta}$ function for Example 2.

which has exact value = 1. This example is motivated by one of the examples in Jin et al. (2008) as we will need to approximate an integral over this contour later. The simulation is done on the region $[0,1]\times[-1,1]$ with $\Delta x = 0.02,\ 0.01,\ \ldots,\ 0.000625$. The result is given in Table 3.4 below.

h	$q=1$			$q=2$		
	I_h	$\lvert I-I_h\rvert$	order	I_h	$\lvert I-I_h\rvert$	order
0.02	0.974564499643422	2.5436e-2		0.841313332174004	1.5869e-1	
0.01	0.989909151717204	1.0091e-2	1.3338	0.959332072640665	4.0668e-2	1.9642
0.005	0.995361500206641	4.6385e-3	1.1213	0.985545531195855	1.4454e-2	1.4924
0.0025	0.997459223787804	2.5408e-3	0.8684	0.992540563438825	7.4594e-3	0.9544
0.00125	0.998842217273770	1.1578e-3	1.1339	0.996386628656795	3.6134e-3	1.0457
0.000625	0.999373070806680	6.2693e-4	0.8850	0.998144462283764	1.8555e-3	0.9615

Table 3.4 Errors and orders of the $\tilde{\delta}$ function for Example 3.

In Figure 3.2, we plot the zero contour of the function $z(x,y) = y + \tanh(5(x-0.5))$ over the region $[0,1]\times[-1,1]$ to show the region we integrate over.

3.3 Higher dimensions

The extension to higher dimension is straightforward. Here, we will discuss the extension to four dimension. Let $\mathbf{X} = (X,P) = (x,y,p,q)$ represent a variable in $\mathbb{R}^4$ and let j,k,l,m represent its respective indices. Let $J := j,k,l,m$ (i.e. $\mathbf{X}_J = (x_j, y_k, p_l, q_m)$, $w_J = w(x_j,y_k,p_l,q_m)$, *etc*). The approximation for $\delta(w_J)$ can be given by $\tilde{\delta}_J$ as follows:

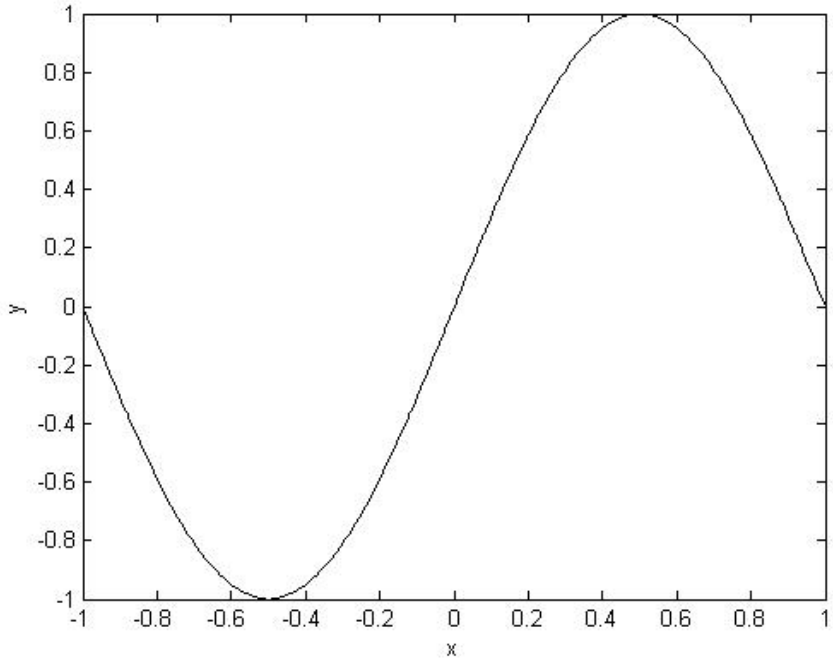

Figure 3.1 The zero contour of the function $z(x, y) = y - \sin(\pi x)$ over the region $[-1, 1]^2$.

$$\tilde{\delta}(w_J) = \tilde{\delta}_J^{(+x)} + \tilde{\delta}_J^{(-x)} + \tilde{\delta}_J^{(+y)} + \tilde{\delta}_J^{(-y)} + \tilde{\delta}_J^{(+p)} + \tilde{\delta}_J^{(-p)} + \tilde{\delta}_J^{(+q)} + \tilde{\delta}_J^{(-q)},$$

where

$$\tilde{\delta}_J^{(+x)} = \begin{cases} \dfrac{|w_{j+1,k,l,m} D_x^0 w_J|}{h^2 |D_x^+ w_J| |\nabla_0 w_J|} & \text{if } w_J w_{j+1,k,l,m} \le 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$\tilde{\delta}_J^{(-x)} = \begin{cases} \dfrac{|w_{j-1,k,l,m} D_x^0 w_J|}{h^2 |D_x^- w_J| |\nabla_0 w_J|} & \text{if } w_{j-1,k,l,m} w_J \le 0 \\ 0 & \text{otherwise.} \end{cases}$$

Here, D_x^+, D_x^-, D_x^0 are backward, forward, and central difference in the $x-$direction. $\nabla_0 w_J$ is defined by $\sqrt{(D_x^0 w_J)^2 + (D_y^0 w_J)^2 + (D_p^0 w_J)^2 + (D_q^0 w_J)^2}$. The other six components can be defined similarly.

Here, we work on the integral of the form

$$\mathcal{I} = \int f(\mathbf{X}) \delta(W(\mathbf{X}))\, d\mathbf{X},$$

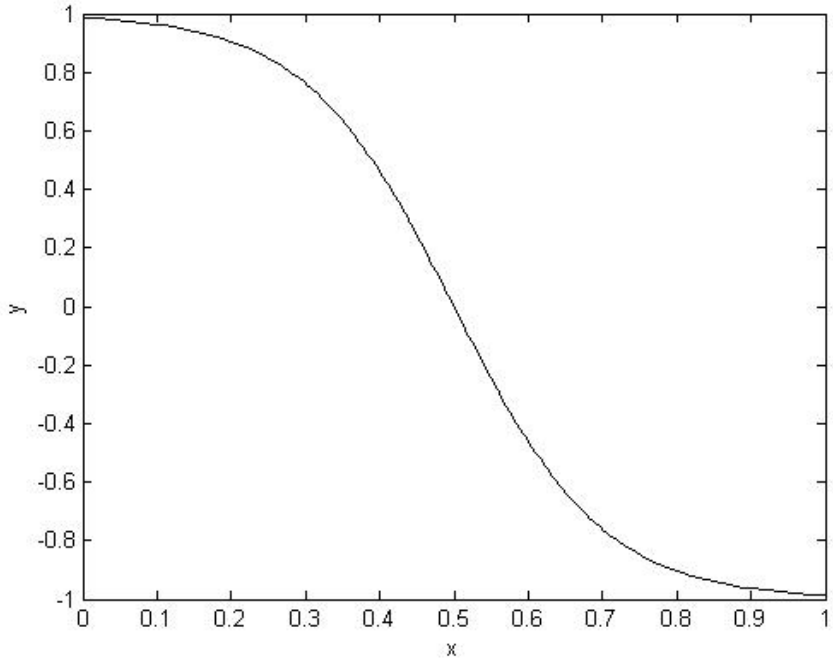

Figure 3.2 The zero contour of the function $z(x, y) = y + \tanh(5(x - 0.5))$ over the region $[0, 1] \times [-1, 1]$.

where $W(\mathbf{X}) = (w_1(\mathbf{X}), w_2(\mathbf{X}))$. We adapt the approximation of delta above and use the approximation

$$\mathcal{I} = h^4 \sum_J \hat{f}_J \tilde{\delta}_J^1 \tilde{\delta}_J^2 + O(h),$$

where $\hat{f} := \dfrac{f}{|\nabla w_1| \cdot |\nabla w_2|}$ and $\tilde{\delta}_J^1$, $\tilde{\delta}_J^2$ are approximations of $\delta(w_1)$, $\delta(w_2)$ respectively.

CHAPTER 4. GAUSSIAN BEAMS FOR SCHRÖDINGER EQUATION

4.1 Introduction

We consider the equation

$$P\psi = 0, \quad (t, y) \in \mathbb{R} \times \mathbb{R}^n, \tag{4.1}$$

$$\psi(0, y) = A_{in}(y)e^{iS_{in}(y)/\epsilon}, \tag{4.2}$$

where $P := -i\epsilon\partial_t + H(x, -i\epsilon\partial_y)$ is a linear differential operator with a real principal symbol $\tau + H(y, p)$. The components of the initial value (4.2) satisfy $A_{in} \in C_0^\infty(\mathbb{R}^n)$ and $S_{in} \in C^\infty(\mathbb{R}^n)$.

To construct the Gaussian beam solution to the above problem, we begin with the classical WKB ansatz

$$\psi^\epsilon(t, y) = A^\epsilon(t, y)e^{i\Phi(t,y)/\epsilon}. \tag{4.3}$$

The amplitude has an asymptotic expansion in terms of ϵ:

$$A^\epsilon(t, y) = A(t, y) + \epsilon A_1(t, y) + \cdots + \epsilon^N A_N(t, y). \tag{4.4}$$

We wish to build asymptotic solutions to (4.1); i.e., we want $P\psi^\epsilon = O(\epsilon^2)$. Substituting (4.3) into (4.1) and setting the coefficients of terms $O(\epsilon^j)$, $j = 0$ or 1, to be zero leads to the following $\epsilon-$independent PDEs

$$\partial_t\Phi + H(y, \nabla_y\Phi) = 0, \tag{4.5}$$

$$\partial_t A + H_p \cdot \nabla_y A = -\frac{A}{2}\left[Tr(H_{yp}) + Tr(\nabla_y^2\Phi H_{pp})\right]. \tag{4.6}$$

Let the bi-characteristics of the Hamilton-Jacobi equation be $X(t; X_0) := (x(t; X_0), p(t; X_0))$

emanating from the initial state $X_0 = (x_0, p_0)$, then

$$\frac{d}{dt}x = H_p, \quad x(0) = x_0, \tag{4.7}$$

$$\frac{d}{dt}p = -H_x, \quad p(0) = p_0, \tag{4.8}$$

or in compact form as

$$\frac{d}{dt}X(t; X_0) = v(X(t; X_0)), \quad X(0; X_0) = X_0, \tag{4.9}$$

where $v := (H_p, -H_x)$. The idea underlying Gaussian beams is to build asymptotic solutions concentrated on a single curve in physical space $(t, y) \in \mathbb{R} \times \mathbb{R}^n$, which we take as the ray $\gamma = \{(x(t; X_0), t)\}$ defined above. The phase should be real on the ray γ, but is required to satisfy

$$Im(\Phi) \geq c(y - x(t; X_0))^2, \tag{4.10}$$

for some $c > 0$, which gives the Gaussian profiles. According to the Gaussian beam theory by Ralston (1982), the first order Gaussian beam phase takes the following form

$$\begin{aligned}\Phi(t, y; X_0) = S(t; X_0) + p(t; X_0) \cdot (y - x(t; X_0)) \\ + \frac{1}{2}(y - x(t; X_0)) \cdot M(t; X_0)(y - x(t; X_0)),\end{aligned} \tag{4.11}$$

associated with the amplitude

$$A(t, y; X_0) = A(t; X_0). \tag{4.12}$$

Plug these into the PDEs (4.5)-(4.6) for Φ and A with $p = \Phi_y(t, x(t; X_0))$ we have

$$\frac{d}{dt}S(t; X_0) = p \cdot H_p - H(x, p), \tag{4.13a}$$

$$\frac{d}{dt}M(t; X_0) + H_{xx} + H_{xp}M + MH_{px} + MH_{pp}M = 0, \tag{4.13b}$$

$$\frac{d}{dt}A(t; X_0) = -\frac{A}{2}[Tr[H_{xp} + Tr[MH_{pp}]]. \tag{4.13c}$$

It is shown in Ralston (1982) that $Im(M)$ remains positive definite if it is chosen so initially, so that (4.10) holds for all time. This construction ensures that the following Gaussian beam ansatz is an approximate solution,

$$\psi_{GB}(t, y; X_0) = A(t; X_0) \exp\left(\frac{i}{\epsilon}\Phi(t, y; X_0)\right). \tag{4.14}$$

4.1.1 Recovery of the wave fields by superposition

4.1.1.1 Phase space-based recovery

For linear equations as we consider in this chapter, the high frequency wave field ψ at (t, y) in physical space can be generated by a superposition of Gaussian beams,

$$\psi^\epsilon(t,y) = \frac{1}{(2\pi\epsilon)^{n/2}} \int_{\Omega(0)} \Psi_{GB}(t,y;X_0)dX_0, \tag{4.15}$$

where

$$\Omega(0) = \{X_0, \quad x_0 \in K_0 := supp(A_{in}), \quad p_0 \in range(\partial_x S_{in}(x))\} \tag{4.16}$$

is an open domain in phase space from which we construct initial Gaussian beams from the given data.

Let $\Omega(t)$ be the image of $\Omega(0)$ under the Hamiltonian flow, symbolically expressed as

$$\Omega(t) = X(t, \Omega(0)). \tag{4.17}$$

We thus define the phase space-based Gaussian beam ansatz $\Psi_{PGB}(t, y, X)$ by changing X_0 to X through the Hamiltonian map:

$$\Psi_{PGB}(t, y, X(t; X_0)) \equiv \Psi_{GB}(t, y; X_0). \tag{4.18}$$

Using the change of variables we see that the superposition over the moving domain $\Omega(t)$ remains a correct asymptotic solution,

$$\psi^\epsilon(t,y) = \frac{1}{(2\pi\epsilon)^{n/2}} \int_{\Omega(t)} \Psi_{PGB}(t,y,X)dX. \tag{4.19}$$

Here

$$\Psi_{PGB}(t,y,X) = \tilde{A}(t,X) \exp\left(\frac{i}{\epsilon}\tilde{\Phi}(t,y,X)\right), \tag{4.20}$$

where

$$\tilde{\Phi}(t,y,X) = \tilde{S}(t,X) + p\cdot(y-x) + \frac{1}{2}(y-x)\cdot\tilde{M}(t,X)(y-x). \tag{4.21}$$

Hence, $\Psi_{PGB}(t,y,X)$ can be obtained by solving Liouville-type PDEs for $\tilde{S},\tilde{M},\tilde{A}$ in phase space:

$$\mathcal{L}[\tilde{S}] = p\cdot H_p - H(x,p), \tag{4.22a}$$

$$\mathcal{L}[\tilde{M}] + H_{xx} + H_{xp}\tilde{M} + \tilde{M}H_{px} + \tilde{M}H_{pp}\tilde{M} = 0, \tag{4.22b}$$

$$\mathcal{L}[\tilde{A}] = -\frac{\tilde{A}}{2}\left[TrH_{xp} + Tr(\tilde{M}H_{pp})\right], \tag{4.22c}$$

where $\mathcal{L}$ is the Liouville operator defined by

$$\mathcal{L} := \partial_t + H_p\cdot\nabla_x - H_x\cdot\nabla_p. \tag{4.23}$$

4.1.1.2 Projected recovery

For highly oscillatory initial data, the classical phase stationary theory suggests that the main contribution comes from the set

$$\Omega(0) = \{(x_0,p_0),\quad x_0\in K_0,\quad p_0 = \nabla_y S_0(x_0)\}. \tag{4.24}$$

It suffices to use the reduced Gaussian beam superposition

$$\psi^\epsilon(t,y) = \frac{1}{(2\pi\epsilon)^{n/2}}\int_{K_0}\Psi_{GB}(t,y;x_0)dx_0, \tag{4.25}$$

which when $t=0$ can be explicitly expressed as an integral in phase space,

$$\psi^\epsilon(0,y) = \frac{1}{(2\pi\epsilon)^{n/2}}\int_{K_0}\Psi_{GB}(0,y;x_0)dx_0 \tag{4.26}$$

$$= \frac{1}{(2\pi\epsilon)^{n/2}}\int_{\Omega(0)}\Psi_{PGB}(0,y;X_0)\delta(p_0-\nabla_x S_{\text{in}}(x_0))dX_0. \tag{4.27}$$

In order to track the time evolution of the surface $p=\nabla_x S_{in}(x)$, we introduce a level set function $w=w(t,X)$ such that

$$\mathcal{L}[w] = 0,\quad w(0,X) = p-\nabla_x S_{in}(x). \tag{4.28}$$

We now come to the domain propagation based Gaussian beam superposition,

$$\psi^\epsilon(t,y) := \frac{1}{(2\pi\epsilon)^{n/2}}\int_{\Omega(t)}\Psi_{PGB}(t,y,X)\delta(w(t,X))dX. \tag{4.29}$$

This superposition is uniquely determined once the initial data for $(\tilde{S}, \tilde{A}, \tilde{M})$ is specified. According to the above form of $\psi^\epsilon(0, y)$, where

$$\psi_{PGB}(0, y, X) = \tilde{A}(0, X) \exp\left(\frac{i}{\epsilon}\Phi(0, y, X)\right), \tag{4.30}$$

$$\Phi(0, y, X) = \tilde{S}(0, X) + p \cdot (y - x) + \frac{1}{2}(y - x) \cdot \tilde{M}(0, X)(y - x). \tag{4.31}$$

We shall take $\tilde{A}(0, X) = A_{in}(x), \tilde{S}(0, X) = S_{in}(x)$, and $\tilde{M}(0, X) = \partial_x^2 S_{in}(x) + iI$.

4.1.2 Gaussian beam superposition for the Schrödinger equation

In the case of Schrödinger equation, we have

$$-i\epsilon\partial_t\psi + V(y)\psi - \frac{\epsilon^2}{2}\Delta\psi = 0, \quad (t, y) \in \mathbb{R} \times \mathbb{R}^n, \tag{4.32}$$

$$\psi(0, y) = A_{in}(y)e^{iS_{in}(y)/\epsilon}. \tag{4.33}$$

In this case, the Hamiltonian is $H(y, p) = \frac{|p|^2}{2} + V(y)$. The phase space based Gaussian beam components $\tilde{S}, \tilde{M}, \tilde{A}$ can be obtained by solving the following equations:

$$\mathcal{L}[\tilde{S}] = -V(x) + \frac{|p|^2}{2}, \quad \tilde{S}(0, X) = S_{in}(x), \tag{4.34a}$$

$$\mathcal{L}[\tilde{M}] + \partial_x^2 V(x) + \tilde{M}^2 = 0, \quad \tilde{M}(0, X) = \partial_x^2 S_{in}(x) + iI, \tag{4.34b}$$

$$\mathcal{L}[\tilde{A}] = -\frac{\tilde{A}}{2}\tilde{M}, \quad \tilde{A}(0, X) = A_{in}(x), \tag{4.34c}$$

where the Liouville operator can now be specified as

$$\mathcal{L} := \partial_t + p\partial_x - V'(x)\partial_p. \tag{4.35}$$

Our goal is to approximate the domain propagation-based Gaussian beam superposition for the Schrödinger equation using (4.29) with w satisfying

$$w_t + pw_x - V'(x)w_p = 0, \quad w(0, X) = p - \partial_x S_{in}(x). \tag{4.36}$$

Note that with the above chosen initial data, it is guaranteed to have a control over initial data (see Liu and Ralston (2010)):

$$\|\psi_{in}(\cdot) - \psi^\epsilon(0, \cdot)\|_{L^2} \leq C\epsilon^{1/2}, \tag{4.37}$$

for some constant C independent of ϵ.

4.2 Numerical implementation

4.2.1 Discretization strategies.

Here we outline what is needed to numerically approximate the recovery scheme (4.29). The first step to evaluate (4.29) is to use a special kind of quadrature rule to approximate the integral whose integrand contains a delta function. In other words, we want to find $\tilde{\delta}$ so that (4.29) can be approximated by

$$\phi^{\epsilon}_{\Delta,\eta}(t,y) = \frac{\Delta x \Delta p}{(2\pi\epsilon)^{n/2}} \sum_{X_{i,j}\in\Omega(t)} \Psi_{i,j}(t,y)\tilde{\delta}(w_{i,j}(t)), \tag{4.38}$$

where for any $t > 0$,

$$w_{i,j}(t) = w(t, X_{i,j}), \tag{4.39}$$

is obtained from solving the level set equation (4.36), and

$$\Psi_{i,j}(t,y) = \tilde{A}(t,X_{i,j}) \exp\left(\frac{i}{\epsilon}\Phi(t,y,X_{i,j})\right), \tag{4.40}$$
$$\Phi(t,y,X_{i,j}) = \tilde{S}(t,X_{i,j}) + p_j \cdot (y - x_i) + \frac{1}{2}(y-x_i)\cdot \tilde{M}(t,X_{i,j})(y-x_i),$$

with $\tilde{A}, \tilde{S}, \tilde{M}$ obtained by solving the Liouville PDEs (4.34a-4.34c). In phase space we use rectangle meshes of size $\Delta x \Delta p$, where

$$\Delta x := \prod_{i=1}^{n} \Delta x_i, \quad \Delta p := \prod_{i=1}^{n} \Delta p_i,$$

in multi-dimensional case. Each node of the mesh is labeled by $X_{i,j} = (x_i, p_j)$. Also, we define $\eta > 0$ to be the support size of $\tilde{\delta}$, i.e. $\tilde{\delta}(w(X_{i,j})) = 0$ if $\text{dist}(X_{i,j}, \Gamma(t)) > \eta$. The error introduced in this step may depend on $\Delta x, \Delta p$, and η only. Accuracy can be improved by a better choice of $\tilde{\delta}$, as detailed in chapter 3.

Because $\tilde{\delta}$ has a compact support, we can further simplify the summation (4.38) to

$$\psi^{\epsilon}_{\Delta,\eta,\mathcal{G}}(t,y) = \frac{\Delta x \Delta p}{(2\pi\epsilon)^{n/2}} \sum_{i,j\in\mathcal{G}} \Psi_{i,j}(t,y)\tilde{\delta}(w_{i,j}(t)), \tag{4.41}$$

where

$$\mathcal{G} = \{(i,j) \,|\, X_{i,j} \in \Omega(t),\ \tilde{\delta}(w(t,X_{i,j})) \neq 0\}. \tag{4.42}$$

To complete our numerical discretization, we are left to prepare the following:

- A semi-Lagrangian method for computing $w_{i,j}(t)$.
- A fast search algorithm for the effective index set $\mathcal{G}$.
- A refinement of discrete delta functions
- Efficient computation of the Gaussian beams $\Psi_{i,j}(t, y)$.

Once these are done, the Gaussian beam superposition is completed. The search algorithm and delta approximation have been addressed in chapters 2 and 3 respectively. We discuss the computation of $w_{i,j}(t)$ and $\Psi_{i,j}(t, y)$ below.

4.2.2 Computing the level set function

Recall that w satisfies the Liouville equation

$$\partial_t w + p\partial_x w - V'(x)\partial_p w = 0, \quad w(0, x, p) = p - \nabla_x S_{\text{in}}(x). \tag{4.43}$$

By the method of characteristics, it is equivalent to finding the trajectory

$$\frac{d}{dt}X = (p, -V'(x)), \quad t > 0 \tag{4.44}$$

and

$$\frac{dw}{dt} = \frac{d}{dt}w(t, X(t; X_0)) = 0, \quad w(0) = p_0 - \nabla_x S_{\text{in}}(x_0). \tag{4.45}$$

Take τ as a final time, and K be the time steps to be taken. Hence we have the discretized time variables

$$t^k = k\Delta t, \quad \Delta t = \tau/K.$$

Let Θ be an ODE solver of (4.44) that traces the characteristic curve for one step, if X^k denotes the numerical solution at t^k, then

$$X^{k+1} = \Theta(X^k), \quad k = 0, \ldots, K-1.$$

For each grid point $X_{i,j}$ at t^K, we trace back to find the numerical approximation of the initial position

$$X^0 = \Theta^{-K}(X_{i,j}),$$

from which we take the value for w

$$w^0 := w(0, X^0) = p^0 - \nabla_x S_{\text{in}}(x^0),$$

and update as

$$w^{k+1} = w^k, \quad k = 0, \cdots, K-1.$$

Finally, we take

$$w^K_{i,j} = w^K.$$

to approximate $w(t^K, X_{i,j})$. The sign of this quantity can help to determine if a given grid $X_{i,j}$ belongs to the effective index set $\mathcal{G}$.

As indicated above, our main task with the ODE solver is to find X^0 from $X^K := \{X_{i,j}\}$. The following observation is helpful.

Lemma 4.2.1. *Assume $H(x,p)$ is Lipschitz continuous on x, p and satisfies $H(x,p) = H(x,-p)$. Let $\Theta : X(0) \to X(\tau)$ be a forward ODE solver for the Hamiltonian equation (4.9), then*

$$X(0) = J\Theta(JX(\tau)), \quad J := \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix}$$

for all $\tau > 0$.

Proof. Define $Y(t) = JX(\tau - t)$. We first show that Y also satisfies (4.9). Write $Y(t) = (y(t), q(t))$, then it follows that $y(t) = x(\tau - t) =: x(T)$ and $q(t) = -p(\tau - t) =: -p(T)$. Thus, we have that

$$y'(t) = \frac{dx}{dT} \cdot \frac{dT}{dt} = -\frac{dx}{dT} = -H_p(x(T), p(T)) = H_q(y(t), -q(t)), \tag{4.46}$$

$$q'(t) = -\frac{dp}{dT} \cdot \frac{dT}{dt} = -\frac{dp}{dT} = -H_x(x(T), p(T)) = -H_y(y(t), -q(t)). \tag{4.47}$$

Because of the symmetry condition on H, we have that

$$\frac{d}{dt} Y(t) = \begin{pmatrix} H_q(y(t), q(t)) \\ -H_y(y(t), q(t)) \end{pmatrix} = J\nabla_Y H(y(t), q(t)). \tag{4.48}$$

So, the ODE solver Θ can be applied to Y.

Now, we have from definition of Y that $Y(0) = JX(\tau)$ and $Y(\tau) = JX(0)$. Because $J^2 = I$, we have $X(0) = JY(\tau) = J\Theta(Y(0)) = J\Theta(JX(\tau))$ as desired.

□

Take τ to be a time step for the iteration. With the above result we have $X^k = J\Theta(JX^{k+1})$. Repeating this process yields,

$$X^k = J(\Theta)^{K-k}(JX^K), \quad k = K-1, \cdots 0. \tag{4.49}$$

One option for the ODE solver Θ for the Hamiltonian flow is the Runge-Kutta method, another option is a symplectic solver which is specialized for the Hamiltonian flow. In our simulation we adapt the usual Runge-Kutta method.

4.2.3 Computing the Gaussian beam components.

In the previous section, we found the set $\mathcal{G}$ of effective indices , and an approximation for $\delta(w)$, so that $\psi^\epsilon(\tau, y)$ can be approximated by

$$\psi^\epsilon_{\Delta,\eta,\mathcal{G}}(t^K, y) \quad = \quad \frac{\Delta x \Delta p}{(2\pi\epsilon)^{n/2}} \sum_{i,j\in\mathcal{G}} \Psi^K_{i,j}(y)\tilde{\delta}(w^K_{i,j}). \tag{4.50}$$

We already knew how to compute $w^K_{i,j}$, so our final task is to calculate $\Psi^K_{i,j}(y)$, which is defined as

$$\Psi^K_{i,j}(y) = \tilde{A}(t^K, X_{i,j}) \exp\left(\frac{i}{\epsilon}\Phi^K_{i,j}(y)\right), \tag{4.51}$$

$$\Phi^K_{i,j}(y) := \tilde{S}(t^K, X_{i,j}) + p_j \cdot (y - x_i) + \frac{1}{2}(y - x_i) \cdot \tilde{M}(t^K, X_{i,j})(y - x_i). \tag{4.52}$$

Note that for each $X_{i,j}$, we have obtained an approximate Hamiltonian map at t^k

$$X^k = \Theta^{(k-K)}(X_{i,j}), \quad k = K-1, K-2, \cdots, 0.$$

Instead of solving the Liouville-type equations, we shall use the trajectory information of X^k to approximate $\tilde{A}, \tilde{S}$ and $\tilde{M}$ at each time step (t^k, X^k).

Note that $S(t, X_0) = \tilde{S}(t, X(t; X_0))$ for any initial position X_0, and S satisfies the ODE

$$\frac{d}{dt}S = F(X) := -V(x) + \frac{|p|^2}{2}.$$

We use the trapezoidal rule to estimate $S(t^k, X^k)$ in the following way

$$\begin{aligned} S^0 &= S_{\text{in}}(x^0), \quad X^0 = \Theta^{-K}(X_{i,j}), \\ S^{k+1} &= S^k + \frac{\Delta t}{2}\left(F^k + F^{k+1}\right), \quad k = 0, 1, \cdots, K-1, \\ F^k &:= -V(x^k) + \frac{|p^k|^2}{2}, \quad X^k = \Theta^{-(K-k)}(X_{i,j}), \\ \tilde{S}(t^K, X_{i,j}) &\sim S^K. \end{aligned}$$

The error comes from both numerically solving the ODE for S and the error of the map predication X^k.

Similarly we can estimate $\tilde{M}(t^K, X_{i,j})$

$$\begin{aligned} M^0 &= \partial_x^2 S_{\text{in}}(x^0) + i, \quad X^0 = \Theta^{-K}(X_{i,j}), \\ M^* &= M^k + \Delta t[-(M^k)^2 - \partial_x^2 V(x^k)], \\ M^{k+1} &= M^k + \frac{\Delta t}{2}\left(-(M^k)^2 - \partial_x^2 V(x^k) - (M^*)^2 - \partial_x^2 V(x^{k+1})\right), \quad k = 0, 1, \cdots, K-1, \\ \tilde{M}(t^K, X_{i,j}) &\approx M^K. \end{aligned}$$

Finally, from $\frac{dA}{dt} = -\frac{A}{2}M$ it follows

$$\begin{aligned} A^0 &= A_{\text{in}}(x^0), \quad X^0 = \Theta^{-K}(X_{i,j}), \\ A^{k+1} &= A^k \exp\left(-\frac{\Delta t}{4}\left(M^k + M^{k+1}\right)\right), \quad k = 0, 1, \cdots, K-1, \\ \tilde{A}(t^K, X_{i,j}) &\approx A^K. \end{aligned}$$

These ODE solvers are all second order in time.

4.3 Numerical results in one dimension

In this section we run the numerical simulation on different types of potential function $V(x)$. We will use two different initial phase $S_{in}(x)$:

$$\begin{aligned} S_1 &= -\frac{1}{2}\left(x - \frac{1}{2}\right)^2, \\ S_2 &= -\frac{1}{5}\log\left[2\cosh\left(5\left(x - \frac{1}{2}\right)\right)\right]. \end{aligned}$$

In all examples, we will use the amplitude:

$$A_{in}(x) = e^{-25\left(x-\frac{1}{2}\right)^2},$$

with the compact support $[0, 1]$.

Error computation. In all examples below, we compute the L^2 error between the reference solution $\psi(t, y)$ and the computed solution $\psi^\epsilon(t, y)$ by using

$$\|\psi(t,\cdot) - \psi^\epsilon(t,\cdot)\|^2 \approx \sum_j \left(\psi(t, x_j) - \psi^\epsilon(t, x_j)\right)^2 \Delta x, \tag{4.53}$$

where x_j are equidistant sample points with $\Delta x = 1/20480$.

4.3.1 Zero Potential.

In the first two examples, we test the accuracy of the scheme on the zero potential $V(x) = 0$. In the first example where $S_{in}(x)$ is quadratic, we expect to get the first order of accuracy in terms of ϵ. In the second example where $S_{in}(x)$ is not quadratic, we expect to get error of order $O(\epsilon^{1/2})$.

Example 1: Zero potential with quadratic phase For this example we can solve for $\psi(t, y)$ analytically,

$$\psi(t, y) \quad = \quad \frac{1}{\sqrt{2\pi i\epsilon t}} \int_{\mathbb{R}} A_{in}(x_0) e^{-ix_0^2/(2\epsilon) + i(y-x_0)^2/(2\epsilon t)}\, dx_0. \tag{4.54}$$

We compare $\psi^\epsilon(t, y)$ with $\psi(t, y)$ at the caustic time $t = 1$ by computing the L^2 error (4.53) on the sample points in $[0, 1]$. As we vary h and ϵ, the convergence rate r can be observed over the region given in Figure 4.1.

Example 2: zero potential with non-quadratic phase This example is taken from Jin et al. (2008). In this example we compute the reference solution ψ using the Strang splitting spectral method adapted from Bao et al. (2002). Figure 4.2 shows the region where convergence can be observed. Here, the L^2 error (4.53) is computed from the sample points in $[0, 1]$ at the time $t = 0.5$ where the caustic occurs.

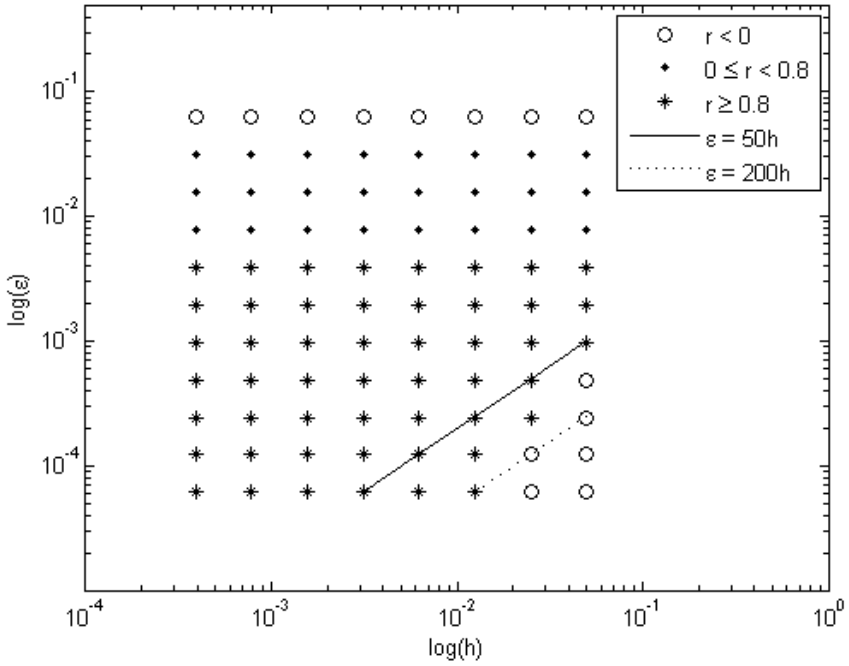

Figure 4.1 The region where the convergence is observed in example 1.

As ϵ decreases, we get the expected order of accuracy, $r = 1$, and $r = 0.5$ for example 1 and 2 respectively. We observe that we get the desired convergence rate when ϵ is small. For each of those ϵ's, there is an appropriate range of the mesh size h that yields the convergence. Note that the value of h that yields the convergence is larger than ϵ, as we wanted to see.

4.3.2 Quadratic Potential.

In these examples, we use the quadratic potential $V(x) = \frac{1}{2}\left(x - \frac{1}{2}\right)^2$ together with both of the initial phases S_1 and S_2. Because the potential is quadratic, we expect to see similar convergence rate as the zero potential examples above. Since we are only interested in the convergence rate, we run the simulation only on one fixed mesh size, $h = 1/1280$. Table 4.1 shows the L^2 errors (4.53) computed on the sample points in $[0, 1]$ at the caustic times ($t = 0.8$ for S_1, $t = 0.5$ for S_2).

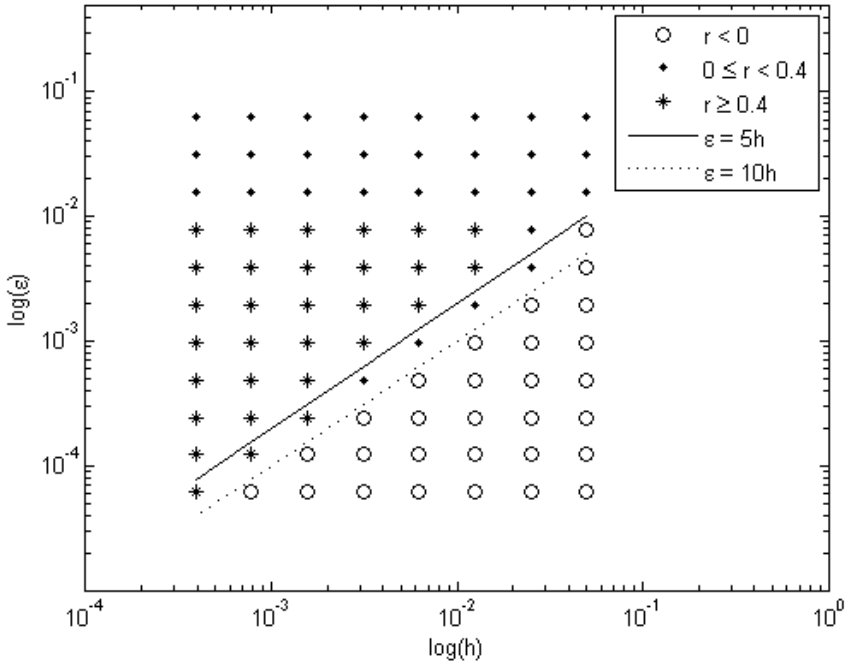

Figure 4.2 The region where the convergence is observed in example 2.

4.3.3 Lazy Potential.

In these examples, we use the lazy potential $V(x) = \frac{1}{12}\left(x - \frac{1}{2}\right)^4$ together with both of the initial phases S_1 and S_2. Because the potential is no longer quadratic, we expect to see the half order of convergence from both initial phases. We run the simulation on a fixed mesh $h = 1/1280$. The observation points are chosen the same way as above. Table 4.2 shows the L^2 errors (4.53) computed on the sample points in $[0, 1]$ at the caustic times ($t = 1.0$ for S_1, $t = 0.5$ for S_2).

The behavior of the errors for the non-zero potentials is similar to that of the zero potential. At a fixed mesh size, the error does not necessarily decrease as ϵ gets smaller. In fact, only when ϵ gets smaller than a threshold, the convergence is observed. Also, the convergence rate reaches its peak at some value of ϵ and it starts to drop again when ϵ is too small compared to h.

ϵ	S_1		S_2	
	$\|(\psi^\epsilon - \psi)(0.8, \cdot)\|_2$	order	$\|(\psi^\epsilon - \psi)(0.5, \cdot)\|_2$	order
$1/2^3$	1.9002e-1		2.3625e-1	
$1/2^4$	2.3357e-1	-0.2977	2.3122e-1	0.0310
$1/2^5$	1.7778e-1	0.3937	1.9082e-1	0.2771
$1/2^6$	1.1553e-1	0.6219	1.4664e-1	0.3799
$1/2^7$	6.8411e-2	0.7559	1.0814e-1	0.4394
$1/2^8$	3.7791e-2	0.8562	7.5684e-2	0.5148
$1/2^9$	1.9969e-2	0.9203	4.9759e-2	0.6050
$1/2^{10}$	1.0299e-2	0.9553	3.0740e-2	0.6948
$1/2^{11}$	5.2747e-3	0.9653	1.8094e-2	0.7646
$1/2^{12}$	2.7623e-3	0.9332	1.0680e-2	0.7607
$1/2^{13}$	1.6256e-3	0.7649	7.5391e-3	0.5024
$1/2^{14}$	1.3865e-3	0.2295	8.6161e-3	-0.1927

Table 4.1 Errors and orders for the examples with quadratic potential.

ϵ	S_1		S_2	
	$\|(\psi^\epsilon - \psi)(1.0, \cdot)\|_2$	order	$\|(\psi^\epsilon - \psi)(0.5, \cdot)\|_2$	order
$1/2^3$	1.1733e-1		2.3244e-1	
$1/2^4$	1.9168e-1	-0.7080	2.3072e-1	0.0107
$1/2^5$	1.7657e-1	0.1185	1.9084e-1	0.2738
$1/2^6$	1.1530e-1	0.6148	1.4664e-1	0.3801
$1/2^7$	6.8250e-2	0.7565	1.0811e-1	0.4398
$1/2^8$	3.7683e-2	0.8569	7.5647e-2	0.5152
$1/2^9$	1.9896e-2	0.9214	4.9721e-2	0.6054
$1/2^{10}$	1.0237e-2	0.9587	3.0710e-2	0.6951
$1/2^{11}$	5.1958e-3	0.9784	1.8080e-2	0.7643
$1/2^{12}$	2.6240e-3	0.9856	1.0690e-2	0.7581
$1/2^{13}$	1.3393e-3	0.9703	7.6150e-3	0.4893
$1/2^{14}$	7.5153e-4	0.8336	8.8489e-3	-0.2166

Table 4.2 Errors and orders for the examples with lazy potential.

4.3.4 Periodic Potential.

In this example adapted from Qian and Ying (2010), we use the periodic potential $V(x) = \cos(2\pi(x+1))$ with $S_{in}(x) = S_2$. Table 4.3 shows the error at $t = 0.5$. The simulation is done with $h = 1/1280$. The observation points are taken from $[-0.5, 1.5]$.

ϵ	$1/(2^4\pi)$	$1/(2^5\pi)$	$1/(2^6\pi)$	$1/(2^7\pi)$	$1/(2^8\pi)$	$1/(2^9\pi)$	$1/(2^{10}\pi)$	$1/(2^{11}\pi)$	$1/(2^{12}\pi)$	$1/(2^{13}\pi)$
L^2	0.3094	0.2889	0.2704	0.2504	0.2278	0.2029	0.1761	0.1487	0.1256	0.1208

Table 4.3 Errors for the example with periodic potential.

From Table 4.3, we see that the error decreases as ϵ gets smaller. However, convergence process is very slow. This is because the geometry of the interface becomes too complex to identify some appropriate values of ϵ and h that yield the convergence. In Figure 4.3, we show the interface at the time $t = 0.5$, where the narrow geometry of the domain $\Omega(0.5)$ poses difficulties.

4.4 Numerical results in two dimension

In this section we run the numerical simulation on two dimensional problems with phase space (X, P) where $X = (x_1, x_2) \in \mathbb{R}^2, P = (p_1, p_2) \in \mathbb{R}^2$. The initial amplitude for all examples is given by

$$A_{in}(X) = e^{-25\left(x_1^2 + x_2^2\right)},$$

for $x_1, x_2 \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ and zero elsewhere.

We compute the L^2 error between the reference solution $\psi(t, Y)$, $Y = (y_1, y_2) \in \mathbb{R}^2$, and the computed solution $\psi^\epsilon(t, Y)$ by using

$$\|\psi(t,\cdot) - \psi^\epsilon(t,\cdot)\|^2 \approx \sum_j \left(\psi(t, y_j, 0) - \psi^\epsilon(t, y_j, 0)\right)^2 \Delta y, \tag{4.55}$$

where y_j are equidistant sample points with $\Delta y = 1/2048$.

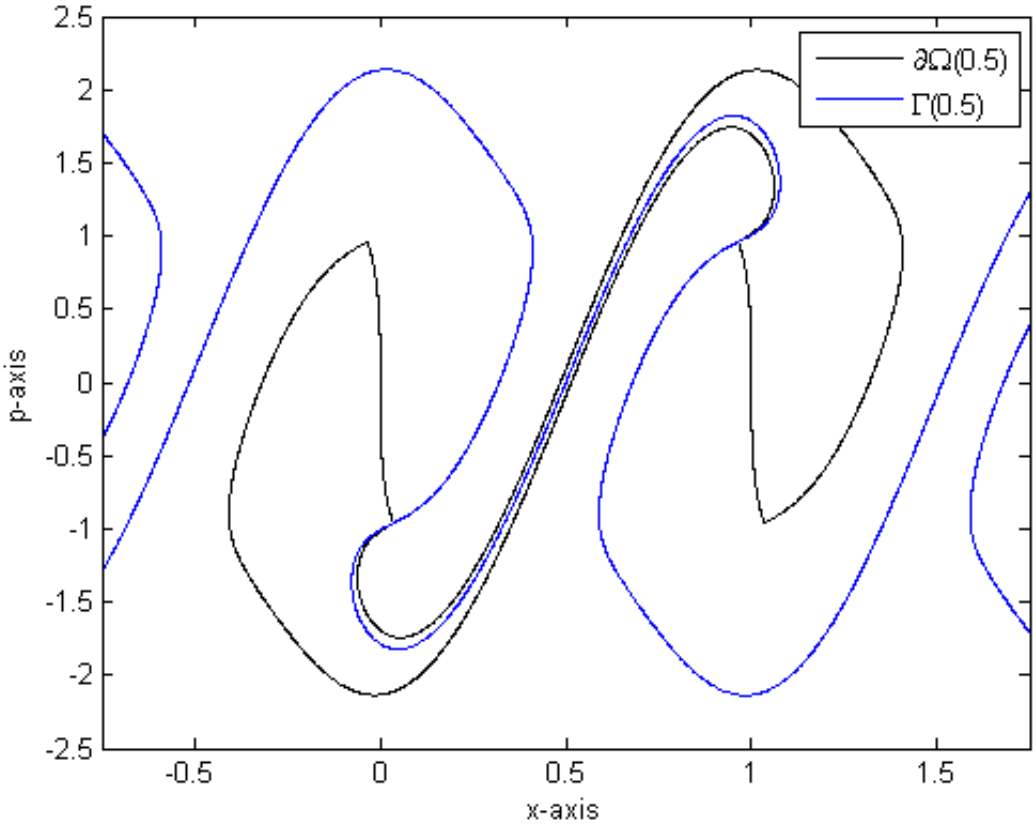

Figure 4.3 The zero level set $\Gamma(0.5)$ and the domain $\Omega(0.5)$ for the example with periodic potential.

4.4.1 Zero Potential.

We start with the high accuracy example (Liu and Ralston (2010)) with $V = 0$ and $S_{in}(X) = -|X|^2/2$. The exact solution can be obtained using the Fourier transform. The errors at time $t = 0.5$ (before caustic) and $t = 1.0$ (at caustic) are in the Table 4.4 below.

4.4.2 Non-zero Potential.

In this example, as taken from Jin et al. (2008), we use $V = 10$ and

$$S_{in}(X) = -\frac{1}{5}\log\left[2\left(\cosh 5x_1 + \cosh 5x_2\right)\right].$$

ϵ	$t = 0.5$		$t = 1.0$	
	$\|(\psi^\epsilon - \psi)(0.5, \cdot)\|_2$	order	$\|(\psi^\epsilon - \psi)(1.0, \cdot)\|_2$	order
$1/2^7$	1.6041e-01		1.0961e-01	
$1/2^8$	9.6150e-02	0.7384	8.5738e-02	0.3544
$1/2^9$	5.2651e-02	0.8688	6.4213e-02	0.4171
$1/2^{10}$	2.7683e-02	0.9275	4.6994e-02	0.4504
$1/2^{11}$	1.4512e-02	0.9318	3.4124e-02	0.4617
$1/2^{12}$	8.0905e-03	0.8429	2.4957e-02	0.4513

Table 4.4 Errors and orders for the 2D example with zero potential.

For the reference solution, we use the Strang splitting spectral method adapted from Bao et al. (2002) to compute the one-dimensional Schrödinger equation with $V = 5$. It follows that the product $\psi_1(t, y_1) \cdot \psi_2(t, y_2)$ is a solution to the two dimensional problem $\psi(t, y_1, y_2)$ because of the symmetric property of this example. We should point out that our numerical implementation is not limited to any symmetry property.

The error at the caustic time ($t = 0.5$) is given in Table 4.5 below. Note that the behavior of the convergence is similar to that in 1D case: we observe convergence only when the mesh size and ϵ are compatible. If h is too large compared to ϵ, we will not see the convergence.

ϵ	$\|\left(\psi^\epsilon - \psi\right)(0.5, \cdot)\|_2$	order
$1/2^7$	1.6212e-01	
$1/2^8$	1.2004e-01	0.4336
$1/2^9$	8.0619e-02	0.5743
$1/2^{10}$	5.1279e-02	0.6527
$1/2^{11}$	3.1615e-02	0.6978
$1/2^{12}$	2.9333e-02	0.1081

Table 4.5 Errors and orders for the 2D example with non-zero potential.

CHAPTER 5. ENERGY-PRESERVING LOCAL DISCONTINUOUS GALERKIN METHOD FOR BURGER-POISSON EQUATION

5.1 The discontinuous Galerkin method

5.1.1 LDG formulation

We develop a local discontinuous Galerkin (LDG) method for the BP equation subject to initial data $u_0(x)$, posed on $I = [0, L]$ with periodic boundary conditions. Let us partition the interval I into $0 = x_{1/2}, x_{3/2}, \ldots, x_{N+1/2} = L$ to get N equal subintervals and denote each cell by $I_j = [x_{j-1/2}, x_{j+1/2}], j = 1, \ldots, N$. The center of the cell is $x_j = \frac{1}{2}\left(x_{j-1/2} + x_{j+1/2}\right)$. Here the uniform mesh is taken just for simplicity of the analysis; one may well use non-uniform meshes in the implementation of the method.

The piecewise polynomial space V_h^k is defined as the space of polynomials of degree up to k in each cell I_j, that is,

$$V_h^k = \{v : v|_{I_j} \in P^k(I_j), j = 1, 2, \ldots, N\}. \tag{5.1}$$

Note that functions in V_h^k are allowed to have discontinuities across the interfaces. The solution of the DG method is denoted by u_h, which belongs to the finite element space V_h^k. We denote the limit values of u_h at $x_{j+1/2}$ from the right and from the left by $(u_h)^+_{j+1/2}$ and $(u_h)^-_{j+1/2}$ respectively. Let ω be a piecewise smooth function, its jump across the cell interface be denoted by $[\omega] := \omega^+ - \omega^-$, and its average at the cell interface, $\frac{\omega^+ + \omega^-}{2}$, be denote by $\{\omega\}$.

To define the LDG method, we introduce an auxiliary variable $p = \phi_x$ and rewrite (1.4a)-

(1.4b) as follows:

$$u_t + \left(\frac{u^2}{2}\right)_x - p = 0, \tag{5.2a}$$

$$p - \phi_x = 0, \tag{5.2b}$$

$$p_x - \phi - u = 0. \tag{5.2c}$$

Then, the scheme is defined as follows: find $u_h, p_h, \phi_h \in V_h^k$ such that

$$\int_{I_j} (u_h)_t \rho \, dx - \int_{I_j} \frac{u_h^2}{2} \rho_x \, dx + \widehat{\frac{u_h^2}{2}} \rho \, \big|_{\partial I_j} - \int_{I_j} p_h \rho \, dx = 0, \tag{5.3a}$$

$$\int_{I_j} p_h \gamma \, dx + \int_{I_j} \phi_h \gamma_x \, dx - \widehat{\phi_h} \gamma \, \big|_{\partial I_j} = 0, \tag{5.3b}$$

$$- \int_{I_j} p_h q_x \, dx + \widehat{p_h} q \, \big|_{\partial I_j} - \int_{I_j} (\phi_h + u_h) q \, dx = 0, \tag{5.3c}$$

$$\int_{I_j} (u_h - u)|_{t=0} v \, dx = 0, \tag{5.3d}$$

for all test functions ρ, γ, q, v in the finite element space V_h^k. The choice for numerical fluxes $\widehat{u_h^2}$, $\widehat{\phi_h}$, $\widehat{p_h}$ is given by

$$\widehat{u_h^2} = \frac{1}{3}\left((u_h^+)^2 + u_h^+ u_h^- + (u_h^-)^2\right), \tag{5.4a}$$

$$\widehat{\phi_h} = \theta \phi_h^+ + (1-\theta)\phi_h^-, \tag{5.4b}$$

$$\widehat{p_h} = (1-\theta) p_h^+ + \theta p_h^-, \tag{5.4c}$$

where $\theta \in [0, 1/2]$. Here, the numerical fluxes at the endpoints of I can be defined using $U_{1/2}^- := U_{N+1/2}^-$ and $U_{N+1/2}^+ := U_{1/2}^+$ where U represents u_h^2, ϕ_h, or p_h. The resulting LDG scheme (5.3) subject to the fluxes (5.4) with $\theta = 1/2$ is called LDG-C.

For discontinuous solutions, an entropy flux for $\widehat{u_h^2}$ is needed in order to capture the entropy solution. One well-known choice is the Lax-Friedrich flux of the form

$$\widehat{u_h^2} = \frac{1}{2}\left((u_h^-)^2 + (u_h^+)^2 - \sigma(u_h^+ - u_h^-)\right), \ \sigma = 2 \max_{u \in [u_h^-, u_h^+]} |u|, \tag{5.5}$$

with which the resulting LDG scheme is called LDG-D.

In practice, one may adopt an adaptive numerical flux

$$\widehat{u_h^2} = \begin{cases} \frac{1}{3}\left((u_h^+)^2 + u_h^+ u_h^- + (u_h^-)^2\right) & \text{if } |u^+ - u^-| < 10^{-2} \\ \frac{1}{2}\left((u_h^-)^2 + (u_h^+)^2 - 2\sigma(u_h^+ - u_h^-)\right) & \text{otherwise.} \end{cases} \tag{5.6}$$

Here, 10^{-2} may vary as long as it can serve as a shock detector. The resulting scheme is called LDG-Ad.

Remark 5.1.1. *Such a dissipative numerical flux is sufficient for the scheme to capture shocks at the cell interfaces. In practice, shock may well occur in the interior of computational cells, and a limiter is necessary to be imposed, as a result the approximation degenerates to first-order around shocks. In this work we use the TVBM limiter introduced by Cockburn and Shu Cockburn and Shu (1989).*

Before concluding this section, we outline the algorithm to compute the numerical solution.

5.1.2 Algorithm

1. We use U_h to denote the vector containing the degree of freedom for u_h. We compute both ψ and p from solving the coupled system (5.3b), (5.3c)

$$\Phi_h = A_{1-\theta}\Phi_h - U_h, \quad P_h = A_\theta \Phi_h. \tag{5.7}$$

2. Given u_h only, the coupled system is wellposed for $\theta \in [0,1]$ and leads to

$$\Phi_h = -(I - A_{1-\theta}A_\theta)^{-1}U_h, \quad P_h = -A_\theta(I - A_{1-\theta}A_\theta)^{-1}U_h,$$

which when substituted into (5.3a) gives a closed ODE system for u_h:

$$\frac{d}{dt}U_h = -\frac{1}{2}D(u_h^2) + A_\theta(I - A_{1-\theta}A_\theta)^{-1}U_h, \tag{5.8}$$

where $D(u_h^2)$ denotes the vector containing the degree of freedom of the DG differentiation of u_h^2 with the numerical flux (5.4a).

3. We use a time discretization method to solve the obtained semi-discrete system for u_h.

This algorithm indicates that it is important that the coupled system (5.3b), (5.3c) is well-posed, and we will show this in Section 3.

Notation. We use $\|\cdot\|_{m,\Omega}$ as the H^m-norm over domain Ω, and $|\cdot|_{m,\Omega}$ as its semi-norm. For $m = 0$, we simply use $\|\cdot\|_\Omega$ to denote the L^2-norm over domain Ω. We also use the notation $\|\cdot\|_{\infty,\Omega}$ to denote the L^∞ norm over domain Ω. The domain Ω could be a computational cell I_j or a master domain $\hat{I} := [-1, 1]$. If Ω is the whole domain, we do not specify the domain unless necessary. For piecewise smooth function we use the same notation to denote contributions from all cells, for example

$$\|\omega\|_m^2 = \sum_{j=1}^{N} \|\omega\|_{m,I_j}^2.$$

5.2 Analytical properties of the scheme

5.2.1 Existence, uniqueness, and stability

In this section, we prove the existence, uniqueness, and stability of p_h, ϕ_h obtained from (5.3b)-(5.3c) with numerical fluxes (5.4b)-(5.4c), given u_h.

Lemma 5.2.1. *The numerical scheme (5.3b)-(5.3c) with the numerical flux (5.4b)-(5.4c) for any $\theta \in [0, 1]$ satisfies*

$$2\|p_h\|^2 + \|\phi_h\|^2 \le \|u_h\|^2. \tag{5.9}$$

Proof. We choose $\gamma = p_h$ and $q = \phi_h$. Then (5.3b)-(5.3c) gives

$$\begin{aligned}
&\int_{I_j} p_h^2\,dx + \int_{I_j} \phi_h (p_h)_x\,dx - \widehat{\phi_h} p_h \,|_{\partial I_j} = 0,\\
&-\int_{I_j} p_h (\phi_h)_x\,dx + \widehat{p_h}\phi \,|_{\partial I_j} - \int_{I_j} \phi_h^2\,dx = \int_{I_j} u_h \phi_h\,dx.
\end{aligned}$$

Subtracting the two equations above gives

$$\begin{aligned}
&\int_{I_j} \left(p_h^2 + \phi_h^2\right)\,dx\\
&= -\int_{I_j} \phi_h (p_h)_x\,dx + \widehat{\phi_h} p_h \,|_{\partial I_j} - \int_{I_j} p_h (\phi_h)_x\,dx + \widehat{p_h}\phi \,|_{\partial I_j} - \int_{I_j} u_h \phi_h\,dx.
\end{aligned}$$

Take summation over j and use the periodic boundary condition to get

$$\int_I \left(p_h^2 + \phi_h^2\right)\, dx = -\int_I \phi_h (p_h)_x\, dx - \sum_j \left(\widehat{\phi_h}[p_h]\right)_{j+\frac{1}{2}} - \int_I p_h (\phi_h)_x\, dx$$
$$- \sum_j \left(\widehat{p_h}[\phi]\right)_{j+\frac{1}{2}} - \int_I u_h \phi_h\, dx.$$

The first four terms on the right-hand side can be simplified to

$$-\int_I (\phi_h p_h)_x\, dx - \sum_j \left(\widehat{\phi_h}[p_h] + \widehat{p_h}[\phi]\right)_{j+\frac{1}{2}}$$
$$= \sum_j \left([\phi_h p_h] - \widehat{\phi_h}[p_h] - \widehat{p_h}[\phi]\right)_{j+\frac{1}{2}} = 0,$$

because of the choice of numerical fluxes (5.4b)-(5.4c). Therefore, we have that

$$\int_I \left(p_h^2 + \phi_h^2\right)\, dx \le \int_I |u_h \phi_h|\, dx \le \frac{1}{2}\|u_h\|^2 + \frac{1}{2}\|\phi_h\|^2,$$

which proves (5.9). □

Remark 5.2.1. *The inequality (5.9) shows that (5.3b) and (5.3c) produce a unique pair (p_h, ϕ_h) for any given u_h.*

5.2.2 Discrete conservation laws

In this section, we look at the properties of the numerical solution u_h that are analogous to (1.6a)-(1.6b).

Theorem 5.2.2. *For the LDG scheme (5.3) subject to numerical fluxes (5.4) with any $\theta \in [0, 1/2]$, the following relations hold for all $t > 0$:*

$$\int_0^L u_h(t,x)dx = \int_0^L u_h(0,x)dx, \tag{5.10}$$
$$\int_0^L u_h^2(t,x)dx = \int_0^L u_h^2(0,x)dx + (2\theta - 1)\sum_j \int_0^t \left([\phi_h]^2 + [p_h]^2\right)_{j+\frac{1}{2}}\, d\tau. \tag{5.11}$$

Hence the scheme is conservative for $\theta = 1/2$, and the scheme is energy stable for $0 \le \theta < 1/2$.

Remark 5.2.2. *For solutions with discontinuities, we use the numerical flux (5.5) or (5.6) together with (5.4b), (5.4c) with $\theta = 1/2$ so that the quadratic entropy dissipates at admissible discontinuities. Our numerical tests indicate that the choice with $\theta \in (0, 1/2)$ works as well.*

Proof. Because (5.3) holds for any test function in V_h^k, we choose $\rho = 1$ and $\gamma = 1$ in (5.3a)-(5.3b), respectively, to obtain

$$\int_{I_j} (u_h)_t \, dx + \widehat{\frac{u_h^2}{2}} \, |_{\partial I_j} - \widehat{\phi_h} \, |_{\partial I_j} = 0.$$

Take summation over all j and use the periodic boundary condition, we have

$$\frac{d}{dt} \int_I u_h \, dx = -\sum_j \left(\widehat{\frac{u_h^2}{2}} \, |_{\partial I_j} - \widehat{\phi_h} \, |_{\partial I_j} \right) = 0,$$

This proves (5.10).

Next, we choose the test functions $\rho = u_h$, $\gamma = -\phi_h$, and $q = -p_h$ in (5.3a)-(5.3c) to obtain

$$\int_{I_j} (u_h)_t \, u_h \, dx - \int_{I_j} \frac{u_h^2}{2} \, (u_h)_x \, dx + \widehat{\frac{u_h^2}{2}} u_h \, |_{\partial I_j} - \int_{I_j} p_h u_h \, dx = 0, \tag{5.12}$$

$$- \int_{I_j} p_h \phi_h \, dx - \int_{I_j} \phi_h \, (\phi_h)_x \, dx + \widehat{\phi_h} \phi_h \, |_{\partial I_j} = 0, \tag{5.13}$$

$$\int_{I_j} p_h \, (p_h)_x \, dx - \widehat{p_h} p_h \, |_{\partial I_j} + \int_{I_j} (\phi_h + u_h) p_h \, dx = 0, \tag{5.14}$$

Integrating some terms out and adding the above three relations together, we get

$$\int_{I_j} (u_h)_t \, u_h \, dx + \left(\widehat{\frac{u_h^2}{2}} u_h - \frac{u_h^3}{6} \right) |_{\partial I_j} + \left(\widehat{\phi_h} \phi_h - \frac{\phi_h^2}{2} \right) |_{\partial I_j} - \left(\widehat{p_h} p_h - \frac{p_h^2}{2} \right) |_{\partial I_j} = 0.$$

Summing the terms above for all $j = 1, \ldots, N$ and using the periodic boundary condition, we get

$$\begin{aligned}
\frac{1}{2} \frac{d}{dt} \int u_h^2 \, dx &= \sum_j - \left(\widehat{\frac{u_h^2}{2}} u_h - \frac{u_h^3}{6} \right) |_{\partial I_j} - \left(\widehat{\phi_h} \phi_h - \frac{\phi_h^2}{2} \right) |_{\partial I_j} + \left(\widehat{p_h} p_h - \frac{p_h^2}{2} \right) |_{\partial I_j} \\
&= \sum_j \left(\widehat{\frac{u_h^2}{2}} [u_h] - \frac{1}{6} [u_h^3] \right)_{j+\frac{1}{2}} + \left(\widehat{\phi_h} [\phi_h] - \frac{1}{2} [\phi_h^2] \right)_{j+\frac{1}{2}} \\
&\quad - \left(\widehat{p_h} [p_h] - \frac{1}{2} [p_h^2] \right)_{j+\frac{1}{2}} \\
&= \sum_j \left(\widehat{\phi_h} - \{\phi_h\} \right) [\phi_h] + \left(\{p_h\} - \widehat{p_h} \right) [p_h] \\
&= \sum_j \left(\theta - \frac{1}{2} \right) \left([\phi_h]^2 + [p_h]^2 \right)
\end{aligned}$$

which proves (5.11). □

CHAPTER 6. ESTIMATIONS AND NUMERICAL RESULTS

6.1 Error estimations

In this section we estimate the error from approximating p_h, ϕ_h, u_h. We proceed by defining a global projection with some established properties to prove that the errors from approximating p_h and ϕ_h can be controlled by the errors from approximating u_h. Then, we show that the error from approximating u_h is of optimal order. In the use of other boundary conditions, there is a need to refine the proof by carefully estimating the errors induced from boundary terms.

6.1.1 The global projection

Let $\omega \in L^2(I)$, and be smooth on each I_j, say $\omega|_{I_j} \in H^s(I_j)$ for $s \geq k+1$, we define the projection Q_θ such that it satisfies the following properties:

$$\int_{I_j} (Q_\theta \omega)\, v\, dx = \int_{I_j} \omega v\, dx, \quad \forall v \in P^{k-1},\ j = 1, \cdots, N, \tag{6.1a}$$

$$\widehat{Q_\theta \omega}_{j+\frac{1}{2}} = \widehat{\omega}_{j+1/2}, \quad j = 1, \ldots, N, \tag{6.1b}$$

where

$$\widehat{V} := \theta V^+ + (1-\theta) V^-.$$

For $j = N$, we use the periodic extension to define $(Q_\theta \omega)^+_{N+1/2}$, in order to be consistent with the numerical flux defined in (5.4).

We first show that the projection $Q_\theta w$ is well defined.

Lemma 6.1.1. *The projection Q_θ satisfying (6.1) is uniquely defined for either $\theta \neq \frac{1}{2}$ or $\theta = \frac{1}{2}$ with k even and N odd.*

Proof. Let $\{\psi_l\}_{l=0}^k$ be a set of orthogonal Legendre polynomials on $[-1,1]$ of degree up to k. We can write the projection Q_θ of $\omega \in H^{k+1}(I)$ on each cell I_j as

$$(Q_\theta \omega)(x_j + \frac{h}{2}\xi) = \sum_{l=0}^{k} a_l^j \psi_l(\xi), \quad \xi \in [-1,1].$$

With $v = \psi_i$, the condition (6.1a) gives

$$a_i^j = \frac{2i+1}{2} \int_{-1}^{1} \omega(x_j + \frac{h}{2}\xi)\psi_i(\xi)\, d\xi, \quad i = 0, \dots, k-1, \tag{6.2}$$

where we have used $\int_{-1}^{1} \psi_i^2(\xi)\, d\xi = \frac{2}{2i+1}$.

It remains to determine a_k^j for $j = 1, \dots, N$. Since $\psi_l(\pm 1) = (\pm 1)^l$, the condition (6.1b) gives

$$\theta \left(\sum_{l=0}^{k} a_l^{j+1} (-1)^l \right) + (1-\theta) \left(\sum_{l=0}^{k} a_l^j \right) = \hat{\omega}(x_{j+\frac{1}{2}}), \quad j = 1, \dots, N. \tag{6.3}$$

Because ω is periodic, we require that

$$\sum_{l=0}^{k} a_l^{N+1} \psi_l(\xi) = \sum_{l=0}^{k} a_l^1 \psi_l(\xi), \quad \forall \xi \in [-1,1].$$

which allows us to write the system (6.3) as

$$\begin{bmatrix} 1-\theta & (-1)^k\theta & 0 & \cdots & 0 \\ 0 & 1-\theta & (-1)^k\theta & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ (-1)^k\theta & 0 & 0 & \cdots & 1-\theta \end{bmatrix} \cdot \begin{pmatrix} a_k^1 \\ a_k^2 \\ \vdots \\ a_k^N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}, \tag{6.4}$$

where $b_j = \hat{\omega}(x_{j+\frac{1}{2}}) - \theta \left(\sum_{l=0}^{k-1} a_l^{j+1} (-1)^l \right) - (1-\theta) \left(\sum_{l=0}^{k-1} a_l^j \right)$. The determinant of the coefficient matrix A above is given by $(1-\theta)^N + (-1)^{N+1+kN}\theta^N$, which is non-zero for all $\theta \neq \frac{1}{2}$. When $\theta = \frac{1}{2}$, the determinant is non-zero whenever N is odd and k is even. This proves the lemma. □

Lemma 6.1.2. *For $\omega|_{I_j} \in H^{k+1}(I_j)$ for $j = 1 \cdots, N$, we have the following projection error*

$$\|Q_\theta \omega - \omega\| \leq C h^{k+1} |\omega|_{k+1}, \tag{6.5}$$

where C depends on $k \geq 1$ and θ.

Proof. The proof is carried out in two steps.

Step 1. We first establish the following inequality

$$\|(Q_\theta - I)\omega\|^2 \le Ch \sum_{j=1}^{N} \|\tilde{\omega}^j\|_{1,\hat{I}}^2, \tag{6.6}$$

where

$$\tilde{\omega}^j(\xi) := \omega(x_j + \frac{h}{2}\xi), \quad \xi \in [-1,1] = \hat{I}. \tag{6.7}$$

By the Cauchy-Schwarz inequality, we have from (6.2) that for $j = 1, \ldots, N$,

$$|a_i^j|^2 \le \frac{2i+1}{2} \|\tilde{w}^j\|_{0,\hat{I}}^2, \quad i = 0, \ldots, k-1. \tag{6.8}$$

Hence

$$\sum_{j=0}^{N} \sum_{i=0}^{k-1} |a_i^j|^2 \le (k - 1/2) \sum_{j=1}^{N} \|\tilde{\omega}^j\|_{0,\hat{I}}^2.$$

From (6.4) of the form

$$a_k = A^{-1} b,$$

where $a_k = [a_k^1, \ldots, a_k^N]^T$ and $b = [b_1, \ldots, b_N]^T$, it follows that

$$\begin{aligned}
\sum_{j=1}^{N} (a_k^j)^2 = b^T (A^{-1})^T A^{-1} b &\le C \sum_{j=1}^{N} (b_j)^2 \\
&\le C \sum_{j=1}^{N} \left[(\tilde{\omega}^j(1))^2 + \sum_{l=0}^{k-1} (a_l^{j+1})^2 + \sum_{l=0}^{k-1} (a_l^j)^2 \right] \\
&\le C \sum_{j=1}^{N} \|\tilde{\omega}^j\|_{1,\hat{I}}^2,
\end{aligned}$$

where we have used the Sobolev inequality $|\tilde{\omega}^j|_{\infty,\hat{I}} \le C \|\tilde{\omega}^j\|_{1,\hat{I}}$. Hence,

$$\begin{aligned}
\|Q_\theta \omega\|^2 &= \sum_{j=1}^{N} \|Q_\theta \omega\|_{0,I_j}^2 \\
&= \sum_{j=1}^{N} \left[\frac{h}{2} \sum_{l=0}^{k-1} (a_l^j)^2 \|\psi_l\|_{0,\hat{I}}^2 + \frac{h}{2} (a_k^j)^2 \|\psi_k\|_{0,\hat{I}}^2 \right] \\
&\le h \sum_{j=1}^{N} \left[\sum_{i=0}^{k-1} |a_i^j|^2 + (a_k^j)^2 \right] \le Ch \sum_{j=1}^{N} \|\tilde{\omega}^j\|_{1,\hat{I}}^2.
\end{aligned}$$

Step 2. For any $v \in V_h^k(I)$, we have that $Q_\theta v = v$. Therefore, using (6.6) we have

$$\begin{aligned}\|Q_\theta\omega - \omega\|^2 &= \|(Q_\theta - I)(\omega - v)\|^2 \\ &\le Ch\sum_{j=1}^{N} \|\tilde{\omega}^j - \hat{v}^j\|_{1,\hat{I}}^2.\end{aligned}$$

The left hand sides does not depend on v at all, we then have

$$\begin{aligned}\|Q_\theta\omega - \omega\|^2 &\le Ch\sum_{j=1}^{N} \inf_{\hat{v}^j \in P^k[-1,1]} \|\tilde{\omega}^j - \hat{v}^j\|_{1,\hat{I}}^2 \\ &\le Ch\sum_{j=1}^{N} |\tilde{\omega}^j|_{k+1,\hat{I}}^2 =: Ch^{2k+2}|\omega|_{k+1}^2,\end{aligned}$$

where the Bramble-Hilbert lemma (ref. Ciarlet (1978)) has been used. The proof of (6.5) is complete. □

Lemma 6.1.3. *For $k \ge 1$ the following inequality holds,*

$$\sum_{j=1}^{N} \left|(\omega - Q_\theta\omega)(x^-_{j+\frac{1}{2}})\right|^2 \le C|\omega|_{k+1}^2 h^{2k+1}. \tag{6.9}$$

The constant C depends on k and θ.

Proof. On each interval I_j, using the orthogonality relation (6.1a), we have

$$\begin{aligned}\omega(x)|_{I_j} := \tilde{\omega}^j(\xi) &= \sum_{l=0}^{\infty} \omega_l^j \psi_l(\xi), \\ Q_\theta\omega(x)|_{I_j} := \widetilde{(Q_\theta\omega)}^j(\xi) &= \sum_{l=0}^{k-1} \omega_l^j \psi_l(\xi) + \alpha_k^j \psi_k(\xi).\end{aligned}$$

Hence, by $\psi_l(1) = 1$, we have

$$\left|(\omega - Q_\theta\omega)(x^-_{j+\frac{1}{2}})\right|^2 \le 2\left|\sum_{l=k}^{\infty} \omega_l^j\right|^2 + 2|\alpha_k^j|^2. \tag{6.10}$$

To control the first term on the right-hand side of (6.10), we consider the following expression

$$\partial_\xi \tilde{\omega}^j(\xi) = \sum_{l=0}^{\infty} \beta_l^j \psi_l(\xi). \tag{6.11}$$

Following the idea in Castillo et al. (2002), we integrate (6.11) with respect to ξ to get

$$\tilde{\omega}^j(\xi) = \tilde{\omega}^j(-1) + \sum_{l=0}^{\infty} \beta_l^j \int_{-1}^{\xi} \psi_l(\nu) d\nu.$$

Using the property of Legendre polynomials

$$\int_{-1}^{\xi} \psi_i(\nu)d\nu = \frac{1}{2i+1}\left(\psi_{i+1}(\xi) - \psi_{i-1}(\xi)\right),$$

we can write

$$\tilde{\omega}^j(\xi) = \tilde{\omega}^j(-1) + \left(\beta_0^j - \frac{\beta_1^j}{3}\right)\psi_0(\xi) + \sum_{l=1}^{\infty}\left(\frac{\beta_{l-1}^j}{2l-1} - \frac{\beta_{l+1}^j}{2l+3}\right)\psi_l(\xi).$$

Therefore,

$$\omega_i^j = \left(\frac{\beta_{i-1}^j}{2i-1} - \frac{\beta_{i+1}^j}{2i+3}\right), \quad i \geq 1.$$

Thus,

$$\sum_{l=k}^{\infty}\omega_l^j = \left(\frac{\beta_{k-1}^j}{2k-1} + \frac{\beta_k^j}{2k+1}\right),$$
$$\sum_{l=k}^{\infty}\omega_l^{j+1}(-1)^l = (-1)^k\left(\frac{\beta_{k-1}^{j+1}}{2k-1} - \frac{\beta_k^{j+1}}{2k+1}\right).$$

These ensure the following estimates

$$\left|\sum_{l=k}^{\infty}\omega_l^j\right|^2 \leq \frac{1}{2k-1}\left(\frac{2(\beta_{k-1}^j)^2}{2k-1} + \frac{2(\beta_{k+1}^j)^2}{2k+1}\right) \leq \frac{1}{2k-1}\|\partial_\xi\tilde{\omega}^j\|_{0,\hat{I}}^2 \tag{6.12a}$$

$$\left|\sum_{l=k}^{\infty}\omega_l^{j+1}(-1)^l\right|^2 \leq \frac{1}{2k-1}\|\partial_\xi\tilde{\omega}^{j+1}\|_{0,\hat{I}}^2. \tag{6.12b}$$

The second term on the right-hand side of (6.10) is determined by (6.1b), i.e.,

$$\theta\alpha_k^{j+1}(-1)^k + (1-\theta)\alpha_k^j = \theta\left(\sum_{l=k}^{\infty}\omega_l^{j+1}(-1)^l\right) + (1-\theta)\left(\sum_{l=k}^{\infty}\omega_l^j\right), \tag{6.13}$$

where we have used $\hat{\omega}(x_{j+1/2}) = \theta\tilde{\omega}^{j+1}(-1) + (1-\theta)\tilde{\omega}^j(1)$. We then have from (6.12) and (6.13) that

$$\begin{aligned}\sum_{j=1}^{N}|\alpha_k^j|^2 &\leq C\sum_{j=1}^{N}\left(\left|\sum_{l=k}^{\infty}\omega_l^{j+1}(-1)^l\right|^2 + \left|\sum_{l=k}^{\infty}\omega_l^j\right|^2\right)\\ &\leq C\sum_{j=1}^{N}\|\partial_\xi\tilde{\omega}^{j+1}\|_{0,\hat{I}}^2 + \|\partial_\xi\tilde{\omega}^j\|_{0,\hat{I}}^2\\ &\leq C\sum_{j=1}^{N}\|\partial_\xi\tilde{\omega}^j\|_{0,\hat{I}}^2.\end{aligned}$$

Insertion of these estimates back into (6.10) yields

$$\sum_{j=1}^{N}\left|(\omega-Q_\theta\omega)(x^-_{j+\frac{1}{2}})\right|^2 \le C\sum_{j=1}^{N}\|\partial_\xi\tilde{\omega}^j\|^2_{0,\hat{I}}.$$

Recall that $Q_\theta v = v$ for any $v \in P^k$, we proceed

$$\begin{aligned}
\sum_{j=1}^{N}\left|(\omega-Q_\theta\omega)(x^-_{j+\frac{1}{2}})\right|^2 &\le C\sum_{j=1}^{N}\inf_{\tilde{v}\in P^k}\|\partial_\xi\tilde{\omega}^j-\partial_\xi\tilde{v}\|^2_{0,\hat{I}}\\
&= C\sum_{j=1}^{N}\inf_{\tilde{p}\in P^{k-1}}\|\partial_\xi\tilde{\omega}^j-\tilde{p}\|^2_{0,\hat{I}}\\
&\le C\sum_{j=1}^{N}|\partial_\xi\tilde{\omega}^j|^2_{k,\hat{I}}\\
&= \left(\frac{h}{2}\right)^{2k+2}\left(\frac{h}{2}\right)^{-1}C|\omega|^2_{k+1} \le C|\omega|^2_{k+1}h^{2k+1}.
\end{aligned}$$

□

We will use the error estimates obtained in Lemmas 6.1.2-6.1.3 to estimate the error of the computed solution. Moreover, for any $w \in V_h^k$, we utilize the following inverse properties which can be easily derived from the classical ones (see e.g., Ciarlet (1978)),

$$\|\partial_x w\| \le Ch^{-1}\|w\|, \tag{6.14a}$$

$$\|w\|_{\Gamma_h} \le Ch^{-1/2}\|w\|, \tag{6.14b}$$

$$\|w\|_\infty \le Ch^{-1/2}\|w\|, \tag{6.14c}$$

where

$$\|w\|^2_{\Gamma_h} := \sum_{j=1}^{N}\left(\left|w^-_{j+1/2}\right|^2+\left|w^+_{j+1/2}\right|^2\right).$$

The constant C is independent of w and h.

Remark 6.1.1. *Here the estimates for inverse inequalities are valid for piecewise polynomials; the proof usually uses the equivalence of norms for finite dimensional problems and some scaling techniques. It is often desired that the constant is as small as possible Warburton and Hesthaven (2003). Here we just list these results without any further specification of the constants.*

6.1.2 Auxiliary results

Lemma 6.1.4. *Let (u, p, ϕ) be the exact solution of the system (5.2). Let (u_h, p_h, ϕ_h) be obtained from (5.3) with the choice of fluxes (5.4). Let $\theta \in [0, 1]$ be such that both Q_θ and $Q_{1-\theta}$ are uniquely defined, then the following inequality holds for all $t > 0$.*

$$\|Q_{1-\theta}p - p_h\|^2 + \|Q_\theta\phi - \phi_h\|^2 \leq \|Q_{1-\theta}p - p\|^2 + 2\|Q_\theta\phi - \phi\|^2 + 2\|u - u_h\|^2. \tag{6.15}$$

Proof. Since the scheme with fluxes (5.4) is consistent, (5.3b)-(5.3c) also hold for (u, p, ϕ). In other words,

$$\int_{I_j} p\gamma + \int_{I_j} \phi\gamma_x - \phi\gamma\,|_{\partial I_j} = 0, \tag{6.16a}$$

$$-\int_{I_j} pq_x + pq\,|_{\partial I_j} - \int_{I_j} (\phi + u)q = 0, \tag{6.16b}$$

Subtracting (5.3b)-(5.3c) from (6.16a)-(6.16b), we get the error equations

$$\int_{I_j} (p - p_h)\gamma + \int_{I_j} (\phi - \phi_h)\gamma_x - \left(\phi - \widehat{\phi_h}\right)\gamma\,|_{\partial I_j} = 0, \tag{6.17a}$$

$$-\int_{I_j} (p - p_h)q_x + (p - \widehat{p_h})\,q\,|_{\partial I_j} - \int_{I_j} (\phi - \phi_h)q = \int_{I_j} (u - u_h)q. \tag{6.17b}$$

Define $\epsilon_p = Q_{1-\theta}p - p$, $w_p = Q_{1-\theta}p - p_h$, $\widehat{\epsilon_p} = \widehat{Q_{1-\theta}p} - p$, and $\widehat{w_p} = \widehat{Q_{1-\theta}p} - \widehat{p_h}$. (Similar definition can be given for ϵ_ϕ, w_ϕ, $\widehat{\epsilon_\phi}$, and $\widehat{w_\phi}$ associated with Q_θ.) We then choose $\gamma = w_p$, $q = w_\phi$ and take the summation of (6.17) over j to get

$$\begin{aligned}
&\sum_j \int_{I_j} (w_p - \epsilon_p)w_p + \sum_j \int_{I_j} (w_\phi - \epsilon_\phi)(w_p)_x + \sum_j (\widehat{w_\phi} - \widehat{\epsilon_\phi})\,[w_p]_{j+\frac{1}{2}} = 0,\\
&-\sum_j \int_{I_j} (w_p - \epsilon_p)(w_\phi)_x - \sum_j (\widehat{w_p} - \widehat{\epsilon_p})\,[w_\phi]_{j+\frac{1}{2}} - \sum_j \int_{I_j} (w_\phi - \epsilon_\phi)w_\phi\\
&\quad = \int_I (u - u_h)w_\phi.
\end{aligned}$$

Take the difference of both equations, we get

$$\int_I w_p^2 + \int_I w_\phi^2 = \int_I \epsilon_p w_p + \int_I \epsilon_\phi w_\phi + \varrho_1 + \varrho_2 + \varrho_3 - \int_I (u - u_h)w_\phi, \tag{6.18}$$

where

$$\varrho_1 = -\sum_j \int_{I_j} (w_\phi (w_p)_x + w_p (w_\phi)_x) - \sum_j \widehat{w_\phi}[w_p]_{j+\frac{1}{2}} - \sum_j \widehat{w_p}[w_\phi]_{j+\frac{1}{2}},$$
$$\varrho_2 = \sum_j \int_{I_j} \epsilon_\phi (w_p)_x + \sum_j \widehat{\epsilon_\phi}[w_p]_{j+\frac{1}{2}},$$
$$\varrho_3 = \sum_j \int_{I_j} \epsilon_p (w_\phi)_x + \sum_j \widehat{\epsilon_p}[w_\phi]_{j+\frac{1}{2}}.$$

First, note that $w_\phi (w_p)_x + w_p (w_\phi)_x = (w_p w_\phi)_x$, so

$$\varrho_1 = \sum_j [w_p w_\phi]_{j+\frac{1}{2}} - \sum_j \widehat{w_\phi}[w_p]_{j+\frac{1}{2}} - \sum_j \widehat{w_p}[w_\phi]_{j+\frac{1}{2}} = 0,$$

with the choice of numerical fluxes (5.4b)-(5.4c). As for ϱ_2, the property (6.1a) of Q_θ gives

$$\sum_j \int_{I_j} \epsilon_\phi (w_p)_x = \sum_j \int_{I_j} (Q_\theta \phi - \phi)(w_p)_x = 0,$$

since $(w_p)_x$ is in P^{k-1}. On the other hand, the propery (6.1b) of Q_θ gives

$$\sum_j \widehat{\epsilon_\phi}[w_p]_{j+\frac{1}{2}} = \sum_j \left(\widehat{Q_\theta \phi} - \phi \right) [w_p]_{j+\frac{1}{2}} = 0.$$

Similarly, the term ϱ_3 vanishes by the properties (6.1) of $Q_{1-\theta}$.

Using $\varrho_i = 0$ for $i = 1, 2, 3$ and the Young's inequality $ab \le \frac{a^2}{2\mu} + \frac{\mu b^2}{2}$ with $\mu = 1$ for the first term and $\mu = \frac{1}{2}$ for the last two terms in (6.18), we get

$$\frac{1}{2}\|w_p\|^2 + \frac{1}{2}\|w_\phi\|^2 \le \frac{1}{2}\|\epsilon_p\|^2 + \|\epsilon_\phi\|^2 + \|u - u_h\|^2, \tag{6.19}$$

which proves (6.15). □

6.1.3 Main theorem

Theorem 6.1.5. *Let $u \in L^\infty((0,T]; H^s(I)), s \ge k+1$, be the smooth solution to (1.4), for $0 < t < T$. If k is even, then the numerical solution, u_h, obtained from the scheme (5.3) and the numerical fluxes (5.4) satisfies*

$$\sup_{t\in[0,T]} \|u(t,\cdot) - u_h(t,\cdot)\| \le C\|u\|_{L^\infty((0,T];H^{k+1}(I))} h^{k+1}. \tag{6.20}$$

The constant C may depend on T and the data given, but is independent of the mesh size.

Proof. Since the scheme (5.3) with fluxes (5.4) is consistent, (5.3a) also holds for (u, p, ϕ). In other words,

$$\int_{I_j} u_t \rho - \int_{I_j} \frac{u^2}{2} \rho_x + \widehat{\frac{u^2}{2}} \rho \,|_{\partial I_j} - \int_{I_j} p\rho = 0. \tag{6.21}$$

Define $w = Q_{1/2}u - u_h$ and $\epsilon = Q_{1/2}u - u$. We have that $u - u_h = w - \epsilon$. Subtracting (5.3a) from (6.21) and choose $\rho = w$, we get

$$\int_{I_j} w_t w = \int_{I_j} \epsilon_t w + \int_{I_j} \left(\frac{u^2}{2} - \frac{u_h^2}{2} \right) w_x - \left(\frac{u^2}{2} - \widehat{\frac{u_h^2}{2}} \right) w \,|_{\partial I_j} + \int_{I_j} e_p w,$$

where $e_p = p - p_h$.

Take summation over all j and introduce $\{u_h\}^2/2$ into the third term on the right-hand side to get

$$\begin{aligned} \int_I w_t w = \int_I \epsilon_t w + \sum_j \int_{I_j} \left(\frac{u^2}{2} - \frac{u_h^2}{2} \right) w_x + \sum_j \left(\frac{u^2}{2} - \frac{\{u_h\}^2}{2} \right) [w]_{j+\frac{1}{2}} \\ + \sum_j \left(\frac{\{u_h\}^2}{2} - \widehat{\frac{u_h^2}{2}} \right) [w]_{j+\frac{1}{2}} + \int_I e_p w. \end{aligned}$$

Using the identity $A^2/2 - B^2/2 = A(A-B) - (A-B)^2/2$, we get

$$\begin{aligned} \int_I w_t w = \int_I \epsilon_t w + \sum_j \int_{I_j} u\,(u - u_h)\, w_x - \frac{1}{2} \sum_j \int_{I_j} (u - u_h)^2 w_x \\ + \sum_j u\,(u - \{u_h\})\,[w]_{j+\frac{1}{2}} - \frac{1}{2} \sum_j (u - \{u_h\})^2 [w]_{j+\frac{1}{2}} \\ + \sum_j \left(\frac{\{u_h\}^2}{2} - \widehat{\frac{u_h^2}{2}} \right) [w]_{j+\frac{1}{2}} + \int_I e_p w. \end{aligned}$$

Let $\{w\} = \{Q_{1/2}u\} - \{u_h\}$ and $\{\epsilon\} = \{Q_{1/2}u\} - u$, we can write

$$\int_I w_t w = \int_I \epsilon_t w + \tau_1 + \tau_2 + \tau_3 + \tau_4 + \tau_5 + \int_I e_p w, \tag{6.22}$$

where

$$\tau_1 = \sum_j \int_{I_j} uww_x + \sum_j u\{w\}[w]_{j+\frac{1}{2}},$$
$$\tau_2 = -\sum_j \int_{I_j} u\epsilon w_x - \sum_j u\{\epsilon\}[w]_{j+\frac{1}{2}},$$
$$\tau_3 = -\frac{1}{2}\sum_j \int_{I_j} w^2 w_x - \frac{1}{2}\sum_j \{w\}^2[w]_{j+\frac{1}{2}},$$
$$\tau_4 = \sum_j \int_{I_j} w\epsilon w_x - \frac{1}{2}\sum_j \int_{I_j} \epsilon^2 w_x + \sum_j \{w\}\{\epsilon\}[w]_{j+\frac{1}{2}} - \frac{1}{2}\sum_j \{\epsilon\}^2[w]_{j+\frac{1}{2}},$$
$$\tau_5 = \sum_j \left(\frac{\{u_h\}^2}{2} - \frac{\widehat{u_h^2}}{2}\right)[w]_{j+\frac{1}{2}}.$$

Note that

$$\begin{aligned}\tau_1 &= \sum_j \int_{I_j} u\left(\frac{w^2}{2}\right)_x + \sum_j u\{w\}[w]_{j+\frac{1}{2}} \\ &= -\sum_j u\left[\frac{w^2}{2}\right]_{j+\frac{1}{2}} - \sum_j \int_{I_j} u_x\left(\frac{w^2}{2}\right) + \sum_j u\{w\}[w]_{j+\frac{1}{2}} \\ &= -\sum_j \int_{I_j} u_x\left(\frac{w^2}{2}\right) \le \frac{1}{2}\|u_x\|_\infty \|w\|^2.\end{aligned}$$

As for τ_2, we write $u(x) = u(x_j) + u'(x_j^*)(x - x_j)$ for all $x \in I_j$ where x_j^* is between x and x_j. Therefore,

$$\begin{aligned}\left|-\sum_j \int_{I_j} u\epsilon w_x - \sum_j u\{\epsilon\}[w]_{j+\frac{1}{2}}\right| &\le \left|-\sum_j u(x_j)\int_{I_j} \epsilon w_x - \sum_j u'(x_j^*)\int_{I_j} \epsilon(x - x_j)w_x\right| + |0| \\ &\le \left|\sum_j u(x_j)\int_{I_j} \epsilon w_x\right| + \sum_j |u'(x_j^*)| \int_{I_j} |\epsilon h w_x| \\ &\le |0| + \frac{1}{2}\|u_x\|_\infty \left(\|\epsilon\|^2 + h^2\|w_x\|^2\right) \\ &\le C(h^{2k+2} + \|w\|^2),\end{aligned}$$

because of the projection properties (6.1), the inverse property (6.14a), and Lemma 6.1.2.

For τ_3, we can show that

$$\begin{aligned}&-\frac{1}{2}\sum_j \int_{I_j}\left(\frac{w^3}{3}\right)_x - \frac{1}{2}\sum_j \{w\}^2[w]_{j+\frac{1}{2}} = \frac{1}{2}\sum_j \left[\frac{w^3}{3}\right] - \frac{1}{2}\sum_j \{w\}^2[w]_{j+\frac{1}{2}} \\ &= \frac{1}{24}\sum_j [w]_{j+\frac{1}{2}}^3 \le Ch^{-1}\|w\|_\infty\|w\|^2 \le Ch^{-3/2}\|w\|^3,\end{aligned}$$

by the inverse properties (6.14b)-(6.14c).

From the inverse properties (6.14a) and (6.14c), it follows that

$$\|w_x\|_\infty \le Ch^{-3/2}\|w\|,$$

with which we are able to estimate terms in τ_4 by using Lemma 6.1.2,

$$\sum_j \int_{I_j} w\epsilon w_x \le C\|w\|\|\epsilon\|\|w_x\|_\infty \le Ch^{k+1-3/2}\|w\|^2 = Ch^{k-1/2}\|w\|^2,$$
$$\frac{1}{2}\sum_j \int_{I_j} \epsilon^2 w_x \le \frac{1}{2}\|w_x\|_\infty\|\epsilon\|^2 \le Ch^{2k+1/2}\|w\|.$$

As for the remaining terms in τ_4,

$$\sum_j \{w\}\{\epsilon\}[w]_{j+\frac{1}{2}} - \frac{1}{2}\sum_j \{\epsilon\}^2[w]_{j+\frac{1}{2}} = 0$$

because of the projection property (6.1b).

Finally, using the fact that $\{u_h\}^2/2 - \widehat{u_h^2}/2 = -[u_h]^2/24$, and $[u_h] = [u - u_h] = [w] - [\epsilon]$, we have

$$\begin{aligned}
\tau_5 = \sum_j \left(\frac{\{u_h\}^2}{2} - \frac{\widehat{u_h^2}}{2}\right)[w]_{j+\frac{1}{2}} &= \sum_j -\frac{1}{24}[u_h]^2[w]_{j+\frac{1}{2}}, \\
&= \sum_j -\frac{1}{24}[w]^3_{j+\frac{1}{2}} + \frac{1}{12}[\epsilon][w]^2_{j+\frac{1}{2}} - \frac{1}{24}[\epsilon]^2[w]_{j+\frac{1}{2}} \\
&\le C\|w\|_\infty(\|w\|^2_{\Gamma_h} + \|\epsilon\|_{\Gamma_h}\|w\|_{\Gamma_h} + \|\epsilon\|^2_{\Gamma_h}) \\
&\le Ch^{-1/2}\|w\|(h^{-1}\|w\|^2 + h^{k+1/2}h^{-1/2}\|w\| + h^{2k+1}) \\
&\le C(h^{2k+2} + \|w\|^2 + h^{-3/2}\|w\|^3),
\end{aligned}$$

where we have used the inverse properties (6.14b)-(6.14c) and Lemma 6.1.3.

The results from τ_1 to τ_5 and (6.22) give

$$\frac{d}{dt}\|w\|^2 \le C_1\left(h^{2k+2} + \|w\|^2 + h^{-3/2}\|w\|^3\right). \tag{6.23}$$

We note that

$$\|w(t=0,\cdot)\|^2 \le C_2 h^{2k+2}, \tag{6.24}$$

because $w(0,\cdot)=\epsilon(0,\cdot)+(u_0-u_h(0,\cdot))$, where $u_h(0,\cdot)$ is prepared using a standard L^2-projection from the given initial data. To solve (6.23) with initial data (6.24), we introduce

$$G(t)=h^{2k+2}+\int_0^t \|w(\tau,\cdot)\|^2+h^{-3/2}\|w(\tau,\cdot)\|^3\,d\tau. \tag{6.25}$$

With this and (6.24), we can write

$$\|w(t,\cdot)\|^2\le CG(t). \tag{6.26}$$

Hence, for $C_*=C\max\{1,\sqrt{C}\}$,

$$G'(t)\le C_*\left(G(t)+h^{-3/2}G(t)^{3/2}\right). \tag{6.27}$$

Integrate (6.27) to get

$$F\left(\frac{G}{G(0)}\right)\le C_*T, \tag{6.28}$$

where

$$F(\eta)=\int_1^\eta \frac{1}{\xi+h^{-3/2}\sqrt{G(0)}\xi^{3/2}}\,d\xi=\int_1^\eta \frac{1}{\xi+h^{k-1/2}\xi^{3/2}}\,d\xi. \tag{6.29}$$

For $k\ge 1$, we have that $F'(\eta)$ is uniformly (with respect to h) positive and bounded above by 1 for all $\eta>1$. Thus, there exists $\tilde{C}$ such that $F(\tilde{C})=C_*T$ for given $T>0$. Therefore, we have that $F\left(\frac{G}{G(0)}\right)\le F(\tilde{C})$ which implies $\frac{G}{G(0)}\le\tilde{C}$. Using this and (6.26), we prove (6.20) as desired. □

6.1.4 Time discretization

We partition the time interval $[0,T]$ into M equal subintervals with boundaries $\{t^n\}, n=0,1,2,\ldots,M$. Set $\Delta t=T/M$ as the time step. In order to preserve both mass and energy at the fully discrete level, we may use the Crank-Nicolson time discretization to find

$$u_h^{n+1}=2u_h^*-u_h^n,$$

where u_h^* is determined by

$$\begin{aligned}
&2\int_{I_j}\frac{u_h^*-u_h^n}{\Delta t}\rho-\int_{I_j}\frac{(u_h^*)^2}{2}\rho_x+\widehat{\frac{(u_h^*)^2}{2}}\rho\,|_{\partial I_j}-\int_{I_j}p_h^*\rho=0,\\
&\int_{I_j}p_h^*\gamma+\int_{I_j}\phi_h^*\gamma_x-\widehat{\phi_h^*}\gamma\,|_{\partial I_j}=0,\\
&-\int_{I_j}p_h^*q_x+\widehat{p_h^*}q\,|_{\partial I_j}-\int_{I_j}(\phi_h^*+u_h^*)\,q=0.
\end{aligned}\tag{6.30}$$

Indeed, this time discretization has the desired and provable properties.

Theorem 6.1.6. *The fully-discrete scheme (6.30) gives solution u_h^n that satisfies*

$$\int_0^L u_h^{n+1}\,dx=\int_0^L u_h^n\,dx,\tag{6.31}$$

$$\int_0^L\left(u_h^{n+1}\right)^2\,dx=\int_0^L(u_h^n)^2\,dx+\Delta t(2\theta-1)\sum_j\left([\phi_h^*]^2+[p_h^*]^2\right)_{j+\frac12},\tag{6.32}$$

for all $0\le n<M$. Here, $\phi_h^*=(\phi_h^{n+1}+\phi_h^n)/2$, and $p_h^*=(p_h^{n+1}+p_h^n)/2$.

Proof. Take the test functions $\rho=1$, $\gamma=1$ in (6.30), adding together, to get

$$\int_{I_j}\frac{u_h^{n+1}-u_h^n}{\Delta t}+\widehat{\frac{(u_h^*)^2}{2}}\,|_{\partial I_j}-\widehat{\phi_h^*}\,|_{\partial I_j}=0,$$

which upon summation over j proves (6.31). Next, we choose the test functions $\rho=u_h^*$, $\gamma=-\phi_h^*$, and $q=-p_h^*$ so that

$$\begin{aligned}
\int_{I_j}\frac{u_h^{n+1}-u_h^n}{\Delta t}u_h^*-\int_{I_j}\frac{(u_h^*)^2}{2}(u_h^*)_x+\widehat{\frac{(u_h^*)^2}{2}}u_h^*\,|_{\partial I_j}-\int_{I_j}p_h^*u_h^*=0,\\
-\int_{I_j}p_h^*\phi_h^*-\int_{I_j}\phi_h^*(\phi_h^*)_x+\widehat{\phi_h^*}\phi_h^*\,|_{\partial I_j}=0,\\
\int_{I_j}p_h^*(p_h^*)_x-\widehat{p_h^*}p_h^*\,|_{\partial I_j}+\int_{I_j}(\phi_h^*+u_h^*)p_h^*=0.
\end{aligned}$$

Summation of the above three equations over j gives

$$\sum_j\int_{I_j}\frac{\left(u_h^{n+1}\right)^2-(u_h^n)^2}{2\Delta t}=\int_0^L\frac{\left(u_h^{n+1}\right)^2-(u_h^n)^2}{2\Delta t}=\left(\theta-\frac12\right)\sum_j\left([\phi_h]^2+[p_h]^2\right),$$

which leads to (6.32). □

Note that the above time discretization is fully nonlinear and requires the costly iteration solver. In practice, one would prefer to use some explicit solver with high order accuracy

for time discretization. In our numerical simulation we choose to use the TVD third-order Runge-Kutta method Gottlieb and Shu (1998) to solve the ODE system of the form $\dot{\mathbf{a}} = \mathfrak{L}(\mathbf{a})$:

$$\begin{aligned}
\mathbf{a}^{(1)} &= \mathbf{a}^n + \Delta t \mathfrak{L}(\mathbf{a}^n), \\
\mathbf{a}^{(2)} &= \frac{3}{4}\mathbf{a}^n + \frac{1}{4}\mathbf{a}^{(1)} + \frac{1}{4}\Delta t \mathfrak{L}(\mathbf{a}^{(1)}), \\
\mathbf{a}^{n+1} &= \frac{1}{3}\mathbf{a}^n + \frac{2}{3}\mathbf{a}^{(2)} + \frac{2}{3}\Delta t \mathfrak{L}(\mathbf{a}^{(2)}),
\end{aligned} \tag{6.33}$$

where $\mathbf{a}^n$ is the coefficient vector of u_h^n.

6.2 Numerical Tests

It is known Fellner and Schmeiser (2004) that one steady solution of system (1.4a)-(1.4b) is given by

$$U_1(x) = \frac{4}{3}\left(e^{-|x|/2} - 1\right). \tag{6.34}$$

The system also has a steady periodic solution of the form

$$U_2(x) = \frac{4}{3}\left(\frac{\cosh\left(\frac{x}{2}\right)}{\cosh\left(\frac{p}{2}\right)} - 1\right), \tag{6.35}$$

for $-p < x < p$ and by periodic continuation with period $2p$. Because the system (1.4a)-(1.4b) is Galilean invariant, a family of traveling-wave solutions (1.4a)-(1.4b) may be obtained from the steady solutions as

$$u(t, x) = U(x - u_0 t) + u_0, \tag{6.36}$$

where U is the steady state solution (6.34) or (6.35). We will use both steady and traveling wave solutions to test our scheme.

Example 1. (Accuracy test) We run the semi-discrete scheme (5.3) and the numerical flux (5.4) with $\theta = 1/2, 0$, along with the third order Runge-Kutta method (6.33) on the steady state problem which has (6.35) as its exact solution. The results for $k = 1, 2, 3, 4$ are given in Tables 6.1 and 6.2 below. Here, we use $\Delta t = 0.001$, final time $t_{\max} = 2$, and $p = 2$. The norms of the error were computed by using the sixteen-point Gauss quadrature rule.

k	N	$\theta = 1/2$					
		L_1	order	L_2	order	L_∞	order
1	10	5.5907e-02		3.3360e-02		4.8122e-02	
	20	2.8223e-02	0.9862	1.6765e-02	0.9926	2.6199e-02	0.8772
	40	1.4137e-02	0.9974	8.3909e-03	0.9986	1.3877e-02	0.9168
	80	7.0694e-03	0.9998	4.1959e-03	0.9999	7.1781e-03	0.9510
2	10	1.6478e-03		1.0558e-03		2.6115e-03	
	20	2.0461e-04	3.0096	1.3282e-04	2.9908	3.8008e-04	2.7805
	40	2.5457e-05	3.0067	1.6615e-05	2.9990	5.3250e-05	2.8355
	80	3.1778e-06	3.0019	2.0768e-06	3.0000	7.2587e-06	2.8750
3	10	2.5013e-04		1.7960e-04		6.4039e-04	
	20	3.1133e-05	3.0062	2.2535e-05	2.9945	9.9254e-05	2.6897
	40	3.8845e-06	3.0026	2.8200e-06	2.9984	1.4970e-05	2.7290
	80	4.8514e-07	3.0013	3.5262e-07	2.9995	2.2034e-06	2.7643
4	10	1.2182e-06		9.6189e-07		4.5036e-06	
	20	3.7401e-08	5.0256	3.0251e-08	4.9908	1.8170e-07	4.6314
	40	1.1545e-09	5.0178	9.4598e-10	4.9990	7.1522e-09	4.6670
	80	3.5841e-11	5.0095	2.9535e-11	5.0013	2.7574e-10	4.6970

Table 6.1 Errors for example 1 (accuracy test) with $\theta = 1/2$.

The results show that the optimal order of accuracy is achieved only when k =even, which is consistent with our theoretical result on the optimal error estimates for k = even. Also such an observation seems unaffected by the choice of $\theta \in [0, 1]$, though we only display results for $\theta = 1/2$ and $\theta = 0$.

Example 2. (Energy-preserving test) We compare the performance of the LDG-C scheme and the LDG-D scheme with $\theta = 1/2$ on the traveling wave version of (6.34), with velocity $u_0 = 1$. The simulation is done on 160 elements with polynomials of degree $k = 4$ over the domain $[-20, 20]$. The time step for the third order Runge-Kutta method (6.33) is $\Delta t = 0.001$, and $t_{\max} = 400$.

In Figure 6.1, we see that both schemes perform well over a short time. However, after a long time ($t = 400$), the LDG-C scheme performs clearly better as we can observe that it produces a smaller phase shift. In terms of L^2-energy, initially, $\|u_h(0, \cdot)\|$ is 2.108223389275528. At $t = 400$, the numerical solution obtained by the LDG-C scheme has L^2 energy $\|u_h(400, \cdot)\| =$ **2.108223**389275528, which agrees with the initial energy up to 6^{th} decimal place. On the other

k	N	$\theta = 0$					
		L_1	order	L_2	order	L_∞	order
1	10	4.4335e-02		3.0487e-02		4.5356e-02	
	20	2.3380e-02	0.9232	1.5731e-02	0.9546	2.3925e-02	0.9227
	40	1.2089e-02	0.9516	8.0245e-03	0.9711	1.1393e-02	1.0704
	80	6.1588e-03	0.9730	4.0569e-03	0.9841	5.6451e-03	1.0131
2	10	1.9403e-04		1.2987e-04		3.0732e-04	
	20	1.7491e-05	3.4716	1.2618e-05	3.3635	3.9729e-05	2.9515
	40	1.4935e-06	3.5498	1.2035e-06	3.3903	5.0581e-06	2.9735
	80	1.3537e-07	3.4637	1.1552e-07	3.3809	6.3888e-07	2.9850
3	10	1.0986e-05		1.0699e-05		4.5712e-05	
	20	1.4206e-06	2.9510	1.3271e-06	3.0111	6.4713e-06	2.8204
	40	1.9201e-07	2.8872	1.6821e-07	2.9800	8.9441e-07	2.8551
	80	2.5077e-08	2.9368	2.1246e-08	2.9850	1.2020e-07	2.8955
4	10	8.5328e-08		5.4694e-08		1.4438e-07	
	20	1.9935e-09	5.4197	1.3489e-09	5.3415	5.4551e-09	4.7261
	40	4.3824e-11	5.5074	3.1240e-11	5.4323	1.9264e-10	4.8236
	80	1.2288e-12	5.1564	8.4612e-13	5.2064	6.3020e-12	4.9339

Table 6.2 Errors for example 1 (accuracy test) with $\theta = 0$.

hand, the LDG-D scheme with $\theta = 1/2$ yields $\|u_h(400,\cdot)\| = \mathbf{2.10}7474302191212$, which agrees with the initial energy up to only 2^{nd} decimal place. Here, the L^2 norms were computed by using the sixteen-point Gauss quadrature rule.

We plot the evolution of the relative energy $\|u_h(0,\cdot)\| - \|u_h(400,\cdot)\|$ in Figure 6.2. In addition to the comparison between LDG-C and LDG-D, we also compare the performance of the flux (5.4) with $\theta = 1/2$ (i.e. LDG-C) and with $\theta = 0$. The result is as we expected: when $\theta = 1/2$, the energy is conserved better than when $\theta \in [0, 1/2)$.

The numerical tests indicate that after long time simulation, phase shift is a main source of error, while the shape of waves remains stable. In order to quantify the shape error we use the formula introduced in Bona et al. (1995)

$$\hat{e}(t,x) = \min_{\xi\in[-0.5,0.5]} \|u_h(t,x) - u(t,x+\xi)\|,$$

for the numerical solution u_h obtained from the LDG-C scheme with $\theta = 1/2$, while $u(t,x)$ is the exact solution. The shape error defined above compares how good the approximation is, modulo the translation group on the periodic domain, and it minimizes the difference between

the numerical approximation and the spatially shifted exact solution. In Figure 6.3, we see that the shape error fluctuates around a constant in time, with a visible periodic behavior. In contrast, the absolute L^2-error grows in time.

In next three examples, we use LDG-Ad on polynomial elements of degree $k = 2$ along with the TVBM limiter introduced in Cockburn and Shu (1989). Here, we use the mesh size $h = 1/16$ for examples 3 and 4, and $h = 1/4$ for example 5. The threshold 10^{-2} in (5.6) depends on the data. It is obtained from numerical experiment. As for the choice of θ, we use $\theta = 1/2$, and we also observe similar results from tests using $\theta \in (0, 1/2)$.

Example 3. We test the conservative scheme for initial data

$$u_0(x) = \begin{cases} 0.5 & 0 \le x \le 1, \\ -x + 1.5, & 1 \le x \le 4, \\ -2.5, & x \ge 4. \end{cases}$$

This initial data has a downward ramp of height 3 and the constant states lying symmetric with respect to $u = -1$, the solution is expected to converge to a stationary solution. In Figure 6.4, we observe a stable pattern formation as analyzed in Fellner and Schmeiser (2004). In our experiment we use a modified initial data in C^2, which agrees with the original data everywhere except for near $x = 1, 4$, so that we can apply directly the TVBM limiter introduced in Cockburn and Shu (1989). Our goal is to observe the stable wave pattern, so the choice of modification is not essential.

Example 4. We consider another initial data of the form

$$u_0(x) = \begin{cases} -0.5 & x \le 8, \\ 15.5 - 2x, & 8 \le x \le 8.5, \\ -1.5, & x \ge 8.5. \end{cases}$$

This example with smaller jump has no stable stationary solution to converge. We plot the computed solution at the different times in Figure 6.5, from which we can see that dispersive effects with oscillations propagate to the left of the ramp as analyzed in Fellner and Schmeiser

(2004).

Example 5. In this example we test interaction of traveling waves. It is known that the interaction of solitons for the KdV equation

$$u_t + uu_x + u_{xxx} = 0,$$

can be illustrated through a family of solutions derived in Hirota (1971). One of them reads as follows

$$u(t,x) = 12\frac{k_1^2 e^{\theta_1} + k_2^2 e^{\theta_2} + 2(k_2-k_1)^2 e^{\theta_1+\theta_2} + a^2(k_2^2 e^{\theta_1} + k_1^2 e^{\theta_2})e^{\theta_1+\theta_2}}{(1+e^{\theta_1}+e^{\theta_2}+a^2 e^{\theta_1+\theta_2})^2},$$

where

$$k_1 = 0.4, \quad k_2 = 0.6, \quad a^2 = \left(\frac{k_1-k_2}{k_1+k_2}\right) = \frac{1}{25},$$
$$\theta_1 = k_1 x - k_1^3 t + x_1, \quad \theta_2 = k_2 x - k_2^3 t + x_2, \quad x_1 = 4, \quad x_2 = 5.$$

Since the BP system is dispersive, and close to the KdV equation in some regime of physical parameters, we use $u(0,x)$ as the initial data for the BP system and run the simulation to observe the interaction of two traveling waves. The result is similar to the KdV case in Yi et al. (2013): the two peaks travel from left to right, and the speed of the tall one is greater than that of the short one; the taller one eventually passes the shorter one. In addition, oscillations develop on the left as time increases. This is similar to the downward ramp of height 2 in Example 4, as shown in Figure 6.6. In fact, if we rescale (t,x) by $(\epsilon t, \epsilon(x+t))$ in the BP system, we obtain

$$\partial_t u + uu_x - u_x + (1-\epsilon^2\partial_x^2)^{-1}u_x = 0,$$

which to the first order leads to

$$u_t + uu_x + \epsilon^2 u_{xxx} = 0.$$

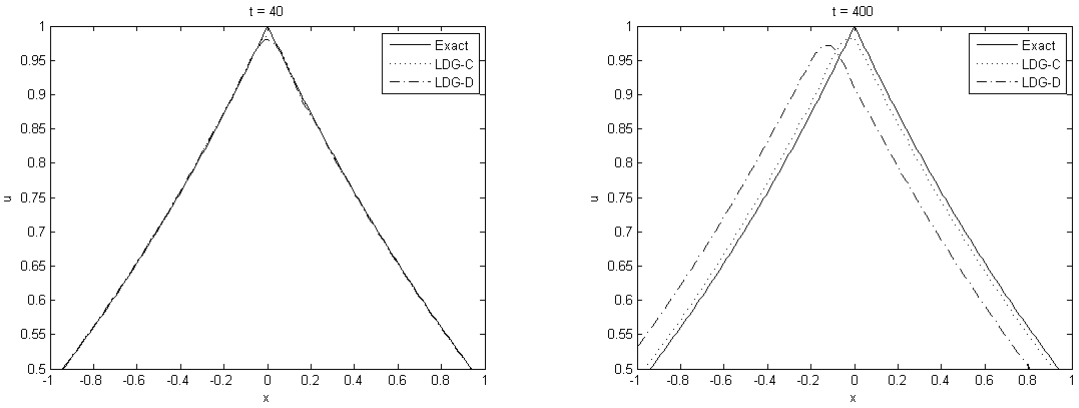

Figure 6.1 Example 2: comparison between the LDG-C and LDG-D scheme with $\theta = 1/2$. Left: $t = 40$. Right: $t = 400$.

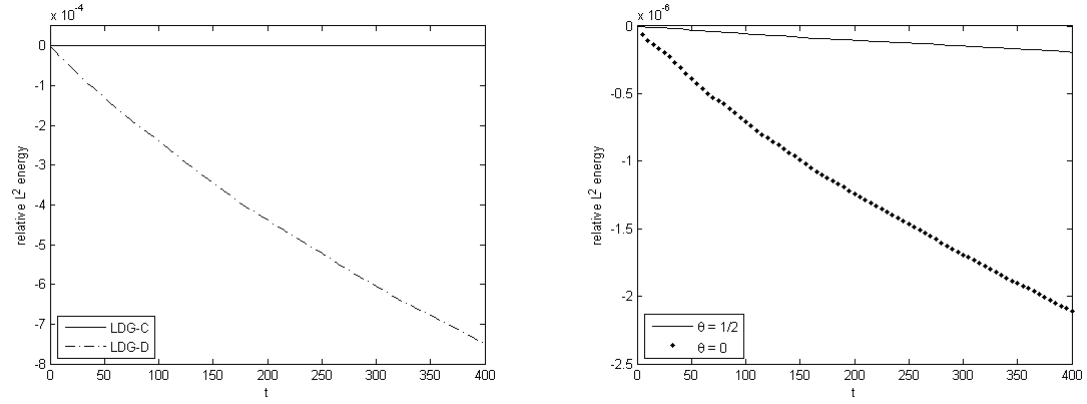

Figure 6.2 Example 2: the evolution of the relative L^2 energy over long-time simulation. Left: comparison between LDG-C and LDG-D with $\theta = 1/2$. Right: comparison between the flux (5.4) with $\theta = 1/2$ (LDG-C) and the flux (5.4) with $\theta = 0$.

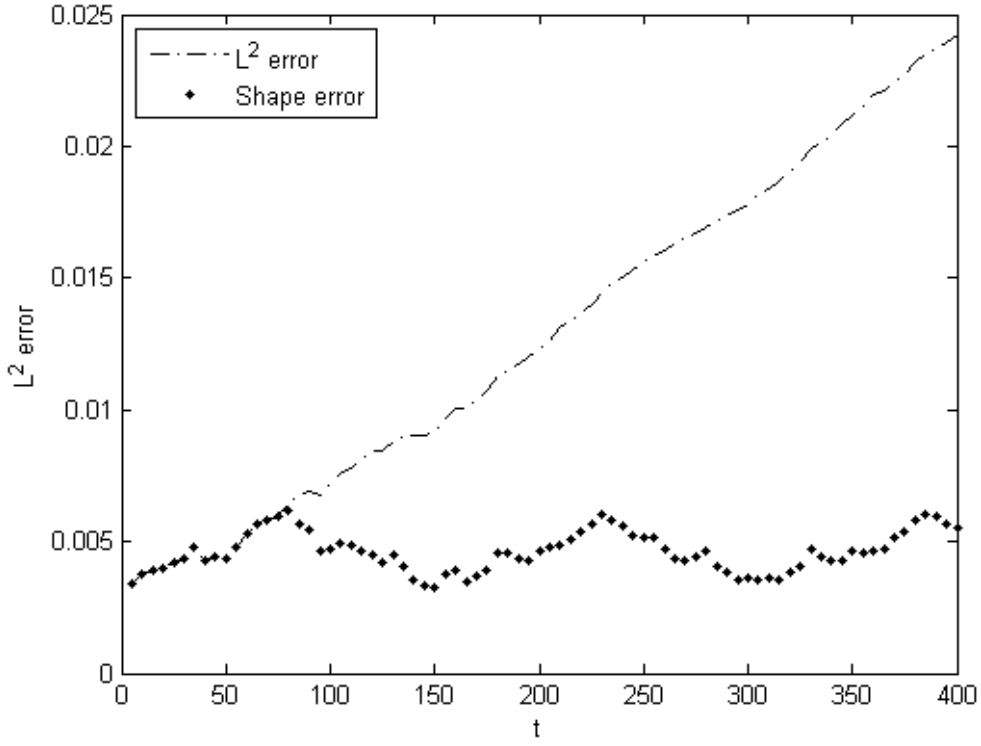

Figure 6.3 Example 2: the evolution of the L^2 error and the shape error obtained from the LDG-C scheme.

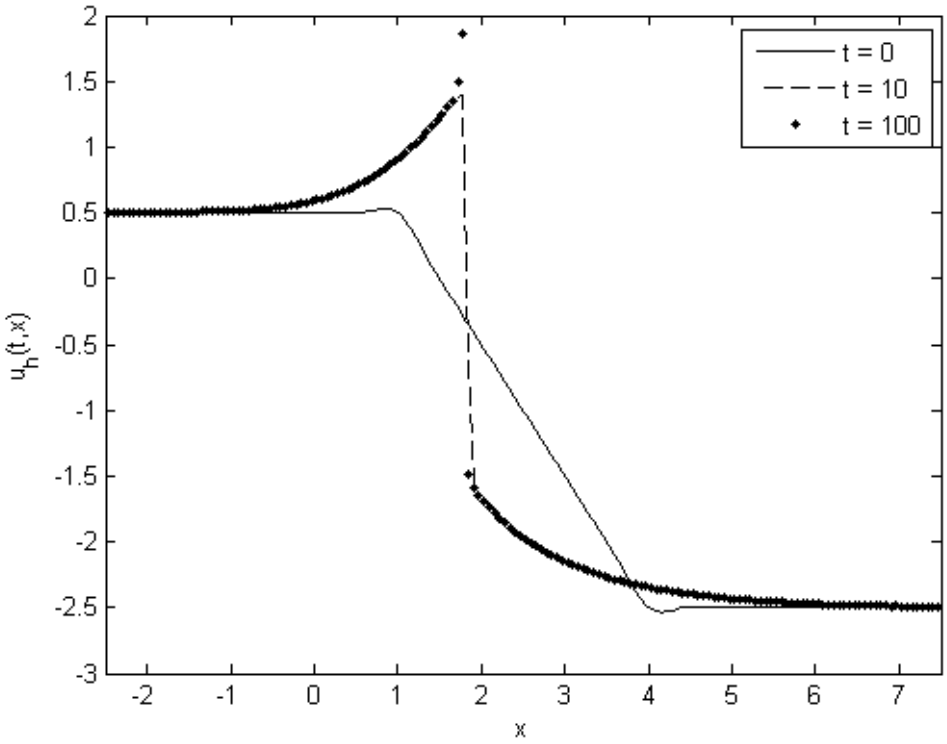

Figure 6.4 Example 3: the computed solution at $t = 0, 10, 100$.

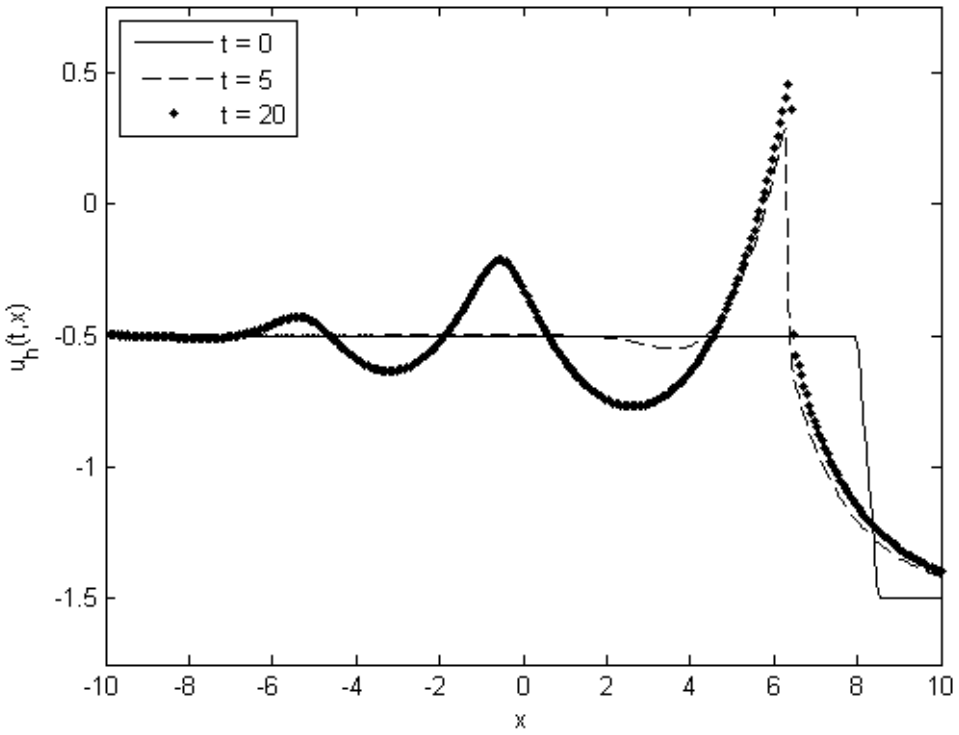

Figure 6.5 Example 4: the computed solution at $t = 0, 5, 20$.

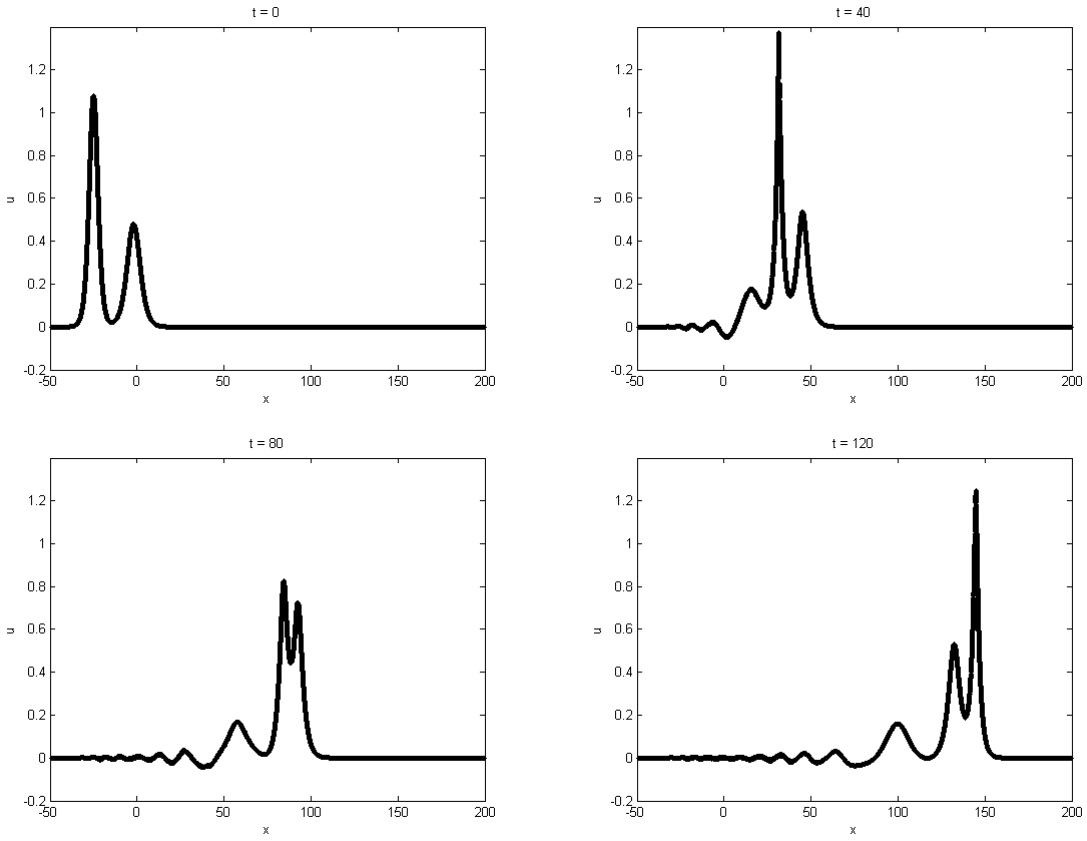

Figure 6.6 Example 5: the evolution of two traveling waves at $t = 0, 40, 80, 120$.

CHAPTER 7. SUMMARY AND DISCUSSION

7.1 General conclusion

In this thesis, we achieved two main objectives, namely the recovery of high frequency wave fields, and the approximation of dispersive waves. We used the Gaussian beam method to recover the highly oscillatory wave fields, and we used the local discontinuous Galerkin method to approximate the dispersive waves while preserving some invariants at the discrete level.

1. We numerically implemented the recovery of high frequency wave for Schrödinger equation using the superposition of Gaussian beams. In doing so, we proposed a new search algorithm that captures a moving manifold driven by a Hamiltonian flow. The algorithm description is for an interface in two dimension, but it can be generalized to a higher dimension. We also modified the existing delta approximation to fit into our framework. The Lagrangian method is adapt in order to compute the components of GB and the level set that represents the moving interface. We presented numerical examples of the solutions for the Schrödinger equation in two and four dimensional phased spaces to verify that our implementation achieved the expected order of accuracy.

2. We proposed the LDG method for Burgers-Poisson equation. The scheme is proven to preserve the solution's momentum and energy, and is of optimal order when polynomial basis of even order is used. We also introduced a global projector and derived its properties, which are useful for the error estimation. The projection technique we presented here can be use as a standard approach for an error estimation in DG.

Various numerical examples were tested in order to justify the performance of the proposed LDG scheme: that is, the error of accuracy is optimal for even-order polynomial basis, and the energy is preserved after a long-time simulation. Examples with discontinuities and with

interaction of peaks were also tested, where we incorporated the adaptive flux.

7.2 Future work

1. The search algorithm we developed can be adapted into other high frequency wave fields besides the Schrödinger equation. It can also be used for a more general interface tracking problem. The requirement is that the moving interface does not break or merge. We plan to explore further more complex wave propagation problems, and other interface problems which have similar feature.

2. The idea for energy-preserving DG method can be applied to other shallow-water wave equations with soliton solutions or to the PDE's that preserve invariants of similar form such as momentum and energy. We plan to work on some nonlocal dispersive PDEs, which yield both smooth wave propagation and short wave breaking.

BIBLIOGRAPHY

Amadori, D. and Gosse, L. (2013). Transient L^1 error estimates for well-balanced schemes on non-resonant scalar balance laws. *J. Differential Equations*, 255(3):469–502.

Bao, W., Jin, S., and Markowich, P. A. (2002). On time-splitting spectral approximations for the Schrödinger equation in the semiclassical regime. *J. Comput. Phys.*, 175(2):487–524.

Bona, J. L., Chen, H., Karakashian, O., and Xing, Y. (2013). Conservative, discontinuous Galerkin-methods for the generalized Korteweg-de Vries equation. *Math. Comp.*, 82(283):1401–1432.

Bona, J. L., Dougalis, V. A., Karakashian, O. A., and McKinney, W. R. (1995). Conservative, high-order numerical schemes for the generalized Korteweg-de Vries equation. *Philos. Trans. Roy. Soc. London Ser. A*, 351(1695):107–164.

Camassa, R. and Holm, D. D. (1993). An integrable shallow water equation with peaked solitons. *Phys. Rev. Lett.*, 71(11):1661–1664.

Camassa, R., Holm, D. D., and Hyman, J. M. (1994). A new integrable shallow water equation. *Advances in Applied Mechanics*, 31(31):1–33.

Castillo, P., Cockburn, B., Schötzau, D., and Schwab, C. (2002). Optimal a priori error estimates for the *hp*-version of the local discontinuous Galerkin method for convection-diffusion problems. *Math. Comp.*, 71(238):455–478.

Celledoni, E., Grimm, V., McLachlan, R. I., McLaren, D. I., O'Neale, D., Owren, B., and Quispel, G. R. W. (2012). Preserving energy resp. dissipation in numerical PDEs using the "average vector field" method. *J. Comput. Phys.*, 231(20):6770–6789.

Ciarlet, P. G. (1978). *The finite element method for elliptic problems*. North-Holland, New York.

Cockburn, B., Hou, S., and Shu, C.-W. (1990). The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Math. Comp.*, 54(190):545–581.

Cockburn, B., Lin, S. Y., and Shu, C.-W. (1989). TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems. *J. Comput. Phys.*, 84(1):90–113.

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(186):411–435.

Cockburn, B. and Shu, C.-W. (1998a). The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463 (electronic).

Cockburn, B. and Shu, C.-W. (1998b). The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems. *J. Comput. Phys.*, 141(2):199–224.

Constantin, A. and Lannes, D. (2009). The hydrodynamical relevance of the Camassa-Holm and Degasperis-Procesi equations. *Arch. Ration. Mech. Anal.*, 192(1):165–186.

Degasperis, A., Kholm, D. D., and Khon, A. N. I. (2002). A new integrable equation with peakon solutions. *Teoret. Mat. Fiz.*, 133(2):170–183.

Engquist, B., Tornberg, A.-K., and Tsai, R. (2005). Discretization of Dirac delta functions in level set methods. *J. Comput. Phys.*, 207(1):28–51.

Fellner, K. and Schmeiser, C. (2004). Burgers-Poisson: a nonlinear dispersive model equation. *SIAM J. Appl. Math.*, 64(5):1509–1525 (electronic).

Fuchssteiner, B. and Fokas, A. S. (1982). Symplectic structures, their Bäcklund transformations and hereditary symmetries. *Phys. D*, 4(1):47–66.

Furihata, D. and Mori, M. (1998). General derivation of finite difference schemes by means of a discrete variation. *TRANSACTIONS-JAPAN SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS*, 8:317–340.

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

Hirota, R. (1971). A direct discontinuous galerkin method for the generalized korteweg-de vries equation: Energy conservation and boundary effect. *Phys. Lett. Rev.*, 27(18):1192–1194.

Holm, D. D. and Staley, M. F. (2003). Wave structure and nonlinear balances in a family of evolutionary PDEs. *SIAM J. Appl. Dyn. Syst.*, 2(3):323–380 (electronic).

Jin, S., Wu, H., and Yang, X. (2008). Gaussian beam methods for the Schrödinger equation in the semi-classical regime: Lagrangian and Eulerian formulations. *Commun. Math. Sci.*, 6(4):995–1020.

Johnson, R. S. (2002). Camassa-Holm, Korteweg-de Vries and related models for water waves. *J. Fluid Mech.*, 455:63–82.

Leung, S. and Qian, J. (2009). Eulerian gaussian beams for schrödinger equations in the semi-classical regime. *Journal of Computational Physics*, 228(8):2951–2977.

Liu, H. (2014). Optimal error estimates of the direct dicontinuous galerkin method for convection-diffusion equations. *Math. Comp.* accepted.

Liu, H., Huang, Y., and Yi, N. (2014). A direct discontinuous galerkin method for the Degasperis–Procesi equation. *Methods Appl. Anal.*, 21(1):83–106.

Liu, H. and Ralston, J. (2010). Recovery of high frequency wave fields from phase space-based measurements. *Multiscale Model. Simul.*, 8(2):622–644.

Liu, H. and Yan, J. (2006). A local discontinuous galerkin method for the korteweg–de vries equation with boundary effect. *J. Comput. Phys.*, 215(1):197–218.

Liu, H. and Yan, J. (2009). The direct discontinuous Galerkin (DDG) methods for diffusion problems. *SIAM J. Numer. Anal.*, 47(1):675–698.

Liu, H. and Yan, J. (2010). The direct discontinuous Galerkin (DDG) method for diffusion with interface corrections. *Commun. Comput. Phys.*, 8(3):541–564.

Liu, H. and Yin, Z. (2010). Global regularity, and wave breaking phenomena in a class of nonlocal dispersive equations. In *Nonlinear partial differential equations and hyperbolic wave phenomena*, volume 526 of *Contemp. Math.*, pages 273–294. Amer. Math. Soc., Providence, RI.

Liu, Y. and Yin, Z. (2006). Global existence and blow-up phenomena for the Degasperis-Procesi equation. *Comm. Math. Phys.*, 267(3):801–820.

Markowich, P. A., Pietra, P., and Pohl, C. (1999). Numerical approximation of quadratic observables of Schrödinger-type equations in the semi-classical limit. *Numer. Math.*, 81(4):595–630.

Markowich, P. A., Pietra, P., Pohl, C., and Stimming, H. P. (2002). A Wigner-measure analysis of the Dufort-Frankel scheme for the Schrödinger equation. *SIAM J. Numer. Anal.*, 40(4):1281–1310.

Matsuo, T. (2008). Dissipative/conservative Galerkin method using discrete partial derivatives for nonlinear evolution equations. *J. Comput. Appl. Math.*, 218(2):506–521.

McLachlan, R. I., Quispel, G. R. W., and Robidoux, N. (1999). Geometric integration using discrete gradients. *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, 357(1754):1021–1045.

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49.

Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *J. Computational Phys.*, 25(3):220–252.

Qian, J. and Ying, L. (2010). Fast Gaussian wavepacket transforms and Gaussian beams for the Schrödinger equation. *J. Comput. Phys.*, 229(20):7848–7873.

Ralston, J. (1982). Gaussian beams and the propagation of singularities. In *Studies in partial differential equations*, volume 23 of *MAA Stud. Math.*, pages 206–248. Math. Assoc. America, Washington, DC.

Reed, W. H. and Hill, T. R. (1973). Triangular mesh methods for the neutrontransport equation. *Los Alamos Report LA-UR-73-479*.

Smereka, P. (2006). The numerical approximation of a delta function with application to level set methods. *J. Comput. Phys.*, 211(1):77–90.

Warburton, T. and Hesthaven, J. S. (2003). On the constants in *hp*-finite element trace inverse inequalities. *Comput. Methods Appl. Mech. Engrg.*, 192(25):2765–2773.

Wen, X. (2008). High order numerical quadratures to one dimensional delta function integrals. *SIAM J. Sci. Comput.*, 30(4):1825–1846.

Whitham, G. B. (1974). *Linear and nonlinear waves*. John Wiley & Sons, New York.

Xu, Y. and Shu, C. (2005). Local discontinuous galerkin methods for two classes of two-dimensional nonlinear wave equations. *Physica D: Nonlinear phenomena*, 208(1-2):21–58.

Xu, Y. and Shu, C. (2007). Error estimates of the semi-discrete local discontinuous galerkin method for nonlinear convection-diffusion and kdv equations. *Computer methods in applied mechanics and engineering*, 196(37-40):3805–3822.

Xu, Y. and Shu, C.-W. (2008). A local discontinuous Galerkin method for the Camassa-Holm equation. *SIAM J. Numer. Anal.*, 46(4):1998–2021.

Xu, Y. and Shu, C.-W. (2010). Local discontinuous Galerkin methods for high-order time-dependent partial differential equations. *Commun. Comput. Phys.*, 7(1):1–46.

Yan, J. and Shu, C. (2003). A local discontinuous galerkin method for kdv type equations. *SIAM Journal on Numerical Analysis*, 40:769–791.

Yi, N., Huang, Y., and Liu, H. (2013). A direct discontinuous galerkin method for the generalized korteweg-de vries equation: Energy conservation and boundary effect. *Journal of Computational Physics*.