NTNU – Trondheim
Norwegian University of
Science and Technology

Numerical Methods for the Benjamin-Ono Equation

Geir Midgard Fiksdal

Master of Science in Physics and Mathematics
Submission date: Januar 2013
Supervisor: Helge Holden, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

In this thesis, we compare four numerical methods for solving the Benjamin-Ono equation. The numerical methods are presented in detail, and we compare them for different test problems. We derive the Hirota bilinear form of the Benjamin-Ono equation, and present spatially periodic exact solutions. The best numerical method is Chan and Kerkhoven's Semi-Implicit Fourier pseudospectral method, originally intended for the Korteweg-de Vries equation. In the last chapter, we study the zero dispersion limit for the Korteweg-de Vries and Benjamin-Ono equation. We observe that the small dispersion term forces the shock formation in the solution to become travelling waves.

Sammendrag

I denne oppgaven sammenligner vi fire numeriske metoder som løser Benjamin-Ono-ligningen. De numeriske metodene er detaljert lagt fram, og vi sammenligner dem på forskjellige testproblemer. Vi utleder Hirotas bilineære form av ligningen, og presenterer eksakte løsninger som er periodiske i rom. Den beste numeriske metoden er Chan og Kerkhovens semi-implisitte Fourier pseudospektral-metode, som originalt var ment for Korteweg-de Vries-ligningen. I siste kapittel studerer vi grensen hvor dispersjonsleddet går mot null i Korteweg-de Vries- og Benjamin-Ono-ligningen. Vi finner at et lite dispersjonsledd forårsaker bølger i løsningen når det oppstår sjokk.

Contents

Preface

This thesis is the final part of my master's degree in Industrial Mathematics at the Norwegian University of Science and Technology (NTNU). The work started on the 3rd of September 2012, and was completed on the 27th of January 2013. Some of the theoretical background, and some of the code, is a continuation of my specialisation project on the Korteweg-de Vries equation in the spring semester of 2012. I would like to thank my supervisor Helge Holden, professor at the Department of Mathematics at NTNU, for his support and ideas during my work, both on the specialisation project and on my master's thesis.

Geir Midgard Fiksdal
Trondheim
January 27, 2013

Chapter 1

Introduction

This thesis deals with numerical methods for solving the Benjamin-Ono equation. The equation was derived first by Benjamin [Ben67], and a few years later by Ono [Ono75]. It describes internal waves between two stratified homogenous fluids with different densities, where one of the layers is infinitely deep. Later the equation has also been found to model the meteorological phenomenon known as the "Morning Glory cloud" in northeastern Australia [PS02].

The Benjamin-Ono equation is a nonlinear evolution equation. Among these equations, we also have the Korteweg-de Vries equation. Both of these have been extensively studied, and they share similar properties. They both have known closed form solutions, but the increasing complexity of these solutions when we want to model more than the simplest cases, is a good reason to develop accurate and efficient numerical methods. We will derive the Hirota bilinear form of the Benjamin-Ono equation, and state the exact solutions as found by [SI79].

The first aim of this thesis, is to study and compare the accuracy and performance of four numerical methods. The first method was proposed by Thomée and Vasudeva Murthy [TV98], and solves the equation in a finite difference like manner. The second and third method are Fourier pseudospectral methods, and were originally intended for the Korteweg-de Vries equation. Here we approximate the spatial derivatives efficiently using the fast Fourier transform. Because of the similarities between these two equations, can we easily adapt these methods to solve the Benjamin-Ono equation. The third and fourth method use operator splitting, in which we split the equation into two simpler equations, and use the solution to the first as initial condition for the second. These methods differ in the way they solve the first of the two equations.

The second aim of this thesis, is to investigate numerically the zero dispersion limit of the Burgers, Korteweg-de Vries and Benjamin-Ono equation. We introduce a constant in front of the dispersive term in each equation, and look at the solutions as we send the constant towards zero. When the constant is zero, we have the inviscid Burgers equation. We will use a known weak solution to this equation as initial condition for the Korteweg-de Vries and Benjamin-Ono equation in the zero dispersion limit.

Chapter 2

Theoretical Background

This chapter will provide the necessary theoretical background needed to derive the numerical methods for solving the Benjamin-Ono equation. The Benjamin-Ono equation is an integro-differential equation defined as

$$u_t + uu_x - Hu_{xx} = 0, \quad \text{for} \quad x \in \mathbb{R}, \quad t > 0, \tag{2.1}$$

where the H denotes the Hilbert transform defined below. The Hilbert transform is often defined differently from what we do here, so the equation can also be found with a plus in front of the last term. We first define the Hilbert transform, and then our problem setup.

Definition 1. *The **Hilbert transform** of a function $f \in L^p$, $1 \leq p < \infty$, is defined as the Cauchy principal value*

$$Hf(x) = \mathrm{PV}\frac{1}{\pi}\int_{\mathbb{R}} \frac{f(x-y)}{y}\,\mathrm{d}y \equiv \lim_{\delta \to 0^+} (Hf)_\delta(x) \tag{2.2}$$

at every point x where the limit exists, where

$$(Hf)_\delta(x) = \frac{1}{\pi}\int\limits_{|y|\geq\delta} \frac{f(x-y)}{y}\,dy.$$

The definition is taken from the book by Butzer and Nessel [BN71], and we refer to them for a thorough analysis on existence and properties. We now go on to define our problem setup.

2.1 Problem Setup

We want to approximate the equation (2.1) by a spatially periodic initial value problem. The known solutions to the infinite problem decay to zero for large $|x|$, but the decay is only polynomial. Thus, we need to choose a large spatial period to permit good numerical approximations to the infinite problem. However, as test problems, will we mainly use analytic solutions that are spatially periodic.

For a finite-dimensional approximation to the equation in (2.1), we look for $2L$-periodic solutions of the periodic problem. We can write the problem setup as

$$\begin{aligned} &u_t + uu_x - \tilde{H}u_{xx} = 0, && (x,t) \in \mathbb{R} \times (0,\infty), \\ &u(x,0) = u_0(x), && x \in \mathbb{R}, \\ &u(x+2L,t) = u(x,t), && (x,t) \in \mathbb{R} \times [0,\infty). \end{aligned} \tag{2.3}$$

The $\tilde{H}$ denotes the periodic version of the Hilbert transform, which we will derive next.

2.2 The Periodic Hilbert Transform

The periodic Hilbert transform can be found from (2.2) in Definition 1. We follow Thomée and Vasudeva Murthy [TV98] here, and consider the $2L$-periodic function f. First we avoid the singularity at $y = 0$ and write the integral in (2.2) for a finite number of periodic intervals, $((2k-1)L, (2k+1)L)$ for $k = 0, 1, \dots, N$

$$\int\limits_{\epsilon \le |y| \le (2N+1)L} \frac{f(x-y)}{y}\,\mathrm{d}y = \int\limits_{\epsilon \le |y| \le L} \frac{f(x-y)}{y}\,\mathrm{d}y \quad + \sum_{\substack{k=-N \\ k \ne 0}}^{N} \int\limits_{(2k-1)L}^{(2k+1)L} \frac{f(x-y)}{y}\,\mathrm{d}y.$$

Now, by making use of the periodicity of f, and transforming the intervals $((2k-1)L, (2k+1)L)$ to $(-L, L)$, we can write the last term as

$$\sum_{\substack{k=-N \\ k \ne 0}}^{N} \int\limits_{-L}^{L} \frac{f(x-y-2kL)}{y+2kL}\,\mathrm{d}y = \sum_{\substack{k=-N \\ k \ne 0}}^{N} \int\limits_{-L}^{L} \left(\frac{1}{y+2kL} + \frac{1}{2kL} \right) f(x-y)\,\mathrm{d}y,$$

where the second term in the parenthesis can be added since they sum to zero when the negative and positive terms cancel. In total we have

$$\int\limits_{\epsilon \le |y| \le L} \frac{f(x-y)}{y}\,\mathrm{d}y \quad + \sum_{\substack{k=-N \\ k \ne 0}}^{N} \int\limits_{-L}^{L} \left(\frac{1}{y+2kL} + \frac{1}{2kL} \right) f(x-y)\,\mathrm{d}y, \tag{2.4}$$

and we can make use of the following series representation of $\cot(y)$, from Chapter 9 in [Hil73],

$$\frac{\pi}{2L}\cot\left(\frac{\pi}{2L}y\right) = \frac{1}{y} + \sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}\left(\frac{1}{y+2kL} + \frac{1}{2kL}\right), \quad \text{for} \quad |y| \leq L.$$

If we let $\epsilon \to 0$ and $N \to \infty$ in (2.4), we get the periodic Hilbert transform. Let L^p_{2L} denote the space of $2L$-periodic functions in L^p, $1 \leq p < \infty$.

Definition 2. *The* ***periodic Hilbert transform*** *of a function $f \in L^p_{2L}$, $1 \leq p < \infty$, is defined as the Cauchy principal value*

$$\tilde{H}f(x) = \mathrm{PV}\frac{1}{2L}\int_{-L}^{L} f(x-y)\cot\left(\frac{\pi}{2L}y\right)\mathrm{d}y = \lim_{\delta\to 0^+}(\tilde{H}f)_\delta(x) \tag{2.5}$$

at every point x where the limit exists, where

$$(\tilde{H}f)_\delta(x) = \frac{1}{2L}\int_{\delta\leq|y|\leq L} f(x-y)\cot\left(\frac{\pi}{2L}y\right)dy.$$

2.3 A Discrete Periodic Hilbert Transform

We will need a discrete periodic variant of the Hilbert transform when we develop the first numerical method in Chapter 4. The discrete transform is proposed by Thomée and Vasudeva Murthy [TV98]. Assume that the periodicity interval $(-L,L)$ is partitioned into $2N$ equidistant intervals of length $h = L/N$, and assume N to be even, $N = 2M$. Then we apply the midpoint rule on the $N = 2M$ intervals (x_{2k}, x_{2k+2}), $k = -M, \ldots, M-1$, of length $2h$, to the integral in (2.5) and obtain

$$(\tilde{H}_hF)_j = \frac{1}{2L}\sum_{k=-M}^{M-1} 2h\cot\left(\frac{\pi}{2L}x_{2k+1}\right)F_{j-(2k+1)}, \tag{2.6}$$

where $F_j = f(x_j)$ for $j = -N, \ldots, N-1$. By doing this we also avoid the singularity at $x = 0$. This is a discrete convolution that we can write like

$$(\tilde{H}_hF)_j = \sum_{k=-N}^{N-1} c_kF_{j-k}, \quad \text{where} \quad c_k = \begin{cases}\frac{h}{L}\cot\left(\frac{\pi kh}{2L}\right), & \text{if } k \text{ is odd},\\ 0, & \text{if } k \text{ is even}.\end{cases} \tag{2.7}$$

The reason for choosing this discretisation will be apparent when we take the discrete Fourier transform of the discrete periodic Hilbert transform.

2.4 The Discrete Fourier Transform

We now define the discrete Fourier transform (DFT), and the inverse discrete Fourier transform. When we calculate the DFT in practice, we use the fast Fourier transform (FFT). Using the definition to compute the DFT is slow, requiring $\mathrm{O}(N^2)$ arithmetic operations. With the FFT, we can reduce the number of arithmetic operations to $\mathrm{O}(N\log N)$ while obtaining the same result.

Definition 3. *For $U = \{U_j\}$, $j = -N,\dots,N-1$, we define the **discrete Fourier transform** to be*

$$\hat{U}_p(t) = \mathscr{F}U(\cdot,t)(p) = \sum_{j=-N}^{N-1} U_j e^{-2\pi i jp/2N}, \quad \text{for} \quad p = -N,\dots,N-1. \tag{2.8}$$

Definition 4. *For $\hat{U} = \{\hat{U}_p\}$, $p = -N,\dots,N-1$, we define the **inverse discrete Fourier transform** to be*

$$U_j(t) = \mathscr{F}^{-1}\hat{U}(\cdot,t)(x) = \frac{1}{2N}\sum_{p=-N}^{N-1} \hat{U}_p e^{2\pi i jp/2N}, \quad \text{for} \quad j = -N,\dots,N-1. \tag{2.9}$$

In our Fourier pseudospectral methods we will approximate derivatives in space using the DFT. We can do that by multiplying each Fourier coefficient by its corresponding wave number, and then taking the inverse transform. The approximation to the qth derivative of a $2L$-periodic function u at (x_j,t) is

$$u^{(q)}(x_j,t) \approx \mathscr{F}^{-1}\hat{Z}(\cdot,t)(x_j), \quad \text{for} \quad \hat{Z}(p,t) = (isp)^q\hat{U}(p,t), \tag{2.10}$$

where $s = \pi/L$.

Informally, we can justify this by noting that for periodic functions, the Fourier transform can be reduced to calculating the Fourier series coefficients, or the "finite Fourier transform". The "finite Fourier transform", defined on $p \in \mathbb{Z}$ for the $2L$-periodic function f, is defined by

$$\hat{f}(p) = \frac{1}{2L}\int_{-L}^{L} f(x)e^{-ip\frac{\pi}{L}x}\mathrm{d}x.$$

So, for the finite Fourier transform of the derivative of $f(x)$ we have, with $s = \pi/L$,

$$2L\cdot\widehat{f_x}(p) = f(x)e^{-ispx}\Big|_{-L}^{L} + isp\int_{-L}^{L} f(x)e^{-ispx}\mathrm{d}x = 2L\cdot isp\,\hat{f}(p),$$

by integration by parts. Successive repetitions give us higher order derivatives. In practice we take the discrete Fourier transform, and we get the approximation in (2.10). For a rigorous analysis on this, we refer to Chapter 4 in the book by Butzer and Nessel [BN71].

We make a remark on notation. The DFT and the inverse DFT takes a vector as argument. But, in the following, whenever we write

$$\mathscr{F}^{-1}\left((isp)^q\hat{U}(p,t)\right)(X_j), \tag{2.11}$$

or similar expressions, we mean the single element at X_j of the inverse DFT of the vector $\hat{Z}(\cdot,t)$ whose elements are $(isp)^q\hat{U}(p,t)$.

2.5 The Fourier Transform of $Hf(x)$, $\tilde{H}f(x)$ and $\tilde{H}_hF$

For the Hilbert transform on the real line, we have the following result for the Fourier transform, found in Chapter 8 in [BN71]. The hats denote Fourier transform.

Proposition 1. *Let $f \in L^p$, $1 < p \le 2$. Then the Fourier transform of the Hilbert transform $Hf(x)$ is given by*

$$\widehat{Hf}(\xi) = -i\text{sign}(\xi)\hat{f}(\xi), \quad \text{a.e.} \tag{2.12}$$

For $p = 1$ this remains valid and then holds everywhere under the additional assumption $Hf(x) \in L^1$.

Now, for the periodic Hilbert transform we have a similar result. The proposition is taken from Chapter 9 in [BN71].

Proposition 2. *Let $f \in L^p_{2\pi}$, $1 < p < \infty$. Then*

$$\widehat{\tilde{H}f}(\xi) = -i\text{sign}(\xi)\hat{f}(\xi). \tag{2.13}$$

For $p = 1$ this holds under the additional assumption $\tilde{H}f(x) \in L^1_{2\pi}$.

We will now show that the discrete periodic Hilbert transform share the same property as the continuous version above, using the discrete Fourier transform.

Lemma 1. *For $\tilde{H}_h$ defined in* (2.6)*, and the DFT in* (2.8)*, we have*

$$\widehat{\tilde{H}_hF}_p = -i\,\widetilde{sign}(p)\hat{F}_p, \quad \textit{where} \quad \widetilde{sign}(p) = \begin{cases} 1, & \textit{if } 1 \le p \le N-1, \\ -1, & \textit{if } -N+1 \le p \le -1, \\ 0, & \textit{if } p = -N, 0. \end{cases} \tag{2.14}$$

Proof. Taking the DFT of the discrete convolution in equation (2.7) we get

$$\widehat{\tilde{H}_h F}_p = \sum_{j=-N}^{N-1} \sum_{k=-N}^{N-1} c_k F_{j-k} e^{-2\pi i j p/2N} = \hat{c}_p \hat{F}_p.$$

Thus, we need to show that $\hat{c}_p = -i\widetilde{\text{sign}}(p)$, which is equivalent to showing that if $\hat{W}_p = -i\widetilde{sign}(p)$, then $W_j = c_j$. So, taking the inverse DFT of $\hat{W}_p$

$$W_j = \frac{1}{2N} \sum_{p=-N}^{N-1} \hat{W}_p e^{2\pi i j p/2N} = \frac{1}{2N} \sum_{p=-N}^{N-1} -i\widetilde{\text{sign}}(p) e^{i\pi j p/2N},$$

which for $\widetilde{\text{sign}}(p)$ in (2.14) can be written as

$$W_j = \frac{-i}{2N} \sum_{p=0}^{N-1} e^{i\pi j p/N} - e^{i\pi j(-p)/N} = \frac{1}{2Ni} \sum_{p=0}^{N-1} \left(e^{i\pi j/N}\right)^p - \left(e^{-i\pi j/N}\right)^p.$$

This is the sum of a geometric series which we can write as

$$W_j = \frac{1}{2Ni} \left(\frac{1 - e^{i\pi j}}{1 - e^{i\pi j/N}} - \frac{1 - e^{-i\pi j}}{1 - e^{-i\pi j/N}} \right).$$

For j even, $e^{i\pi j} = e^{-i\pi j} = 1$, thus $W_j = 0$. For j odd, $e^{i\pi j} = e^{-i\pi j} = -1$, so we have

$$W_j = \frac{1}{2Ni} \left(\frac{2}{1 - e^{i\pi j/N}} - \frac{2}{1 - e^{-i\pi j/N}} \right) = \frac{1}{N} \frac{\sin(\pi j/N)}{1 - \cos(\pi j/N)} = \frac{1}{N} \cot\left(\frac{\pi j}{2N}\right).$$

Since $h = L/N$, we have that c_j is the same as in (2.7), which proves the lemma. □

This property is the reason for choosing the midpoint rule in discretising the Hilbert transform. The midpoint rule is second order accurate, even with the singularity at the origin. Thomée and Vasudeva Murthy prove that the discrete periodic Hilbert transform is a second order approximation of the continuous one for smooth functions.

2.6 A Short Introduction to Operator Splitting

In two of the numerical methods presented in Chapter 4, we will use the idea of operator splitting to find numerical solutions to the problem setup in equation (2.3). We will not prove any results, but it is appropriate to give a short background on the concept of operator splitting. The following explanation is adapted from the book on splitting methods [HKLR10]. We want to find a solution to the Cauchy problem on the form

$$\frac{\partial u}{\partial t} + \mathscr{A}(u) = 0, \quad u(0) = u_0,$$

where $\mathscr{A}(u) = uu_x - Hu_{xx}$, in the case of the BO equation. We can formally denote the solution by

$$u(t) = e^{-t\mathscr{A}} u_0,$$

where $e^{-t\mathscr{A}}$ is the solution operator. The idea is then to write the operator as $\mathscr{A} = \mathscr{A}_1 + \mathscr{A}_2$, and define the easier sub-problems

$$\frac{\partial u}{\partial t} + \mathscr{A}_i u = 0, \quad u(0) = u_0, \quad i = 1, 2,$$

with formal solutions

$$u_i(t) = e^{-t\mathscr{A}_i} u_0, \quad i = 1, 2.$$

For the BO equation we can choose $\mathscr{A}_1$ and $\mathscr{A}_2$ such that we get the equations

$$u_t + uu_x = 0, \quad u_t - Hu_{xx} = 0.$$

Now, the idea is to let the solution to the first equation serve as initial condition for the second equation. So if we let $t_n = n\Delta t$, for a small and positive Δt, we can write

$$u(t_{n+1}) \approx e^{-\Delta t \mathscr{A}_2} e^{-\Delta t \mathscr{A}_1} u_0.$$

For commuting operators we have $e^{-\Delta t \mathscr{A}_2} e^{-\Delta t \mathscr{A}_1} = e^{-\Delta t \mathscr{A}}$. The next step would be to take the limit as $\Delta t \downarrow 0$, and hopefully we get

$$u(t) = e^{-\Delta t \mathscr{A}} = \lim_{\Delta t \downarrow 0, t = n\Delta t} \left(e^{-\Delta t \mathscr{A}_2} e^{-\Delta t \mathscr{A}_1} \right)^n u_0.$$

To get a numerical method, we swap the exact solution operators with numerical solutions.

2.7 A Short Introduction to Fourier Pseudospectral Methods

Since two of our methods are Fourier pseudospectral methods, we want to give a short introduction. Assume the following differential equation

$$Lu = f,$$

where L is a differential operator, and f some function. Spectral methods in general, approximate the solution as a finite sum of basis functions,

$$u(x) \approx u_h(x) = \sum_{n=0}^{N} a_n \phi_n(x).$$

Contrary to e.g. finite element methods, are the basis functions defined as global functions, generally non-zero on the entire domain. This series is substituted into the equation, and we form a residual function

$$R(x;a_0,\dots,a_N) = Lu_h - f.$$

We want to determine the coefficients a_n such that we minimise the residual function in some way. The three main ways of doing this are the Galerkin, tau, and the collocation method. The collocation method is often called the pseudospectral method, and consists in making the residual zero at a set of nodal points. In Fourier pseudospectral methods, we use Fourier series to approximate the solution. Since the Fourier basis functions are periodic, we have to require that the solution satisfies periodic boundary conditions.

For the Benjamin-Ono equation we let $w = u^2/2$, s.t. $w_x = uu_x$, and discretise

$$u_t(x_j,t) = -w_x(x_j,t) + \tilde{H}(u_{xx})(x_j,t), \quad \text{for} \quad j = -N,\dots,N-1.$$

We approximate the solution by the finite Fourier series

$$u(x,t) \approx \frac{1}{2N}\sum_{p=-N}^{N-1} \hat{u}_p(t)e^{ip\frac{\pi}{L}(x+L)} = \frac{1}{2N}\sum_{p=-N}^{N-1} \hat{u}_p(t)e^{ips(x+L)},$$

where $\hat{u}_p$ is given by the DFT previously defined. Then we can easily approximate the derivatives

$$w_x(x,t) \approx \frac{1}{2N}\sum_{p=-N}^{N-1} \hat{w}_p(t)ispe^{ips(x+L)},$$

$$u_{xx}(x,t) \approx -\frac{1}{2N}\sum_{p=-N}^{N-1} \hat{u}_p(t)(sp)^2e^{ips(x+L)}.$$

The Hilbert transform commutes with differentiation, and since the Fourier basis functions are eigenfunctions of the Hilbert transform,

$$\tilde{H}(e^{ips(x+L)}) = -i\,\mathrm{sign}(p)e^{ips(x+L)},$$

we have that

$$\tilde{H}(u_{xx})(x,t) \approx \frac{1}{2N}\sum_{p=-N}^{N-1} \hat{u}_p(t)i\,\mathrm{sign}(p)(sp)^2e^{ips(x+L)}.$$

Then we substitute these expressions into the discretised Benjamin-Ono equation and get

$$\frac{\partial}{\partial t}u(x_j,t) = \frac{1}{2N}\sum_{p=-N}^{N-1}\left(-\hat{w}_p(t)ispe^{ips(x_j+L)} + (ps)^2 i\,\mathrm{sign}(p)\hat{u}_p(t)e^{isp(x_j+L)}\right),$$

for $j = -N, \dots, N-1$. So we have that

$$\frac{\partial}{\partial t}\hat{u}_p(t) = -isp\hat{w}_p(t) + ip^2s^2\operatorname{sign}(p)\hat{u}_p(t).$$

In practice, what we do in the numerical methods, is to transform the equation into discrete Fourier space, where we can easily determine spatial derivatives, and then transform back, according to (2.10). We either do the time integration in discrete Fourier space, or after transforming back. We will also combine this with operator splitting. For more on Fourier pseudospectral methods we refer to [For98, Boy01].

2.8 The Korteweg-de Vries Equation

Two of the numerical methods we will test were originally intended for the Korteweg-de Vries (KdV) equation. The KdV equation is,

$$u_t + uu_x + u_{xxx} = 0. \tag{2.15}$$

The equation was introduced in 1895 by Korteweg and de Vries [KdV95], and models small one dimensional shallow water waves. This equation has been extensively studied, and has many known applications. In common with the Benjamin-Ono equation, we have the inviscid Burgers equation $u_t + uu_x = 0$. Because of the simple expression of the Fourier transform of the Hilbert transform, there is only a small difference in approximating the dispersive u_{xxx} term in the KdV equation, compared to the Hilbert term, Hu_{xx}, in the BO equation. Therefore we have adopted some methods designed for the KdV equation.

The periodic KdV equation has known solutions similar to the solutions to the periodic BO equation. For the KdV equation, these are called Cnoidal waves, and can be seen as periodic solitons. They are analogous to the periodic wave solutions of the BO equation, which we will discuss later.

2.9 The Burgers Equation

The Burgers equation is a third equation which shares two of the same terms as the BO and KdV equations. It is defined as

$$u_t + uu_x = \epsilon u_{xx}, \tag{2.16}$$

where ϵ is the viscosity coefficient. When ϵ is zero, we get the inviscid Burgers equation. This equation can develop shocks and rarefactions in finite time. We will use solutions to the inviscid Burgers equation to see what happens in the zero dispersion limit of the BO, KdV and Burgers equation in Chapter 6.

Chapter 3

Exact Solutions and Test Problems

In this section we present some exact solutions to both the infinite line BO equation, and the periodic BO equation in (2.3). We will use these known exact solutions to test the accuracy of our numerical methods. As a last test problem, we use an initial condition which is not a known explicit solution to the BO equation.

3.1 Periodic Wave Solutions

The BO equation has been found to have spatially N-periodic wave solutions that tend towards the N-soliton solution in the "long wave limit", i.e. when the wave number goes to zero [SI79]. These N-periodic wave solutions have been found using Hirota's method, transforming the equation into a bilinear form using Hirota derivatives. We follow the article by Satsuma and Ishimori [SI79] and derive the Hirota bilinear form of the BO equation. Then we present the 1- and 2-periodic wave solutions, and a peak into the general N-periodic wave solution.

3.1.1 Hirota Bilinear Form

First we introduce the variable transformation

$$u(x,t) = 2i\frac{\partial}{\partial x}\log[G(x,t)/F(x,t)]. \tag{3.1}$$

We now assume (writing equivalent statements for G in brackets) that F $[G]$ can be written as a finite or infinite product of $x - z_n$ $[x - z'_n]$, for z_n $[z'_n]$ in the upper [lower]-half complex plane. Then it can be shown that

$$H\left[2i\frac{\partial}{\partial x}\log(G/F)\right](x,t) = 2\frac{\partial}{\partial x}\log(FG). \tag{3.2}$$

This result will allow us to write the BO equation in Hirota bilinear form. First we define the following differential operator.

Definition 5. *The* ***D-operator*** *or* ***Hirota derivative****, acting on two sufficiently many times differentiable functions $F(x,t)$ and $G(x,t)$, is defined as*

$$D_x^n D_t^m F\cdot G = \left(\frac{\partial}{\partial x}-\frac{\partial}{\partial x'}\right)^n\left(\frac{\partial}{\partial t}-\frac{\partial}{\partial t'}\right)^m F(x,t)G(x',t')\Big|_{x'=x,t'=t}, \tag{3.3}$$

for m,n integer.

Inserting the expression in (3.1) into the BO equation, and using the property in (3.2), we get the following equation

$$\begin{aligned}\frac{\partial}{\partial t}\left(2i\frac{\partial}{\partial x}\log(G/F)\right)+2i\frac{\partial}{\partial x}\log(G/F)\,2i\frac{\partial^2}{\partial x^2}\log(G/F)-2\frac{\partial}{\partial x}\frac{\partial^2}{\partial x^2}\log(FG)=0,\\ \frac{\partial}{\partial x}\left(i\frac{\partial}{\partial t}\log(G/F)\right)-\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\log(G/F)\right)^2-\frac{\partial}{\partial x}\frac{\partial^2}{\partial x^2}\log(FG)=0.\end{aligned}$$

Integrating once with respect to x, and letting the constant of integration be zero, we get

$$i\frac{\partial}{\partial t}\log(G/F)-\left(\frac{\partial}{\partial x}\log(G/F)\right)^2-\frac{\partial^2}{\partial x^2}\log(GF)=0.$$

Straight forward differentiation yields

$$\begin{aligned}\frac{\partial}{\partial t}\log(G/F)&=\frac{G_t}{G}-\frac{F_t}{F},\\ \frac{\partial}{\partial x}\log(G/F)&=\frac{G_x}{G}-\frac{F_x}{F},\\ \frac{\partial^2}{\partial x^2}\log(FG)&=\frac{F_{xx}F-F_x^2}{F}+\frac{G_{xx}G-G_x^2}{G^2}.\end{aligned}$$

Inserting this into the previous equation, and after some manipulation, we end up with

$$\begin{aligned}i(G_tF-GF_t)-(G_{xx}F-2G_xF_x+GF_{xx})=0,\\ (iD_t-D_x^2)G\cdot F=0,\end{aligned} \tag{3.4}$$

which is the Hirota bilinear form of the Benjamin-Ono equation.

3.1.2 1-Periodic Wave Solution

The 1-periodic wave solution is the simplest solution to equation (3.4), and can be written as

$$F = 1 + e^{i\xi+\phi}, \tag{3.5}$$

$$G = 1 + e^{i\xi-\phi}, \tag{3.6}$$

where

$$\xi = k(x-ct) + \xi_0, \tag{3.7}$$

$$c = k\coth\phi, \tag{3.8}$$

and k, ϕ and ξ_0 are real constants. To satisfy the condition for the result in equation (3.2) we must require $\phi/k > 0$. Then, by inserting (3.5) and (3.6) into the bilinear BO equation (3.4) we get the explicit solution to the periodic BO

$$u(x,t) = \frac{2k\tanh\phi}{1+\operatorname{sech}\phi\cos\xi}. \tag{3.9}$$

To compare our numerical methods to the method in [TV98] we need to express this solution differently. So, note that $\tanh\phi = 1/\coth\phi = k/c$, and choose $\xi_0 = \pi$ and $k = \pi/L$. Then we have

$$u(x,t) = \frac{2\pi^2/cL^2}{1-\operatorname{sech}\phi\cos(\pi/L(x-ct))}.$$

We can write $\tanh\phi = \operatorname{sech}\phi\sinh\phi = \frac{\pi}{cL}$, so $\operatorname{sech}\phi = \frac{\pi}{cL}\sinh^{-1}\phi$. Using the algebraic expressions for sinh and sech, we can find an expression for ϕ,

$$e^{\phi} = \sqrt{\frac{1+\pi/cL}{1-\pi/cL}},$$

which inserted into the algebraic expression for sech yields

$$\operatorname{sech}\phi = \sqrt{1-\frac{\pi^2}{c^2L^2}}.$$

Then we have the 1-periodic wave solution in [TV98]

$$u(x,t) = \frac{2c\delta^2}{1-\sqrt{1-\delta^2}\cos(\pi/L(x-ct))}, \quad \text{where} \quad \delta = \frac{\pi}{cL}. \tag{3.10}$$

3.1.3 2-Periodic Wave Solution

The 2-periodic wave solution is a bit more cumbersome as we need to take care of the nonlinear interaction of two waves. We get the 2-periodic solution choosing

$$F = 1 + e^{i\xi_1+\phi_1} + e^{i\xi_2+\phi_2} + e^{i\xi_1+\xi_2+\phi_1+\phi_2+A_{12}}, \tag{3.11}$$

$$G = 1 + e^{i\xi_1-\phi_1} + e^{i\xi_2-\phi_2} + e^{i\xi_1+\xi_2-\phi_1-\phi_2+A_{12}} \tag{3.12}$$

where

$$\xi_j = k_j(x - c_j t) + \xi_j^0, \tag{3.13}$$

$$c_j = k_j \coth\phi_j, \tag{3.14}$$

and k_j, ϕ_j are real parameters satisfying $\phi_j/k_j > 0$, and ξ_j^0 is a complex phase constant. The equations (3.11) and (3.12) satisfy (3.4) given

$$e^{A_{12}} = \frac{(c_1-c_2)^2 - (k_1-k_2)^2}{(c_1-c_2)^2 - (k_1+k_2)^2}.$$

Now, the choice of (3.11) and (3.12) will generally give us a complex u, but we can obtain a real solution by choosing the imaginary part of the phase constants ξ_j^0 to be $A_{12}/2$. Then we have (with ξ_j now real)

$$F = 1 + e^{i\xi_1+\phi_1-A_{12}/2} + e^{i\xi_2+\phi_2-A_{12}/2} + e^{i\xi_1+i\xi_2+\phi_1+\phi_2}, \tag{3.15}$$

$$G = e^{i\xi_1+i\xi_2-\phi_1-\phi_2} \cdot F^*, \tag{3.16}$$

where F^* is the complex conjugate of F. Thus, we have the solution

$$u(x,t) = 2i\frac{\partial}{\partial x}\log\left(e^{i\xi_1+i\xi_2-\phi_1-\phi_2}\frac{F^*}{F}\right) = -2(k_1+k_2) + 2i\frac{\partial}{\partial x}\log\left(\frac{F^*}{F}\right),$$

which is real. Satsuma and Ishimori [SI79] now prove that (3.2) holds for this u under the additional conditions

$$(c_1-c_2)^2 > (|k_1|+|k_2|)^2, \quad \phi_1\phi_2A_{12} > 0, \tag{3.17}$$

and that the explicit 2-periodic wave solution is

$$u(x,t) = 2U_1(x,t)/U_2(x,t), \tag{3.18}$$

where

$$\begin{aligned} U_1(x,t) &= e^{A_{12}/2}(k_1+k_2)\sinh(\phi_1+\phi_2) + e^{-A_{12}/2}(k_1-k_2)\sinh(\phi_1-\phi_2) \\ &\quad + 2(k_1\sinh\phi_1\cos\xi_2 + k_2\sinh\phi_2\cos\xi_1) \\ U_2(x,t) &= e^{A_{12}/2}\big(\cosh(\phi_1+\phi_2) + \cos(\xi_1+\xi_2)\big) + e^{-A_{12}/2}(\cosh(\phi_1-\phi_2) \\ &\quad + \cos(\xi_1-\xi_2)) + 2(\cosh\phi_2\cos\xi_1 + \cosh\phi_1\cos\xi_2). \end{aligned}$$

Since our numerical methods solve the periodic BO equation, we need both of the two periodic waves to be periodic on the same interval. Since $\cos(x)$ is 2π periodic, each of the two waves will have period $2\pi/k_j$ for $j = 1,2$. So, when we solve the equation on an interval I, we need to choose commensurable wave numbers, i.e. $k_{j,n} = 2\pi n/I$ for $n \in \mathbb{N}$. Then $k_{j,1}$ will have one period on I, $k_{j,2}$ will have two, and so on.

3.1.4 N-periodic Wave Solution

We can obtain general N-periodic wave solution by expanding upon the idea of the 2-periodic wave solution, including the interactions of N waves. This solution can be found to solve the Hirota bilinear form in equation (3.4) by induction, but the conditions needed to derive equation (3.2) has only been proved for $N = 3$. The general N-periodic solution can be expressed like

$$F = \sum_{\mu=0,1} \exp\left(\sum_{j=1}^{N} \mu_j (i\xi_j + \phi_j) + \sum_{i<j}^{(N)} \mu_i \mu_j A_{ij} \right) \tag{3.19}$$

$$G = \sum_{\mu=0,1} \exp\left(\sum_{j=1}^{N} \mu_j (i\xi_j - \phi_j) + \sum_{i<j}^{(N)} \mu_i \mu_j A_{ij} \right), \tag{3.20}$$

where

$$\xi_j = k_j(x - c_j t) + \xi_j^0, \tag{3.21}$$

$$c_j = k_j \coth \phi_j, \tag{3.22}$$

$$\phi_j/k_j > 0 \tag{3.23}$$

$$\exp A_{ij} = \left((c_i - c_j)^2 - (k_i - k_j)^2\right) / \left((c_i - c_j)^2 - (k_i + k_j)^2\right). \tag{3.24}$$

The sum, $\sum_{\mu=0,1}$, is the summation over all combinations of $\mu_1 = 0,1$, $\mu_2 = 0,1, \cdots, \mu_N = 0,1$, and $\sum_{i<j}^{(N)}$ is the summation over all combinations of the N μ, with the condition $i < j$. To obtain a real solution we have to choose the phase constants correctly, like we did for the 2-periodic solution. We won't go any further here, since we will only make use of the 2-periodic solution as test problem.

3.2 Soliton Solutions

We will now find the rational solitary solutions by taking the limit $k \to 0$. The 2-soliton solution will be used to compare our results to those of Thomée and Vasudeva Murthy.

3.2.1 1-Soliton Solution

We consider for equation (3.9) the case where $k \ll 1$ and $O(c) = 1$. We use that $\tanh\phi = k/c$, and note that $\tanh\phi \approx \phi$ for small ϕ. Then we have the following Taylor series

$$\operatorname{sech}\phi = 1 - (k/c)^2/2 + O(k^4).$$

Also, choosing the phase constant to be $\xi_0 = \pi$, we get

$$\cos\xi = -\cos(k\theta) = -\left(1 - k^2\theta^2/2\right) + O(k^4), \quad \text{where} \quad \theta = x - ct.$$

So, inserted into equation (3.9) we get

$$u(x,t) = \frac{4c}{1 + c^2\theta^2 + O(k^2)},$$

and when we let $k \to 0$ we get the rational 1-soliton solution in [TV98],

$$u(x,t) = \frac{4c}{1 + c^2\theta^2}, \quad \text{where} \quad \theta = x - ct. \tag{3.25}$$

3.2.2 2-Soliton Solution

To get the rational 2-soliton solution, we can study F and G separately and then combine them and take the limits $k_1 \to 0$ and $k_2 \to 0$. We know that $\exp(i(k_j\theta_j + \pi)) = -\exp(ik_j\theta_j)$. Thus, for F and similarly for G in equations (3.15) and (3.16), we get by choosing the real part of the phase constants to be $\xi_j^0 = \pi$,

$$F = 1 - e^{ik_1\theta_1 + \phi_1 - A_{12}} - e^{ik_1\theta_2 + \phi_2 - A_{12}} + e^{ik_1\theta_1 + ik_2\theta_2 + \phi_1 + \phi_2}, \quad \text{for} \quad \theta_j = x - c_jt - d_j,$$

where we have introduced a new real arbitrary phase constant d_j. For $k_1, k_2 \ll 1$, $c_1, c_2 = O(1)$ and $k_1/k_2 = O(1)$ we have

$$e^{ik_j\theta_j + \phi_j} = 1 + k_j(i\theta_j + 1/c_j) + O(k_j^2),$$
$$e^{A_{ij}} = 1 + 4k_ik_j/(c_i - c_j)^2 + O(k^4).$$

Thus,

$$F = k_1k_2\left[(i\theta_1 + 1/c_1)(i\theta_2 + 1/c_2) + 4/(c_1 - c_2)^2 + O(k_i)\right]$$
$$G = k_1k_2\left[(i\theta_1 - 1/c_1)(i\theta_2 - 1/c_2) + 4/(c_1 - c_2)^2 + O(k_i)\right],$$

and inserting this into (3.1) and letting $k_1, k_2 \to 0$, we have

$$u(x,t) = \frac{2c_1c_2\left[c_1\theta_1^2 + c_2\theta_2^2 + (c_1 + c_2)^3/c_1c_2(c_1 - c_2)^2\right]}{\left[c_1c_2\theta_1\theta_2 - (c_1 + c_2)^2/(c_1 - c_2)^2\right]^2 + (c_1\theta_1 + c_2\theta_2)^2}, \tag{3.26}$$

which is the rational 2-soliton solution we find in the article by Thomée and Vasudeva Murthy.

3.3 An Arbitrary Test Problem

As a final test problem, we want to test our methods for an initial condition which is not a known solution to the BO equation. The function we have chosen is

$$u_0(x) = \sin(\pi x)e^{-x^2}. \tag{3.27}$$

We solve this equation on $[-5, 5]$ up to $T = 2$. To compare our methods, we need to find a reference solution using our best method, and fine grid sizes.

Chapter 4

Numerical Methods

We want to solve the periodic problem in (2.3) with the following methods. Define $2N+1$ equidistant nodal points such that they partition the interval $[-L,L]$ into $2N$ intervals of width $h = L/N$. Let $U(\cdot,t) \in \mathbb{R}^{2N}$ be the vector that approximates the solution $u(x_j,t)$ in the points $x_j = jh$ for $j = -N, -N+1, \ldots, N-1$. Because of the periodicity of u we have that $U(x_N,t) = U(x_{-N},t)$. In all spatial discretisations in the following methods, we use periodic boundary conditions, e.g. $U(x_{-N-1},t) = U(x_{N-1},t)$. Let k denote the time step such that we get the solution for the times $t_n = nk$, $n = 0,1,\ldots$. We will also use the notation $U_j^n = U(x_j,t_n)$.

4.1 The Method of Thomée and Vasudeva Murthy

In this section we describe in detail the method derived by Thomée and Vasudeva Murthy [TV98]. We follow their derivation and notation closely.

In order to discretise uu_x and u_{xx} we define

$$\partial U_j = \frac{U_{j+1} - U_j}{h}, \quad \bar{\partial} U_j = \frac{U_j - U_{j-1}}{h}, \quad \hat{\partial} U_j = \frac{U_{j+1} - U_{j-1}}{2h}.$$

For the discretisation of the nonlinear term, uu_x, we follow the much cited article on soliton interaction by Zabusky and Kruskal [ZK65], and choose

$$F_h(U) = \frac{1}{3}\hat{\partial}(U^2) + \frac{1}{3}U\hat{\partial}U = Q_h U \hat{\partial} U, \quad (Q_h U)_j = \frac{1}{3}(U_{j-1} + U_j + U_{j+1}). \tag{4.1}$$

To approximate the term u_{xx}, we choose 2nd order central differences

$$\Delta_h U_j = \frac{1}{h}(\partial - \bar{\partial})U_j = \frac{1}{h^2}(U_{j+1} - 2U_j + U_{j-1}). \tag{4.2}$$

The DFT of the discretisation of u_{xx} in equation (4.2), can be found as follows

$$\begin{aligned}
\widehat{\Delta_h U}_p &= \sum_{j=-N}^{N-1} (\Delta_h U)_j e^{-\pi i j p/N} = \sum_{j=-N}^{N-1} 2h^{-2}\left(\frac{U_{j+1}+U_{j-1}}{2} - U_j\right) e^{-\pi i j p/N} \\
&= \sum_{j=-N}^{N-1} 2h^{-2}\left(\frac{1}{2}\left(U_{j+1}e^{-\pi i p(j+1)/N}e^{\pi i p/N} + U_{j-1}e^{-\pi i p(j-1)/N}e^{-\pi i p/N}\right) - U_j e^{-\pi i j p/N}\right) \\
&= 2h^{-2}\left(\frac{1}{2}\left(e^{\pi i p/N} + e^{-\pi i p/N}\right) - 1\right)\hat{U}_p = 2h^{-2}(\cos(\pi p/N) - 1)\hat{U}_p.
\end{aligned}$$

From Lemma 1 we get the DFT of the periodic discrete Hilbert transform, and in total we have

$$\widehat{\tilde{H}_h \Delta_h U}_p = -i\widetilde{\operatorname{sign}}(p)\widehat{\Delta_h U}_p = -2i\widetilde{\operatorname{sign}}(p)h^{-2}(\cos(\pi p/N) - 1)\hat{U}_p. \tag{4.3}$$

We will make use of this property when solving the iterative scheme formulated next. We now discretise the equation in time. In the following we denote $U^n = U(\cdot, t_n)$, and let

$$\bar{\partial}_t = \frac{U^n - U^{n-1}}{k}, \quad \text{and} \quad \bar{U}^{n-1/2} = \frac{U^n + U^{n-1}}{2}. \tag{4.4}$$

Then, in total, the Crank-Nicolson scheme is

$$\begin{aligned}
\bar{\partial}_t U^n + F_h(\bar{U}^{n-1/2}) - \tilde{H}_h \Delta_h \bar{U}^{n-1/2} &= 0, &&\text{for} \quad n \geq 1, \\
U_j^0 &= u_0(x_j), &&\text{for} \quad j = -N, \ldots, N-1.
\end{aligned} \tag{4.5}$$

This is a nonlinear equation for U^n given U^{n-1}. We want to write this equation such that we can solve it using an iterative scheme. For a simpler notation we let $W = U^n$, and rewrite (4.5) using 4.4

$$\begin{aligned}
W - U^{n-1} + kF_h(\bar{U}^{n-1/2}) - k\tilde{H}_h \Delta_h \bar{U}^{n-1/2} &= 0 \\
W - \frac{1}{2}k\tilde{H}_h \Delta_h W - \frac{1}{2}k\tilde{H}_h \Delta_h U^{n-1} &= U^{n-1} - kF_h(\bar{U}^{n-1/2}) \\
W - \frac{1}{2}k\tilde{H}_h \Delta_h W &= g - k\bar{F}_h(W),
\end{aligned}$$

where

$$\begin{aligned}
g &= U^{n-1} + \frac{1}{2}k\tilde{H}_h \Delta_h U^{n-1}, \quad \text{and,} \\
\bar{F}_h(W) &= F_h\left(\frac{1}{2}(W + U^{n-1})\right).
\end{aligned}$$

From this we define the iterative scheme

$$\left(I - \frac{1}{2}k\tilde{H}_h\Delta_h\right)W^{j+1} = g - k\bar{F}_h(W^j), \quad \text{for} \quad j \geq 0, W^0 = U^{n-1}. \tag{4.6}$$

In practice what we do when solving this equation is to take the FFT and use the property in equation (4.3). The method is presented in algorithmic form in Algorithm 1.

Both the discretisation of uu_x and u_{xx} are of second order in space. Also, the discretised Hilbert transform is of second order. The Crank-Nicolson scheme in (4.5) is second order in time, so in total, the method is second order in time and space. For a thorough analysis of existence and convergence of the solutions of method defined here, we refer to the original article.

Algorithm 1 The method of Thomée and Vasudeva Murthy

```
Define $U(x_j,0)$ for $j = -N,\dots,N-1$.
Calculate $\mu(p) = -2i\widetilde{\text{sign}}(p)h^{-2}(\cos(\pi p/N) - 1)$,
          $C_1(p) = 1 + k\mu(p)/2$,  and
          $C_2(p) = 1/(1 - k\mu(p)/2)$  for  $p = -N,\dots,N-1$.
for $t = k, 2k, \dots, T$ do
    Transform for $\hat{U}(p) = \mathscr{F}U(\cdot,t-k)(p)$  for  $p = -N,\dots,N-1$.
    Calculate $\hat{g}(p) = \mathscr{F}g(\cdot,t-k)(p) = C_1(p)\hat{U}(p)$  for  $p = -N,\dots,N-1$.
    Assign $W^0(x_j) = U(x_j,t-k)$  for  $j = -N,\dots,N-1$.
    for $l = 1,2,\dots,q$ do
        Calculate $V(x_j) = \bar{F}_h\left(\frac{1}{2}(W^{l-1}(x_j) + U(x_j,t-k))\right)$  for  $j = -N,\dots,N-1$.
        Transform for $\hat{V}(p) = \mathscr{F}V(\cdot)(p)$  for  $p = -N,\dots,N-1$.
        Calculate $\hat{W}(p) = C_2(p)\left(\hat{g}(p) - k\hat{V}(p)\right)$  for  $p = -N,\dots,N-1$.
        Transform for $W^l(x_j) = \mathscr{F}^{-1}\hat{W}(\cdot)(x_j)$  for  $j = -N,\dots,N-1$.
    end for
    Assign $U(x_j,t) = W^q(x_j)$  for  $j = -N,\dots,N-1$.
end for
```

The method requires $2q+1$ FFTs per time-step, where q is the number of iterations for the iterative scheme (4.6). Alternatively, one can replace the for-loop by a while-loop, setting a tolerance for when W is deemed good enough.

4.2 The Semi-Implicit Method of Chan and Kerkhoven

The following method was originally proposed by Chan and Kerkhoven [CK85] for the KdV equation (2.15), and it is the first Fourier pseudospectral method. The method consists in transforming the equation into discrete Fourier space, advancing the equation in time, and then transforming the solution back. The nonlinear term uu_x is advanced in time using a leap-frog discretisation, and the dispersive u_{xx} term is advanced in time using Crank-Nicolson[1]. To save one transformation per iteration, we rewrite the nonlinear term as $uu_x = w_x$, where $w = \frac{1}{2}u^2$.

We discretise and take the DFT of equation (2.3), and once in discrete Fourier space the discretisation in time is

$$\frac{\hat{U}(p,t+k)-\hat{U}(p,t-k)}{2k} = -isp\hat{W}(p,t) + i\,\mathrm{sign}(p)s^2p^2\frac{\hat{U}(p,t+k)+\hat{U}(p,t-k)}{2}. \quad (4.7)$$

The equation can be solved for $\hat{U}(p,t+k)$ by rearranging the terms, without having to solve a system of equations. After rearranging, an explicit expression for $\hat{U}(p,t+k)$ is

$$\hat{U}(p,t+k) = \frac{\hat{U}(p,t-k)\left(1+i\,\mathrm{sign}(p)s^2p^2k\right) - 2ispk\hat{W}(p,t)}{1-i\,\mathrm{sign}(p)s^2p^2k}. \quad (4.8)$$

Note that to calculate $\hat{U}(p,t+k)$ one needs $\hat{U}(p,t-k)$. But it is not necessary to store $\hat{U}(\cdot,t)$ for all t, just for the two previous time steps. Note also that this method requires two initial vectors to be known. For the test problems which are exact solutions to the BO equation, this is no problem. However, using an arbitrary initial condition, we have to find the second initial vector using a one step method, given the first initial vector. We now present the algorithm for the Semi-Implicit method.

[1] Traditionally with Crank-Nicolson, one uses the time steps $t+k$ and t. Chan and Kerkhoven have chosen to call this variant Crank-Nicolson, so we have done the same.

Algorithm 2 The Semi-Implicit method of Chan and Kerkhoven.

```
**Define** $U(x_j,-k)$, $U(x_j,0)$ for $j=-N,\dots,N-1$.
Transform for $\hat{U}(p,-k)=\mathscr{F}U(\cdot,-k)(p)$ and
        $\hat{U}(p,0)=\mathscr{F}U(\cdot,0)(p)$ for $p=-N,\dots,N-1$.
Calculate the coefficients $C_1(p)=1/(1-i\,\text{sign}(p)s^2p^2k)$ and
        $C_2(p)=1+i\,\text{sign}(p)s^2p^2k$ for $p=-N,\dots,N-1$.
**for** $t=0,k,\dots,T-k$ **do**
    Calculate $W(x_j,t)=\frac{1}{2}U(x_j,t)^2$ for $j=-N,\dots,N-1$.
    Transform for $\hat{W}(p,t)=\mathscr{F}W(\cdot,t)(p)$ for $p=-N,\dots,N-1$.
    Calculate $\hat{U}(p,t+k)=C_1(p)\big(C_2(p)\hat{U}(p,t-k)-2ispk\hat{W}(p,t)\big)$
        for $p=-N,\dots,N-1$.
    Invert for $U(x_j,t+k)=\mathscr{F}^{-1}\hat{U}(\cdot,t+k)(x_j)$ for $j=-N,\dots,N-1$.
**end for**
```

All transforms are done using the FFT, and we only need two FFTs per time step.

4.3 Operator Splitting with Taylor Expansion

This splitting method was proposed by Nouri and Sloan [NS89] for the KdV equation (2.15), and it is the second Fourier pseudospectral method. We use the solution to the nonlinear equation

$$u_t+uu_x=0, \tag{4.9}$$

as initial condition for the linear equation

$$u_t-\tilde{H}u_{xx}=0. \tag{4.10}$$

From equation (4.9) we can deduce the following identity

$$\frac{\partial^q u}{\partial t^q}=(-1)^q\frac{\partial^q}{\partial x^q}\left(\frac{u^{q+1}}{q+1}\right) \quad \text{for} \quad q=1,2,\dots \tag{4.11}$$

Now we can find the Taylor expansion of $u(x,t+k)$ around (x,t) in terms of partial derivatives in x. This will serve as an approximation to the solution of (4.9) at time $t+k$, given $u(x,t)$. The Taylor expansion is

$$u(x,t+k)=u(x,t)+\frac{\partial u}{\partial t}(x,t)k+\frac{1}{2!}\frac{\partial^2 u}{\partial t^2}(x,t)k^2+\frac{1}{3!}\frac{\partial^3 u}{\partial t^3}(x,t)k^3+O(k^4),$$

and using the identity (4.11) we get

$$\begin{aligned} u(x,t+k) = u(x,t) &- k\frac{\partial}{\partial x}\left(\frac{1}{2}u(x,t)^2\right) \\ &+ \frac{1}{2}k^2\frac{\partial^2}{\partial x^2}\left(\frac{1}{3}u(x,t)^3\right) \\ &- \frac{1}{6}k^3\frac{\partial^3}{\partial x^3}\left(\frac{1}{4}u(x,t)^4\right) + O(k^4). \end{aligned} \tag{4.12}$$

In the implementation we will use terms up to $O(k^3)$. We will need the discrete Fourier transform of equation (4.12), and the derivatives are found according to (2.10).

To solve the linear equation (4.10), we transform the equation into discrete Fourier space with the DFT,

$$\hat{U}_t(p,t) = i\,\text{sign}(p)s^2p^2\hat{U}(p,t), \tag{4.13}$$

with the same grid and notation as before. This equation is solved using separation of variables. The Fourier transform of (4.12), $\hat{U}^*(p,t+k)$, is used as initial condition when solving (4.13). Thus, the solution in discrete Fourier space is

$$\hat{U}(p,t+k) = \hat{U}^*(p,t+k)e^{i\,\text{sign}(p)s^2p^2k},$$

which is transformed back to obtain the solution $U(x_j,t+k)$. We now present the method in algorithmic form.

Algorithm 3 Operator Splitting with Taylor Expansion

```
Define $U(x_j,0)$ for $j=-N,\dots,N-1$
Transform for $\hat{U}(p,0)=\mathscr{F}U(\cdot,0)(p)$ for $p=-N,\dots,N-1$.
Calculate the constants $\eta(p)=spk$, $E(p)=e^{i\,\mathrm{sign}(p)p^2s^2k}$ for $p=-N,\dots,N-1$.
for $t=0,k,\dots,T-k$ do
    Calculate $W(x_j,t)=\frac{1}{2}U(x_j,t)^2$, $Y(x_j,t)=\frac{1}{3}U(x_j,t)W(x_j,t)$ and
              $Z(x_j,t)=\frac{1}{4}U(x_j,t)Y(x_j,t)$ for $j=-N,\dots,N-1$
    Transform for $\hat{W}(p,t)=\mathscr{F}W(\cdot,t)(p)$, $\hat{Y}(p,t)=\mathscr{F}Y(\cdot,t)(p)$ and
              $\hat{Z}(p,t)=\mathscr{F}Z(\cdot,t)(p)$ for $p=-N,\dots,N-1$.
    Calculate $\hat{U}^*(p,t+k)=\hat{U}(p,t)-i\eta(p)\hat{W}(p,t)-\eta(p)^2\hat{Y}(p,t)$
              $+i\eta(p)^3\hat{Z}(p,t)$ for $p=-N,\dots,N-1$.
    Calculate $\hat{U}(p,t+k)=E(p)\cdot\hat{U}^*(p,t+k)$ for $p=-N\dots,N-1$.
    Invert for $U(x_j,t+k)=\mathscr{F}^{-1}\hat{U}(\cdot,t+k)(x_j)$ for $j=-N,\dots,N-1$.
end for
```

When we take the expansion up to terms of order $O(k^3)$ we have to use four FFTs per iteration.

4.4 Operator Splitting with Lax-Friedrichs

As in the splitting method with Taylor expansion, we use the numerical solution to the nonlinear equation

$$u_t+uu_x=u_t+g(u)_x=0, \quad \text{where} \quad g(u)=\frac{1}{2}u^2, \tag{4.14}$$

as initial condition for the linear equation (4.10). To numerically solve equation (4.14), we use the Lax-Friedrichs (LxF) scheme

$$U_j^{n+1}=\frac{1}{2}\left(U_{j+1}^n+U_{j-1}^n\right)-\frac{k}{2h}\left(g(U_{j+1}^n)-g(U_{j-1}^n)\right).$$

The stability of this scheme is subject to the CFL condition

$$\frac{u_{\max}k}{h}\leq 1.$$

When we choose the global time step, k, and the spatial grid, h, we may breach the CFL condition. We can however choose finer time steps, l, to use with the LxF scheme, as long as the new time steps add up to one global time step. To do that we calculate the maximal velocity, $u_{\max}$, and determine the preliminary new time step $l_p = h/u_{\max}$. To find the number of inner time steps, m, we calculate $m = \lceil \frac{k}{l_p} \rceil$. Now we get the inner time step calculating $l = k/m$. This method will ensure that we don't breach the CFL condition, while having the inner time steps adding up to one global time step. The LxF scheme is of first order in time, and second order in space.

The linear equation is solved exactly like in the previous method by taking the DFT and using separation of variables. The method in algorithmic form is presented next.

Algorithm 4 Operator Splitting with Lax-Friedrichs

```
Define $U(x_j, 0)$ for $j = -N, \dots, N-1$
Calculate the constant $E(p) = e^{i\,\mathrm{sign}(p)p^2 s^2 k}$ for $p = -N, \dots, N-1$.
for $t = 0, k, \dots, T-k$ do
    Determine the time step, $l$, and the number of time steps, $m$, for the
    Lax-Friedrichs method.
    Assign $U_j^0 = U(x_j, t)$, for $j = -N, \dots, N-1$.
    for $n = 0, \dots, m-1$ do
        Assign $g_j^n = \frac{1}{2}(U_j^n)^2$ for $j = -N, \dots, N-1$.
        Calculate $U_j^{n+1} = \frac{1}{2}\left(U_{j+1}^n + U_{j-1}^n\right) - \frac{l}{2h}\left(g_{j+1}^n - g_{j-1}^n\right)$ for $j = -N, \dots, N-1$.
    end for
    Transform for $\hat{U}^*(p, t+k) = \mathscr{F}U^m(p)$ for $p = -N, \dots, N-1$.
    Calculate $\hat{U}(p, t+k) = E(p) \cdot \hat{U}^*(p, t+k)$ for $p = -N \dots, N-1$.
    Invert for $U(x_j, t+k) = \mathscr{F}^{-1}\hat{U}(\cdot, t+k)(x_j)$ for $j = -N, \dots, N-1$.
end for
```

The Fourier transform is only used to solve the linear equation, so we only use two FFTs per time step.

4.5 Comments on Implementation

We have written all our code in MATLAB. One of the main benefits of writing our code in MATLAB, is that manipulations on vectors are done very easily. There is no need to loop over points in space because of MATLAB's built in element wise multiplication of vectors. Element wise multiplication is done using *array multiply* ".*", i.e. $X.{*}Y$ is the element wise multiplication of the same length vectors X and Y. The following code snippet shows how the Semi-Implicit method is implemented. This function returns the solution only in the last time step.

```
function [V,t] = SI(type,x,params)
% Get parameters
N = params.N; T = params.T; k = params.k; s = params.s;

% Pre-allocating
U = zeros(2,2*N+1);

% Get initial condition
U(1,:) = exactSolution(type,x,-k,params);
U(2,:) = exactSolution(type,x,0,params);
FU(1,:) = fft(U(1,1:2*N));
FU(2,:) = fft(U(2,1:2*N));

% Constant stuff
p = [0:N-1 0 -N+1:-1]*s;
coeff1 = 1./(1 - 1i*sign(p).*(p.^2)*k);
coeff2 = 1 + 1i*sign(p).*(p.^2)*k;

i = 2;
for t = k:k:T
    W = 0.5*U(mod(i+1,2)+1,1:2*N).^2;
    FW = fft(W);

    FU(mod(i,2)+1,:) = coeff1.*(coeff2.*FU(mod(i,2)+1,:) - 1i*2*k*p.*FW);
    U(mod(i,2)+1,1:2*N) = ifft(FU(mod(i,2)+1,:));
    U(mod(i,2)+1,2*N+1) = U(mod(i,2)+1,1);

    i = i+1;
end
V = U(mod(i-1,2)+1,:);
end
```

The function takes as input the test problem type, the points in space, and a struct containing all parameters. Since we only want to return the last time step, for each for-loop we overwrite the array containing the solutions which are no longer needed to advance the solution. The only for-loop needed is the one that loops over all time steps. Also note that since MATLAB's definition of the FFT is different than what we have, we have to shift the vector p.

Chapter 5

Comparison of the Numerical Methods

In this chapter we compare our methods for different test problems. The comparison is based on the following values:

1) Time elapsed. The time in the tables is the time elapsed by the method, excluding time spent determining parameters, initial conditions, plots, etc.

2) The maximal point-wise error at the last time step, $||e||_\infty$, where e is the error vector $e = U - u$, U is the numerical solution, and u is the exact solution at the nodal points.

3) The scaled 2-norm at the last time step, defined by

$$||e|| = \left(\frac{1}{2N} \sum_{j=-N}^{N-1} |e_j|^2 \right)^{1/2} = \frac{1}{\sqrt{2N}} ||e||_2,$$

where e is the error defined above.

5.1 Reproducing the Results of Thomée and Vasudeva Murthy

These first two tests are done to see whether we can reproduce the results in [TV98] in a satisfactory manner. To compare, we use the same number of points in space and time as T&VM, and impose the same tolerance on the iterative scheme in (4.6). The tolerance is set to 10^{-6}. This specific number was set because lowering the tolerance did not change the error in the first four significant digits. The number of iterations needed for the iterative scheme can be reduced by selecting the extrapolated initial condition $W^0 = 2U^{n-1} - U^{n-2}$. We will write the number of iterations needed in the **It** column in the results table, with the number of iterations with the extrapolated initial condition in parenthesis. The final solution in each time step, for both initial conditions, will of course be the same within the first four significant digits.

5.1.1 1-Periodic Wave Solution

We solve (2.3) on $x \in [-15, 15]$ for $T_{\text{end}} = 10$ and 100. The 1-periodic wave solution (3.10) at $t = 0$ is used as initial condition, with parameter $c = 0.25$.

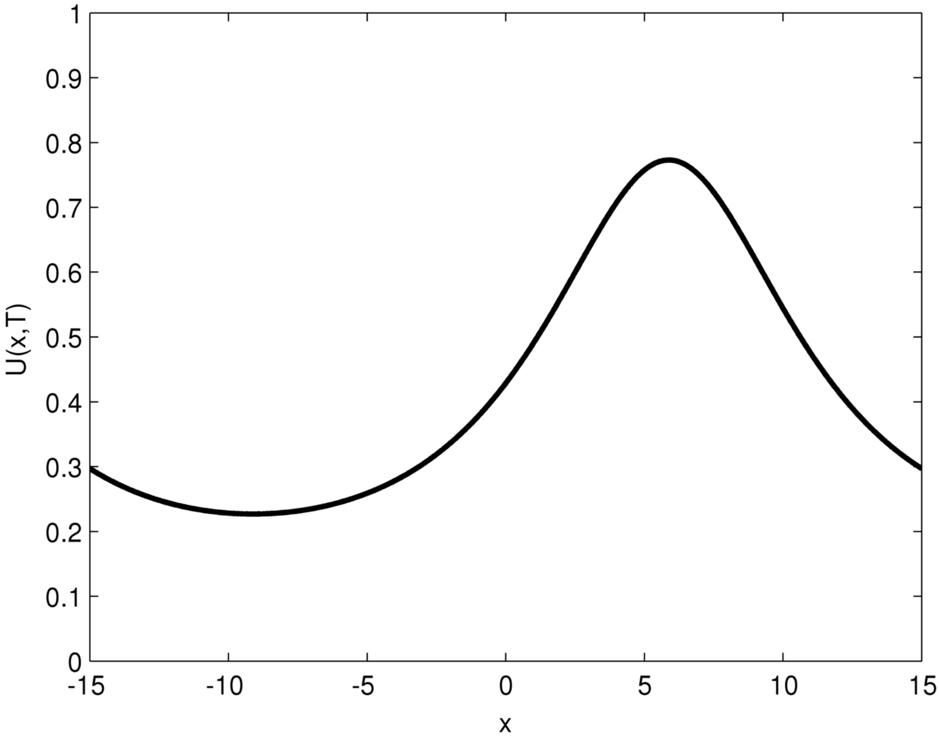


Figure 5.1: The 1-periodic solution at $T = 10$ for $N = 512$ and $k = 0.25$.

t	N	k	**Time** (s)	$\|\|e\|\|_\infty$	$\|\|e\|\|$	**It**
10	256	0.5	0.028	5.91e-5	2.75e-5	6 (5)
	256	0.25	0.038	4.27e-5	1.78e-5	5 (3)
	512	0.5	0.034	3.59e-5	1.79e-5	6 (5)
	512	0.25	0.049	1.48e-5	6.90e-6	5 (3)
	1024	0.5	0.046	3.25e-5	1.58e-5	6 (5)
	1024	0.25	0.067	8.98e-6	4.48e-6	5 (3)
100	256	0.5	0.15	4.08e-4	2.27e-4	6 (5)
	256	0.25	0.25	2.27e-4	1.20e-4	5 (3)
	512	0.5	0.22	2.84e-4	1.64e-4	6 (5)
	512	0.25	0.36	1.02e-4	5.67e-5	5 (3)
	1024	0.5	0.32	2.58e-4	1.48e-4	6 (5)
	1024	0.25	0.55	7.11e-5	4.10e-5	5 (3)

Table 5.1: Results for the 1-periodic solution (3.10) with the method of Thomée and Vasudeva Murthy. The run time listed is for the one with the most iterations.

The values we obtain in the max norm are well within the errors obtained by T&VM for all combinations of spatial points and time steps. For some of the values we get close to a fourth of their error, and we are always below their values. We need, however, one more iteration solving the system in (4.6), except for when we use the extrapolated initial condition for $k = 0.25$. Also in the $||\cdot||$ norm are we well within their values for all combinations of N and k.

At $T = 100$, for increasing N, T&VM get better or equal results except at $N = 1024$ and $k = 0.5$, for which the error is larger than for $N = 512$ and $k = 0.25$. We get the same increase in error for these numbers of points at both $T = 100$ and $T = 10$. In fact, for $T = 100$ we also get a slight increase in error going from $N = 256$ and $k = 0.25$ to $N = 512$ and $k = 0.5$. From the table we can verify the second order convergence. The error in both norms is reduced by a factor of 3.9 (3.99 in the max norm) when we double the points in space, and halve the time step. We will have a closer look at order of convergence in the 1-periodic test problem.

5.1.2 2-Soliton Solution

T&VM also test their method for the 2-soliton solution. We solve (2.3) on $[-100, 100]$, with initial condition (3.26) with parameters $c_1 = 0.3$, $c_2 = 0.6$, and phase constants $d_1 = -30$ and $d_2 = -55$. The spatial period has to be large in order to compare the non-periodic exact soliton solution to the numerical periodic solution. Results for times $T = 10, 90$ and 180 are found in Table 5.2.

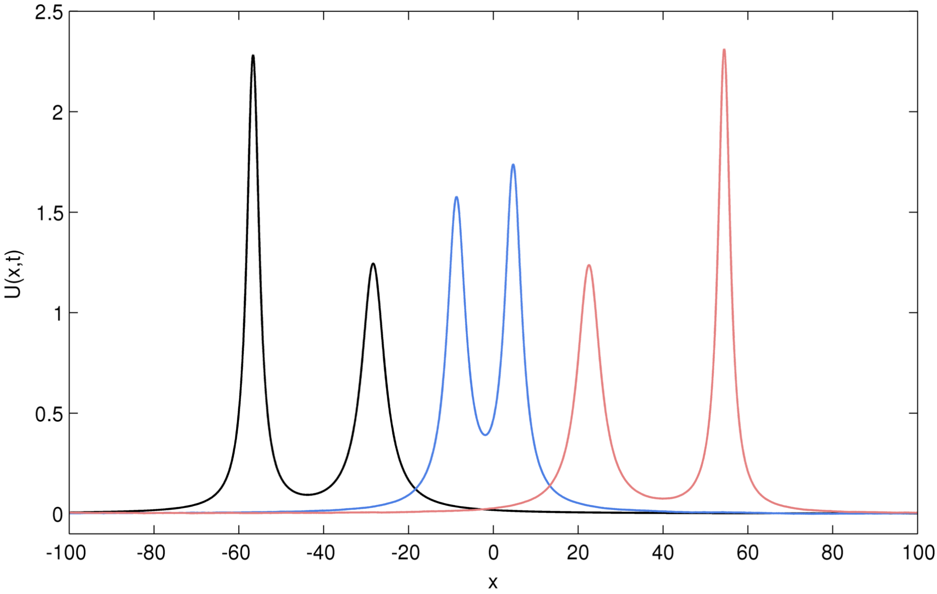


Figure 5.2: The 2-soliton solution at times $t = 0, 90, 180$, in black, blue and red respectively, moving to the right. The parameters are $c_1 = 0.3$, $c_2 = 0.6$, $d_1 = -30$ and $d_2 = -55$, and the grid sizes are $N = 2048$ and $k = 0.1$.

We are well within the errors of T&VM, also for the 2-soliton test problem. The error is much larger than in the 1-periodic wave test problem, as we would expect. Since the speed of each soliton is related to its height, any drop in height due to numerical errors, will reduce the speed and lead to large phase errors. This is most evident after $T = 180$ for $N = 512$ and $k = 0.2$. Figure 5.3 shows this phase error clearly.

The number of iterations per time step is higher than for T&VM, even more so than for the 1-periodic test problem. Now 2 to 3 iterations more are needed. The numbers in the **It** column are the average number of iterations. We have no real explanation for why we need more iterations, but it is not specified by T&VM how the tolerance in the iterative scheme is calculated, so it might be from differences here. We have used the max norm of the difference between the new and the previous W^l.

t	N	k	Time (s)	$\|\|e\|\|_\infty$	$\|\|e\|\|$	It
10	512	0.2	0.10	4.70e-2	5.74e-3	12 (11)
	512	0.1	0.15	4.35e-2	5.28e-3	9 (7)
	1024	0.2	0.16	1.58e-2	2.03e-3	12 (11)
	1024	0.1	0.24	1.21e-2	1.57e-3	9 (7)
	2048	0.2	0.28	7.83e-3	1.15e-3	12 (11)
	2048	0.1	0.40	4.26e-3	7.47e-4	9 (7)
90	512	0.2	0.79	9.79e-2	1.82e-2	10.8 (9.2)
	512	0.1	1.19	9.14e-2	1.69e-2	7.7 (6.2)
	1024	0.2	1.25	3.29e-2	6.17e-3	11.0 (9.6)
	1024	0.1	1.89	2.49e-2	4.81e-3	8.0 (6.4)
	2048	0.2	2.17	1.31e-2	3.20e-3	11.0 (9.7)
	2048	0.1	3.27	8.25e-3	1.96e-3	8.1 (6.5)
180	512	0.2	1.63	5.20e-1	6.78e-2	11.0 (9.5)
	512	0.1	2.41	4.77e-1	6.18e-2	7.8 (6.4)
	1024	0.2	2.60	1.88e-1	2.42e-2	11.2 (9.8)
	1024	0.1	3.85	1.38e-1	1.77e-2	8.2 (6.6)
	2048	0.2	4.47	9.98e-2	1.28e-2	11.3 (9.9)
	2048	0.1	6.65	4.75e-2	6.42e-3	8.2 (6.6)

Table 5.2: Results for the 2-soliton solution (3.26) by the method of Thomée and Vasudeva Murthy. The iteration number is the average number needed per time step to reach the accuracy of 10^{-6} in equation (4.6). The run time listed is for the one with the most iterations.

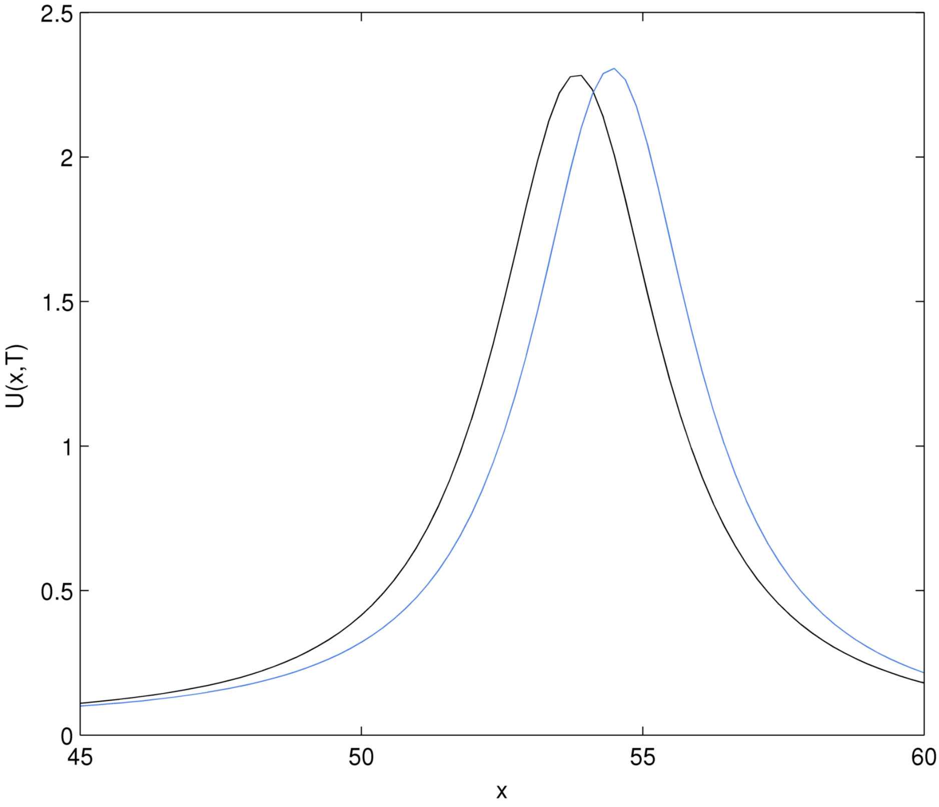


Figure 5.3: The right soliton at $T = 180$ for $N = 512$ and $k = 0.2$. The numerical solution in black, and the exact in blue. The phase error is evident after a small drop in soliton height.

5.2 The 1-Periodic Wave Solution

We want to test all methods with the 1-periodic wave solution (3.10). We choose the same parameter $c = 0.25$, and run the methods to $T = 10$. Because the methods are quite different, we will have to impose an accuracy constrain in one of the norms, rather than specifying the number of spatial points and time steps. We want the methods to reach a maximal error of 1.0×10^{-4} in the scaled 2-norm, and adjust the number of spatial points and time step such that we minimise the run time. The results are summarised in Table 5.3.

#	Method	N	k	Time (s)	$\|\|e\|\|_\infty$	$\|\|e\|\|$
1)	T&VM	100	0.39	0.032	2.53e-4	1.0e-4
2)	Semi-Implicit	8	0.2	0.002	1.78e-4	1.0e-4
3)	Taylor Expansion	9	0.009	0.066	2.20e-4	1.0e-4
4)	Lax-Friedrichs	4400	0.0175	0.88	2.50e-4	1.0e-4

Table 5.3: Comparison of the methods for the 1-periodic wave solution with parameter $c = 0.25$ and $T = 10$.

For the method of T&VM, any fewer steps in space would make the spatial error dominate, making the accuracy impossible to reach. The time step was then chosen such that the accuracy constraint was just met. The same approach was done for the other methods. The best method, by far, is the semi-implicit scheme. We were able to lower N to 8 before the error in space was dominating above 1.0×10^{-4}. From here we could increase the time step to $k = 0.2$. This is half the time step of T&VM, but T&VM still has a run time 16 times longer. The Taylor Expansion method is equal to the Semi-Implicit method, requiring only $N = 9$. However, the time step must be much smaller in the Taylor Expansion method than in the Semi-Implicit method, which obviously leads to a much longer run time. Thus, T&VM is the second best method for the 1-periodic test problem, and the two splitting methods are the two worst.

In last place we have the The Lax-Friedrichs method. It requires $N = 4400$ to reach the accuracy constraint. The time step was chosen by trial and error, but for all spatial grid sizes, the smallest error was found looking for a time step about 5 times the spatial grid size. This results in 4 iterations of the Lax-Friedrichs scheme per global time step.

5.3 Numerical Verification of Order

To test and verify the order of convergence for our methods, we will use the 1-periodic test problem. We choose the same parameters as in the previous section, and measure the error in the $||\cdot||$ norm.

5.3.1 Order in Space of the FPS Methods

First we want to demonstrate the very rapid convergence in space of the Semi-Implicit and the Taylor Expansion method. We will denote them the FPS methods when we talk about both. As we can see in the 1-periodic wave test problem, these methods require only 16 ($2N$) and 18 steps in space to reach the prescribed accuracy. The two others require far more. In spectral methods in general, one talks about "infinite order", and "exponential" and "spectral convergence". These are all names for when the method converges faster than any power, p, of $1/N^p$ as $N \to \infty$. We refer to [Boy01, Tre00] for details.

To indicate the rapid decrease in error, we fix a small time step of $k = 0.00001$, and increase the number of steps in space. We start with $N = 3$ and increase N by 1 until we get to $N = 22$.

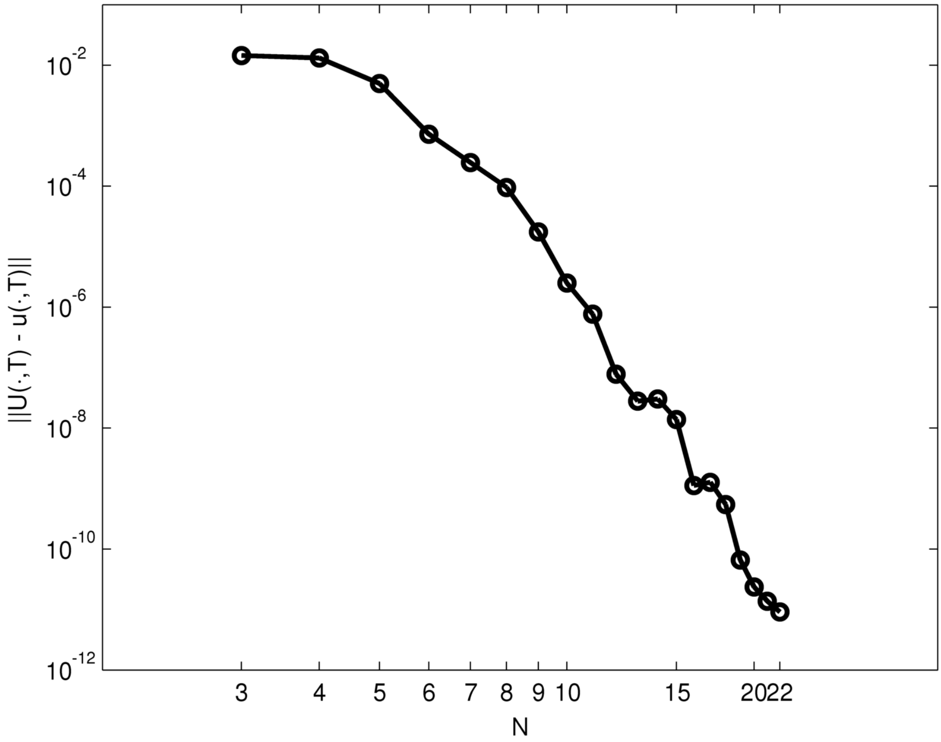


Figure 5.4: Plot of the error against N for the Semi-Implicit method. The time step is fixed at $k = 0.00001$.

As we can see from the plot in Figure 5.4, the error decreases very rapidly. For example, going from $N = 11$ to $N = 12$ reduces the error by 10 times. The error increases slightly from $N = 13$ to $N = 14$ and from $N = 16$ to $N = 17$. But over all the error decreases very rapidly compared to e.g. pure finite difference methods. At the very end, we see the tendency that the error due to discretisation in time starts to dominate.

In Figure 5.5 we have a plot of the error using the Taylor Expansion method. Here the error is dominated earlier by discretisation in time, so no further partitioning in space leads to better results. Therefore we stopped at $N = 16$. The error decreases quickly, though not as quickly as with the Semi-Implicit method. We also notice that line is smoother, with no "bumps" as with the Semi-Implicit method.

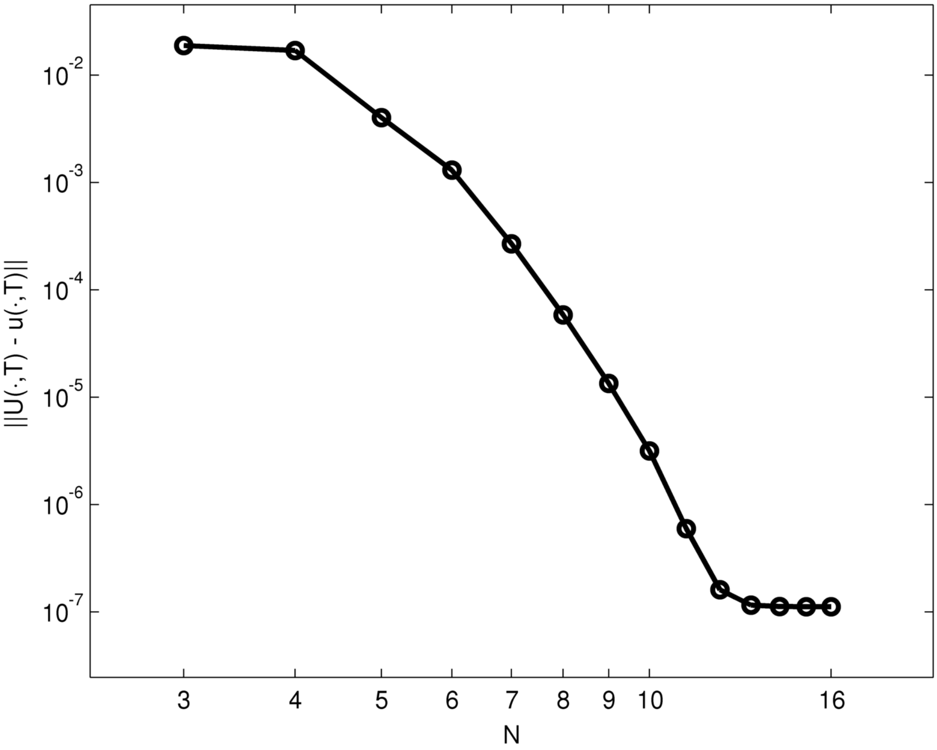


Figure 5.5: Plot of the error against N for the Operator Splitting with Taylor Expansion method. The time step is fixed at $k = 0.00001$.

5.3.2 Order in Time of the FPS Methods

We now want to check the order in time for these two methods. From the previous section we see that by choosing $N = 20$, the error in space should not dominate for either method, choosing appropriate time steps. We start with $M = 10$ steps in time, which for $T = 10$ gives us a time step of $k = 1$. Then we double the points in time four times until we reach $M = 160$, i.e. $k = 0.0625$. Figures 5.6 and 5.7 show the order for the Semi-Implicit and Operator Splitting with Taylor Expansion methods respectively. We can see from the two plots that the Semi-Implicit method is of order 2 in time, and the Taylor Expansion method is of order 1 in time.

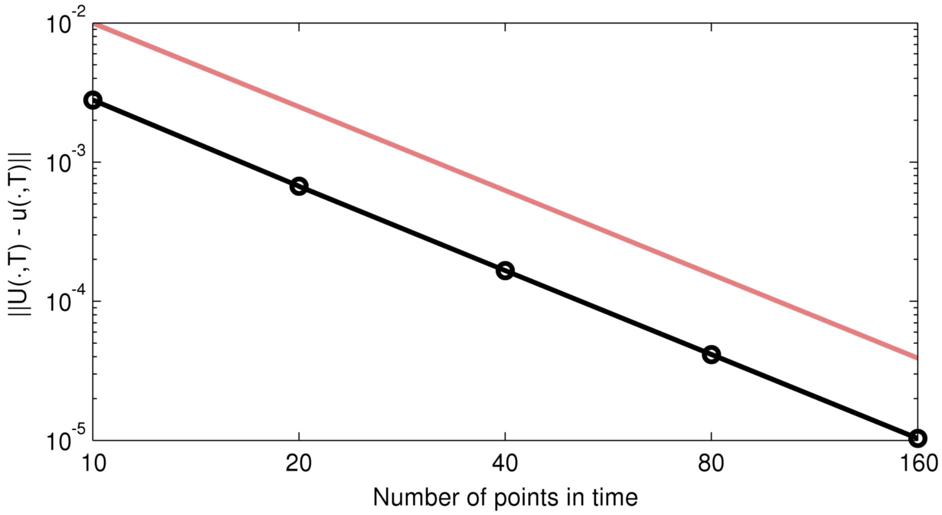


Figure 5.6: Plot of the error against number of points in time for the Semi-Implicit method. The red line has slope −2. The number of points in space is fixed at 40, i.e. $N = 20$.

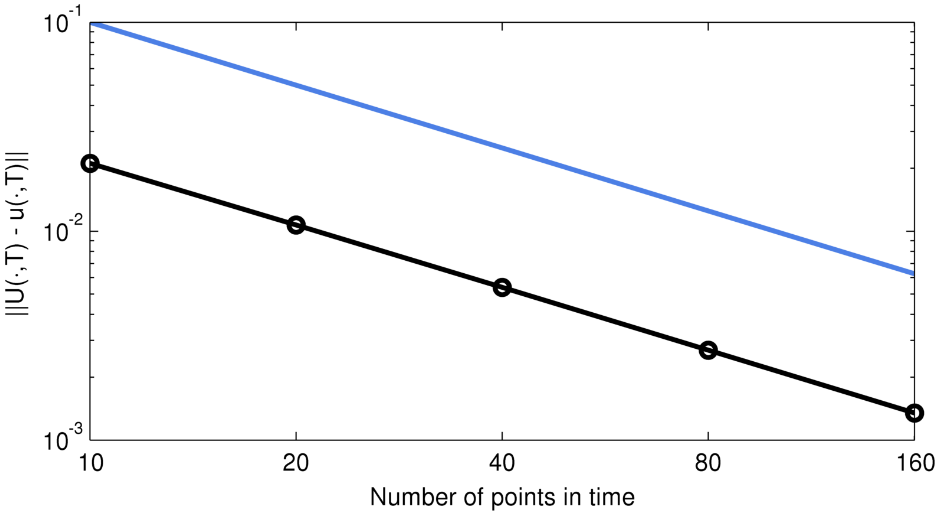


Figure 5.7: Plot of the error against number of points in time for the Operator Splitting with Taylor Expansion method. The blue line has slope −1. The number of points in space is fixed at 40, i.e. $N = 20$.

5.3.3 Order of T&VM and LxF

First we consider the order of convergence in space for the method of T&VM. We must set a time step small enough such that the error due to discretisation in time does not dominate. We choose a time step of $k = 0.001$, and start with $N = 100$. Then we double the steps in space 5 times so we end up with $N = 3200$. Figure 5.8 verifies that the implementation is of second order in space.

For the order of convergence in time, we fix a number of points in space $N = 10000$. We then increase the number of points in time from 10, doubling all the way, up to 320, i.e. $k = 1$ to $k = 0.03125$. Figure 5.9 verifies that the method is of order two also in time.

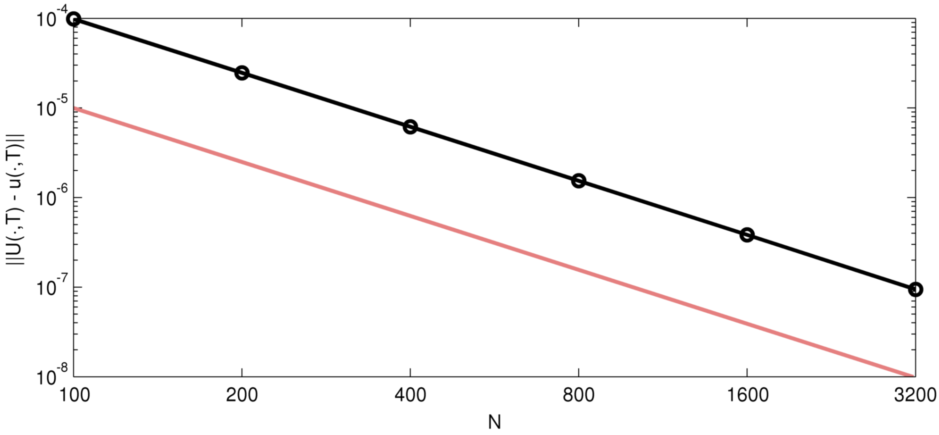


Figure 5.8: Plot of the error against N for the method of T&VM. The red line has slope -2. The time step is fixed at $k = 0.001$.

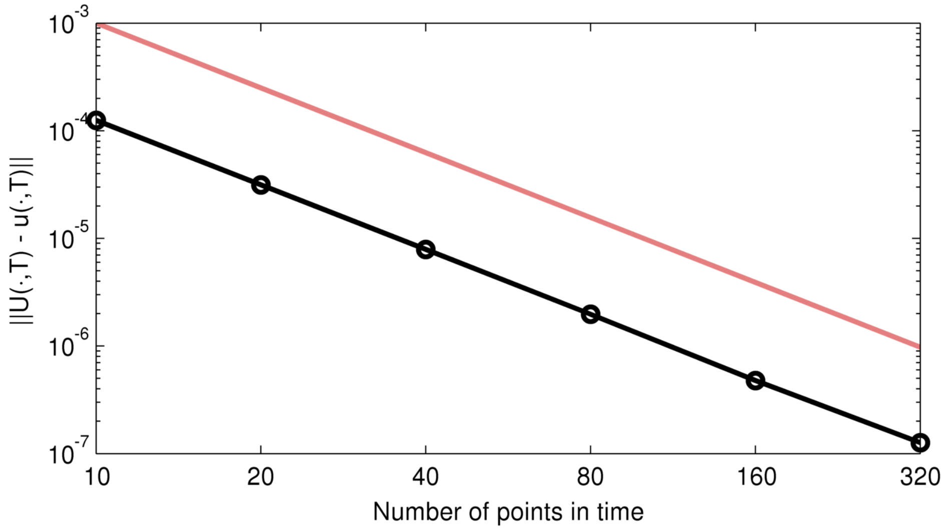


Figure 5.9: Plot of the error against number of points in time for the method of T&VM. The red line has slope -2. Here N is fixed at $N = 10000$.

We now consider the Operator Splitting with Lax-Friedrichs method. For the Lax-Friedrichs step in this method, the time and spatial discretisation parameters are connected. Thus, to check order of convergence, we will double the number of points in space, and at the same time halve the time step. We start with $N = 200$. For now we are really only interested in the rate of the decrease in error, not the magnitude of the error itself. But still, we can use that the error seems to be at its smallest when the time steps is about 5 times the step size in space. So, initially we choose $k = 5h = 5L/N = 0.375$. Figure 5.10 indicates that the methods is first order in total.

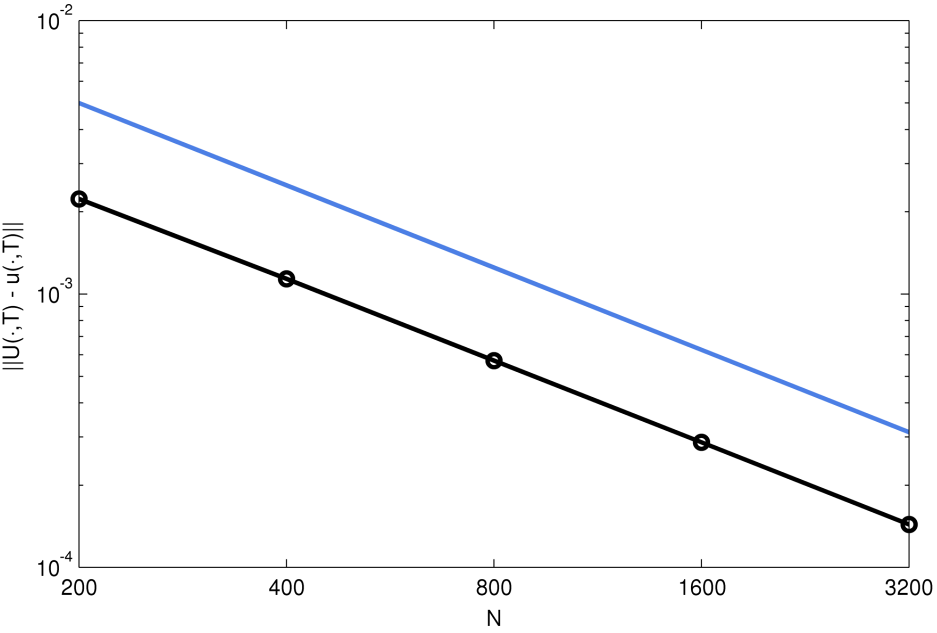


Figure 5.10: Plot of the error against N for the Operator Splitting with Lax-Friedrichs method. The blue line has slope -1. The time step starts at $k = 0.375$ and is halved each time.

5.4 The 2-Periodic Wave Solution

The second test problem is the 2-periodic wave solution. We use (3.18) on $[-L, L] = [-10, 10]$ as initial condition, with parameters

$$k_1 = \pi/L, \quad k_2 = 2\pi/L, \qquad \phi_1 = 0.15, \quad \phi_2 = 1, \qquad \text{and} \qquad T = 10,$$

and phase constants

$$\xi_1^0 = -5 \quad \text{and} \quad \xi_2^0 = 0.$$

We select k_1 and k_2 such that we get one wave train with one period on the spatial interval, and one with two periods. The time period and phase constants are selected such that the wave train with one period interacts with the first peak of the wave train with two periods. This test problem is a lot harder to compute, mainly because of the amplitude of the waves being larger. The constants are selected such that they satisfy the conditions (3.17), together with the condition $\phi_j/k_j > 0$.

Since this test problem is a lot harder than the first one, we require that the maximal error in the $||\cdot||$-norm is 1.0×10^{-3}, i.e. 10 times larger than for the previous test problem. The goal is to select the time and spatial step sizes such that we minimise the run time. The results are in Table 5.4.

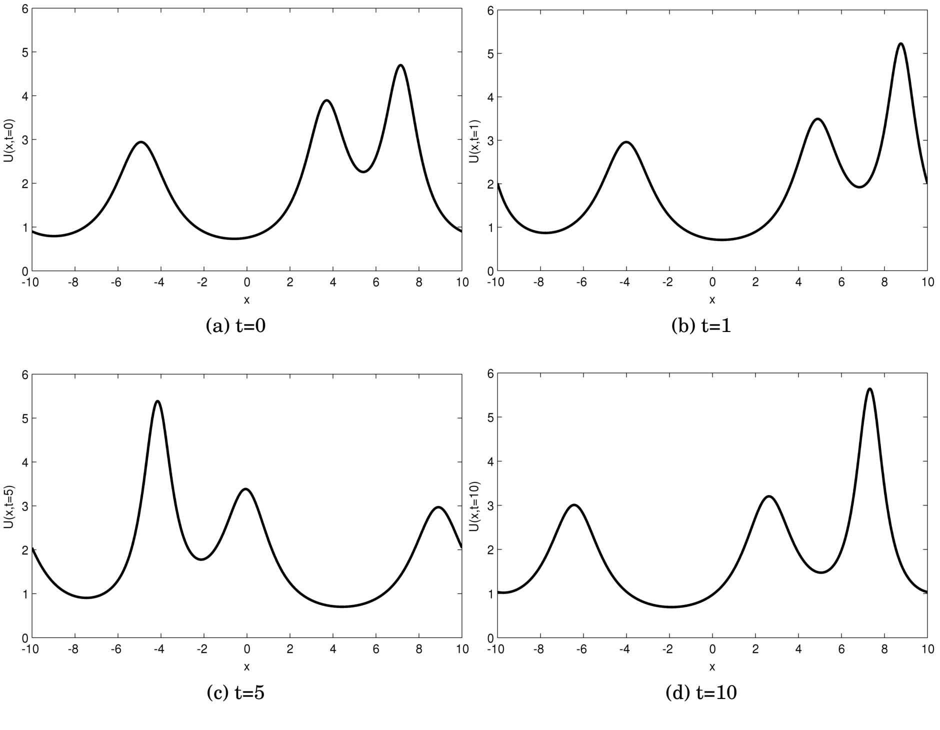
#	**Method**	N	k	**Time** (s)	$\|\|e\|\|_\infty$	$\|\|e\|\|$
1)	T&VM	2000	0.0075	3.89	3.62e-3	1.0e-3
2)	Semi-Implicit	39	0.003	0.13	3.97e-3	9.8e-4
3)	Taylor Expansion	39	0.00065	1.03	3.49e-3	1.0e-3
4)	Lax-Friedrichs	*	*	*	*	*

Table 5.4: Comparison of the methods for the 2-periodic wave solution with parameters $k_1 = \pi/L$, $k_2 = 2\pi/L$, $\phi_1 = 0.15$, $\phi_2 = 1$, $\xi_1^0 = -5$. $\xi_2^0 = 0$, and $T = 10$.

The Semi-Implicit method is again by far the best method. The run time is ten times faster than the second best method, which is the Operator Splitting with Taylor Expansion. Down to third place, from second, comes the method of T&VM.

In the 1-periodic test problem, the method of T&VM had $N = 100$, i.e. a spatial grid size of $h = L/N = 15/100 = 0.15$. To meet the accuracy constraint here (which is not as strict as for the first problem), we had to decrease the grid size to $h = 10/2000 = 0.005$. That is a step size 30 times smaller. For the FPS methods, we only had to decrease the step size about 7.5 times, due to the fast convergence in space. The method of T&VM is still able to cope with the largest time step, over twice as large as the Semi-Implicit. This is consistent with what we found in the first test problem. Larger time steps than in pseudospectral methods is something that T&VM point out in their article, when comparing their method to other methods. The tolerance in the iterative scheme in (4.6) is 10^{-6} here as well.

The Operator Splitting with Lax-Friedrichs does not reach the accuracy constraint in a "reasonable" amount of time. The method really struggles with the more difficult test problem, and the phase error is even more evident than in the 1-periodic case. The constraint can possibly be met, but that would require hours, as opposed to seconds for the other methods. Figure 5.11 shows the solution to the 2-periodic test problem at different times.



(a) t=0

(b) t=1

(c) t=5

(d) t=10

Figure 5.11: The 2-periodic wave solution at different times. The parameters are $k_1 = \pi/L$, $k_2 = 2\pi/L$, $\phi_1 = 0.15$, $\phi_2 = 1$, $\xi_1^0 = -5$. $\xi_2^0 = 0$, and $T = 10$.

5.4.1 Wave Interaction and Phase Shift

An interesting property of the 2-periodic wave solution occurs when the waves interact. Before and after each interaction, the shape of each wave is preserved, but the waves are phase shifted. This is the same property we observe with the soliton solutions, and the N-cnoidal solutions of the KdV equation. When a faster right moving wave passes and interacts with a slower right moving wave, the faster wave is shifted to the right of its path, and the slower is shifted to the left. This is best explained by a figure. Figure 5.12 shows the 2-periodic wave solution, from time $t = 0$ to $T = 20$, with the same parameters as before.

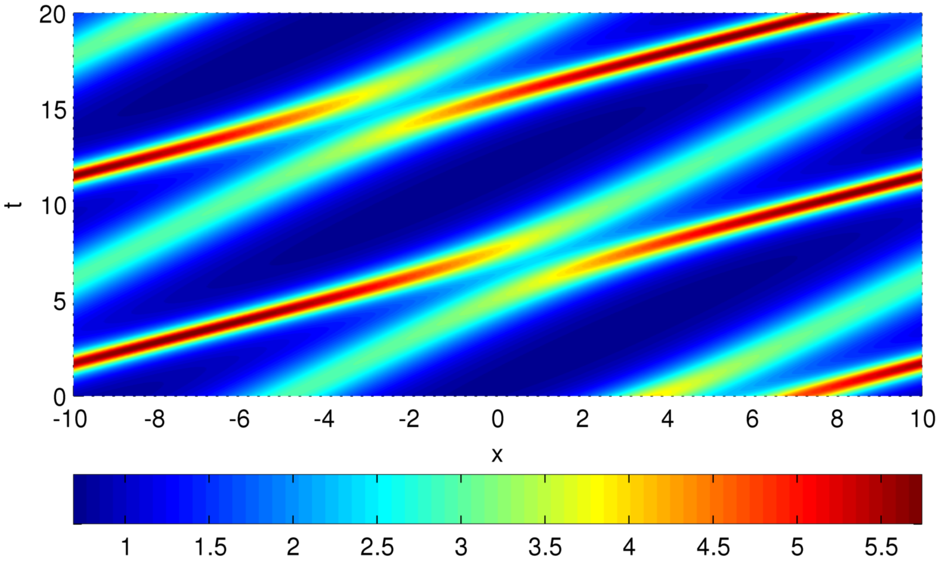


Figure 5.12: The 2-periodic wave solution seen from above. Note the phase shift after the wave interactions.

Figure 5.12 clearly shows how the waves are shifted after the interaction. The interaction between $x = 0$ and $x = 1$ around $t = 5$ causes the faster moving wave in dark red to be shifted to the right, and the slower moving wave in turquoise to be shifted to the left. The same happens when the faster moving wave catches up with the second slower moving wave around $t = 13$. Also notice how the shape is preserved after the interaction.

5.5 The Arbitrary Initial Condition

The last test we run is using the arbitrary initial condition

$$u_0(x) = \sin(\pi x)e^{-x^2},$$

defined in (3.27), which we solve on $[-5,5]$ for time $T = 2$.

So far the Semi-Implicit method has performed best, and we use this method to find a reference solution to which we can compare our methods. A drawback with the Semi-Implicit method is that it requires initial conditions at two time steps. However, the others do not, so we can use one of these to obtain the second initial condition given the first at $t = 0$. We choose the Taylor Expansion method to find the solution at $t = k$, using a time step of $k_I = 10^{-5}k$. Then we have the conditions at $t = 0$ and $t = k$. The reference solution was found using $N = 500$, and a time step $k = 10^{-8}$, such that $k_I = 10^{-13}$. The reference solution was found in 6 hours. We also found a solution with $k = 10^{-7}$, and the maximal point wise discrepancy between this and the better one was 5.9×10^{-7}. To make sure the reference solution does not affect the result, we also did a solution with the Taylor Expansion method. With $N = 500$ and $k = 10^{-8}$, this took 16 hours, and the maximal point wise difference to the Semi-Implicit solution was 4.5×10^{-9}. Thus, the reference solution is deemed good enough for the following test.

We require that the maximal error in the $||\cdot||$-norm is 1.0×10^{-4}. To calculate the error in the nodal points chosen for each method, we find the continuous solution by the Fourier series

$$u_h(x,T) = \frac{1}{2N}\sum_{p=-N}^{N-1} \hat{u}_p(T)e^{ip\frac{\pi}{L}(x+L)},$$

where $N = 500$.

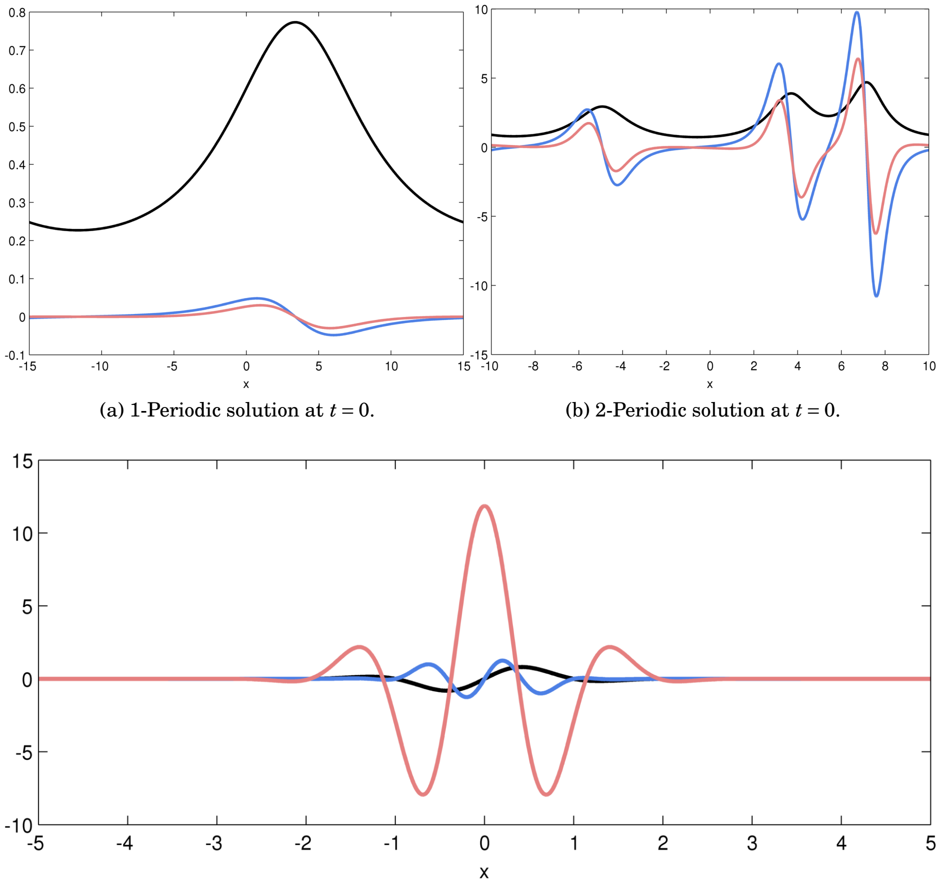
The time step for the Lax-Friedrichs method is chosen to be $k = 5h$. For the Semi-Implicit and Taylor Expansion methods, we chose time steps $k = T/62000$ and $k = T/63000$ respectively. That way we get an integer number of time steps up to $T = 2$, and can compare our methods at the same point in time. The results are found in Table 5.5.

#	Method	N	k	Time (s)	$\|\|e\|\|_\infty$	$\|\|e\|\|$
1)	T&VM	2000	2.50e-4	10.2	2.55e-4	1.0e-4
2)	Semi-Implicit	24	3.22e-5	2.6	1.97e-4	1.0e-4
3)	Taylor Expansion	25	3.17e-5	4.2	2.05e-4	1.0e-4
4)	Lax-Friedrichs	110000	2.27e-4	318.2	2.42e-4	1.0e-4

Table 5.5: Comparison of the methods for the initial condition in (3.27) at $T = 2$. The reference solution is calculated using the Semi-Implicit method with $N = 500$ and $k = 10^{-8}$.

Yet again the Semi-Implicit method is best, but this time it is a much closer race. The Operator Splitting with Taylor Expansion method requires about the same time and spatial step sizes, and takes less than twice the time. The Taylor Expansion method solves the linear equation (4.10) with high accuracy. So, for test problems with many oscillations, where the u_{xx} will be large compared to the other terms, the Taylor Expansion method might have an advantage. Figure 5.13 shows the comparison of the uu_x and Hu_{xx} terms. For the arbitrary initial condition, the Hilbert term is clearly much larger compared to the uu_x term than in the two other test problems. The Lax-Friedrichs method solves the linear equation in the same way as the Taylor Expansion method, but the LxF step requires way too many points in space to be any effective. The method of T&VM needs more steps in space as expected, but can cope with a much larger time step than the Semi-Implicit and Taylor Expansion.

Figure 5.14 shows the solution at different times. We can see small waves breaking out at $t = 0.2$. It is hard to tell because of wave interactions, but it looks like we get a total of 7-8 waves.

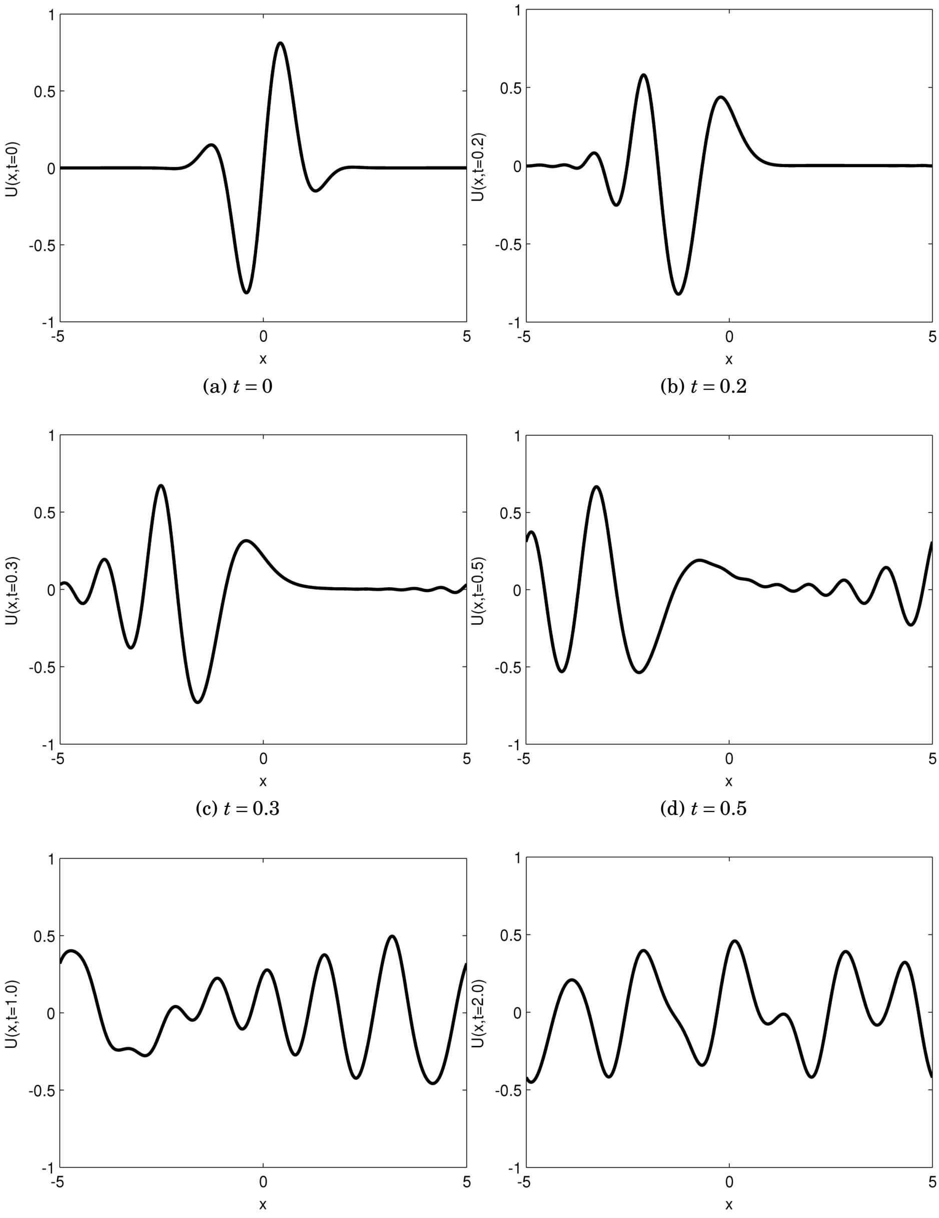


(a) 1-Periodic solution at $t = 0$.

(b) 2-Periodic solution at $t = 0$.

(c) Arbitrary initial condition at $t = 0$

Figure 5.13: Plot of the uu_x and Hu_{xx} terms in the BO equation for different initial conditions. Black is $u(x,0)$, blue is $uu_x(x,0)$ and red is $Hu_{xx}(x,0)$.



(a) $t = 0$

(b) $t = 0.2$

(c) $t = 0.3$

(d) $t = 0.5$

(e) $t = 1.0$

(f) $t = 2.0$

Figure 5.14: The solution with the arbitrary initial condition (3.27) at different times.

Chapter 6

Zero Dispersion Limit

In this chapter, we will study numerically the zero dispersion limit of the Burgers, Korteweg-de Vries and Benjamin-Ono equation. To do this, we introduce a parameter $\epsilon > 0$ in front of the dispersion term in the three equations,

$$u_t + uu_x - \epsilon H u_{xx} = 0, \qquad \text{Benjamin-Ono,} \tag{6.1}$$

$$u_t + uu_x + \epsilon u_{xxx} = 0, \qquad \text{Korteweg-de Vries,} \tag{6.2}$$

$$u_t + uu_x - \epsilon u_{xx} = 0, \qquad \text{Burgers,} \tag{6.3}$$

for $x \in \mathbb{R}$ and $t > 0$. Letting $\epsilon \downarrow 0$, with initial data independent of ϵ, is what we refer to as the zero dispersion limit. If we let $\epsilon = 0$, we have the inviscid Burgers equation,

$$u_t + uu_x = 0. \tag{6.4}$$

Formally, the solutions to the three equations in (6.1)-(6.3) should give us the solution to the inviscid Burgers equation as $\epsilon \downarrow 0$. However, it has been shown both for the KdV equation [LL05a, LL05b, LL05c, DVZ97], and for the BO equation [Mat98, MX11, Xu10], that this is not the case when the solution develops shocks. The inviscid Burgers equation will develop shocks and/or rarefactions in finite time with certain initial conditions. For the Burgers equation, it is known that the solution will approach the solution to the inviscid Burgers equation.

We will now investigate the solutions to the three equations numerically, with smaller and smaller ϵ. As initial condition will we use an exact solution to the inviscid Burgers equation that immediately develops a shock and a rarefaction. We can find the exact weak solution to the inviscid Burgers equation by the method of characteristics.

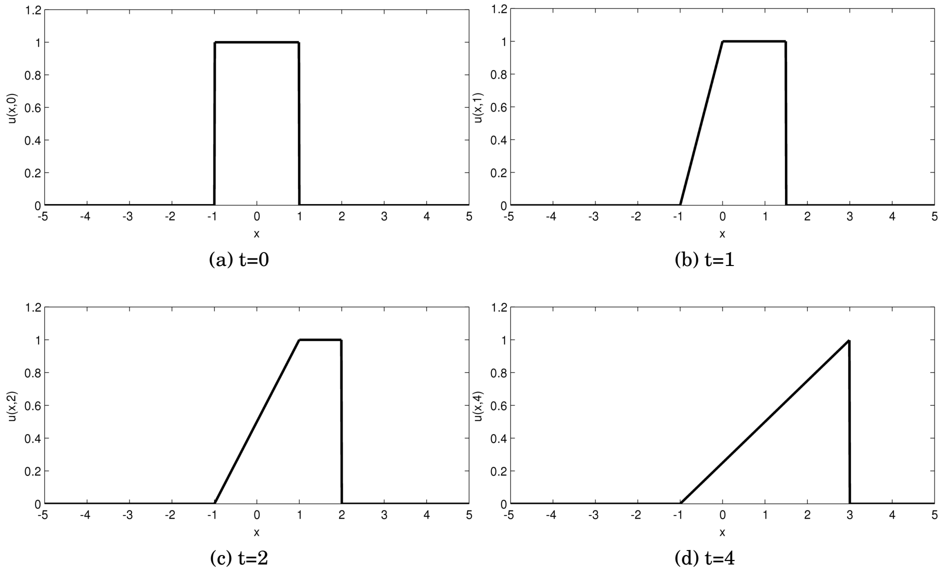
6.1 Weak Solution to the Inviscid Burgers Equation

We have the following initial value problem for the inviscid Burgers equation,

$$u_t + uu_x = 0, \qquad u(x,0) = u_0(x) = \begin{cases} 1, & -1 < x < 1, \\ 0, & \text{otherwise}, \end{cases} \qquad x \in \mathbb{R}, \quad t > 0. \tag{6.5}$$

Immediately we get a shock at $x = 1$, and a rarefaction at $x = -1$. The shock will have a speed of 1/2, and the rarefaction will catch up with the shock after $t = 4$. So for $t = (0,4)$ we have the following weak solution, found by the method of characteristics,

$$u(x,t) = \begin{cases} 0, & x < -1, \\ \frac{x+1}{t}, & -1 < x < t-1, \\ 1, & t-1 < x < \frac{1}{2}t+1, \\ 0, & x > \frac{1}{2}t+1. \end{cases} \tag{6.6}$$



(a) t=0

(b) t=1

(c) t=2

(d) t=4

Figure 6.1: The weak solution to the initial value problem in (6.5).

We have chosen the initial condition such that it is reasonable to approximate the solution numerically with a spatially periodic initial value problem. The reason for doing this, is that all our methods are periodic.

6.2 Problem Setup and Choice of Method

We look for $2L$-periodic solutions to the equations in (6.1)-(6.3), with the conditions

$$u(x,0) = u_0(x) = \begin{cases} 1, & -1 < x < 1, \\ 0, & \text{otherwise}, \end{cases} \tag{6.7}$$

$$u(x+2L,t) = u(x,t),. \tag{6.8}$$

We have chosen to let $L = 5$, and solve the equation up to $T = 2$. We will then solve the equations with ϵ as follows, $\epsilon = 0.01$, 0.001 and 0.0001.

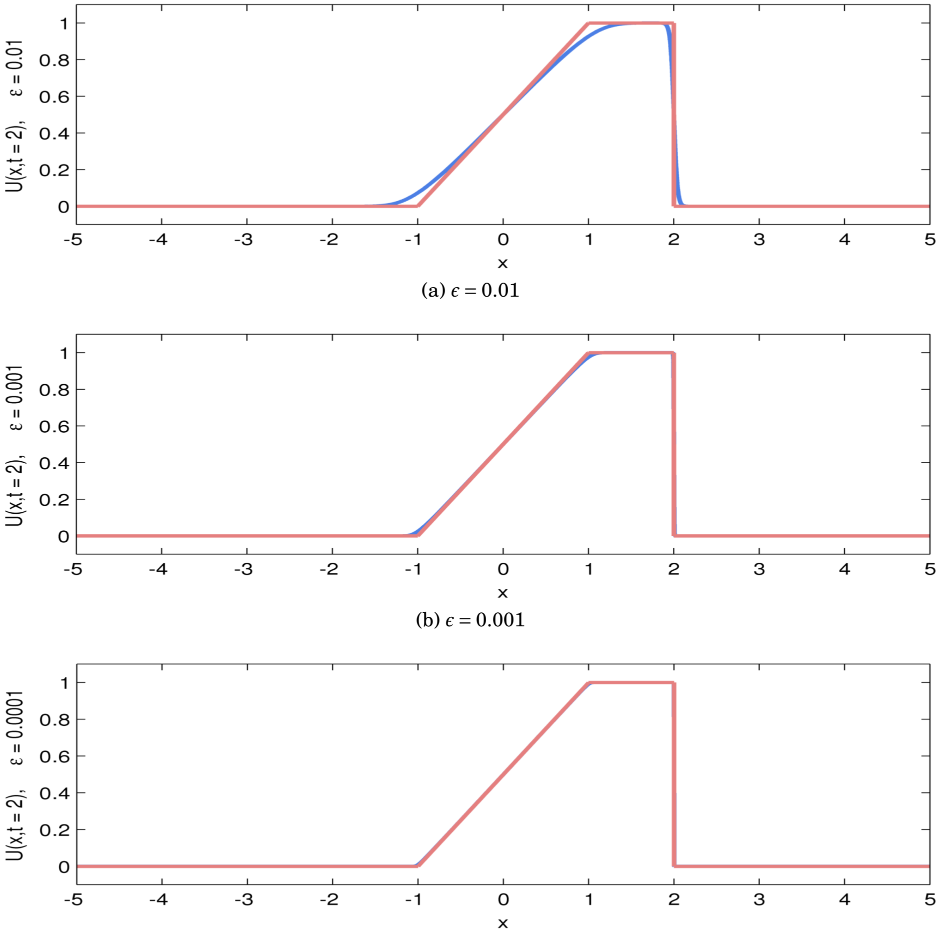
We want to solve all three equations with the same method. The method of Thomée and Vasudeva Murthy can be adapted to solve the KdV and Burgers equation, but it would be a more natural choice to choose one of the other three. The Semi-Implicit and the Operator Splitting with Taylor Expansion method were originally designed for the KdV equation. And the Operator Splitting with Lax-Friedrichs method shares the way of solving the linear dispersive equation with the Taylor Expansion method. So all these tree methods are easy to adapt to solve all three equations, with a small modification in the approximation to the dispersive term. With the discrete Fourier transform, we can approximate the dispersive term in the three equations as follows

$$\begin{aligned} \epsilon\tilde{H}u_{xx} &\approx \mathscr{F}^{-1}\left(i\operatorname{sign}(p)s^2p^2\hat{U}(p,t)\right), \\ \epsilon u_{xxx} &\approx \mathscr{F}^{-1}\left(-is^3p^3\hat{U}(p,t)\right), \\ \epsilon u_{xx} &\approx \mathscr{F}^{-1}\left(-s^2p^2\hat{U}(p,t)\right). \end{aligned}$$

Initial test show that both the Semi-Implicit and the Taylor Expansion method struggle with the discontinuity, and the solutions eventually blow up. The Operator Splitting with Lax-Friedrichs method proves to be the best one in this case. The Lax-Friedrichs step is dissipative and will smooth out any discontinuity. The ratio between the time step and the spatial step determines the amount of smoothing. We will keep this ratio the same for all equations and choice in grid sizes. As we lower ϵ, we have to make sure the step sizes in space and time are small enough to capture any behaviour due to the decreasing dispersion term. We will keep the grid sizes the same for all values of ϵ, for all test problems.

6.3 Zero Dispersion Limit of the Burgers Equation

First we want to verify that the solution of the Burgers equation approaches the solution of the inviscid Burgers equation. We have done three test, with the same grid sizes, $N = 100000$ and $k = 5L/N = 0.00025$, for decreasing $\epsilon = 0.01, 0.001, 0.0001$. The results are plotted in Figure 6.2.



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.001$

(c) $\epsilon = 0.0001$

Figure 6.2: Solution to the Burgers equation in blue, together with the weak solution of the inviscid Burgers equation in red. Here we have $N = 100000$, $k = 5L/N = 0.00025$ and $T = 2$.

As expected, does the solution tend towards the weak solution of the inviscid Burgers equation. The errors in the $||\cdot||$-norm are in Table 6.1.

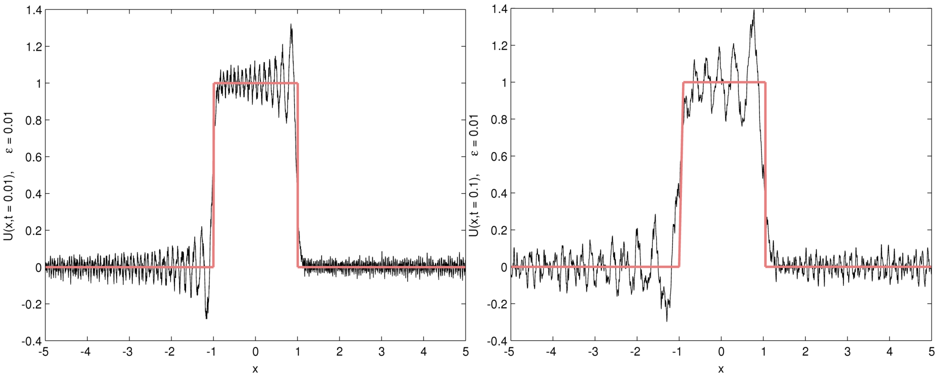
The Burgers Equation	$\boldsymbol{\epsilon} = 0.01$	$\boldsymbol{\epsilon} = 0.001$	$\boldsymbol{\epsilon} = 0.0001$
$\|\|\cdot\|\|$	3.3e-2	9.7e-3	3.3e-3

Table 6.1: Error in the $||\cdot||$-norm for the Burgers equation at $T = 2$ with $N = 100000$, $k = 5L/N = 0.00025$.

6.4 Zero Dispersion Limit of the Korteweg-de Vries Equation

We will now see what happens for the KdV equation when we decrease ϵ. We use the same grid size as for the Burgers equation, $N = 100000$ and $k = 5L/N = 0.00025$. For the KdV equation, it is known that the small dispersion term will force the shock to become travelling waves. And there exists an asymptotic formula that is valid pointwise in the oscillation area around shocks [DVZ97].

The following plots show the solution at two early time steps.



(a) $t = 0.01$ (b) $t = 0.1$

Figure 6.3: Solution to the KdV equation, with $\epsilon = 0.01$, in black, and the weak solution to the inviscid Burgers equation in red.

Immediately, the solution breaks up in many waves at the two discontinuities, and it gets quite chaotic. We also have a lot of high frequency noise. At $T = 2$ the solution looks nothing like the weak solution to the inviscid Burgers equation, and we have omitted that plot. We now decrease ϵ to 0.001, and the solution looks much like the one for $\epsilon = 0.01$. But now the waves have smaller wavelength.

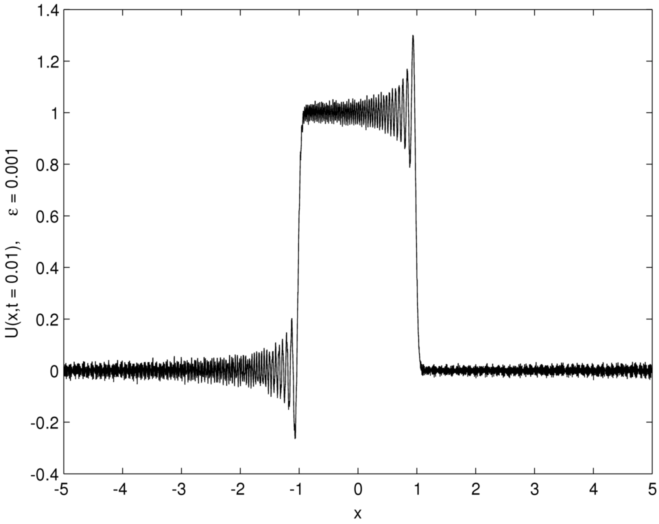
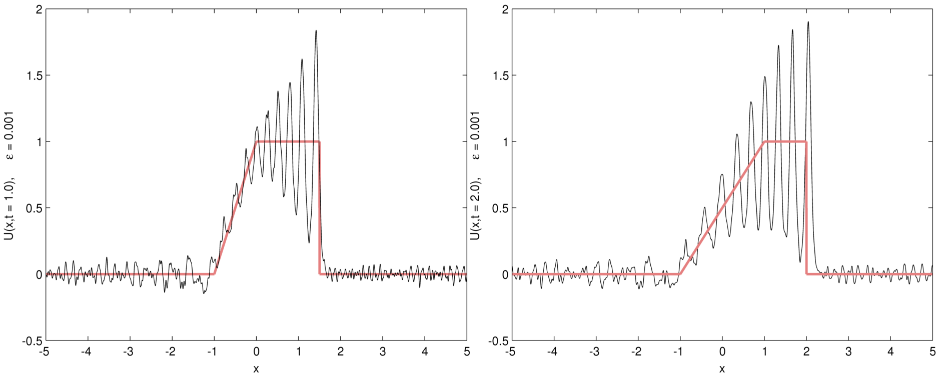


Figure 6.4: Solution to the KdV equation, with $\epsilon = 0.001$, after $t = 0.01$.



(a) $t = 1$, $\epsilon = 0.001$

(b) $t = 2$, $\epsilon = 0.001$

Figure 6.5: Solution to the KdV equation with, $\epsilon = 0.001$, in black. Weak solution to the inviscid Burgers equation in red.

Finally, we run the same test with $\epsilon = 0.0001$. The number of waves breaking out is now so many that we cannot separate them in this plot.

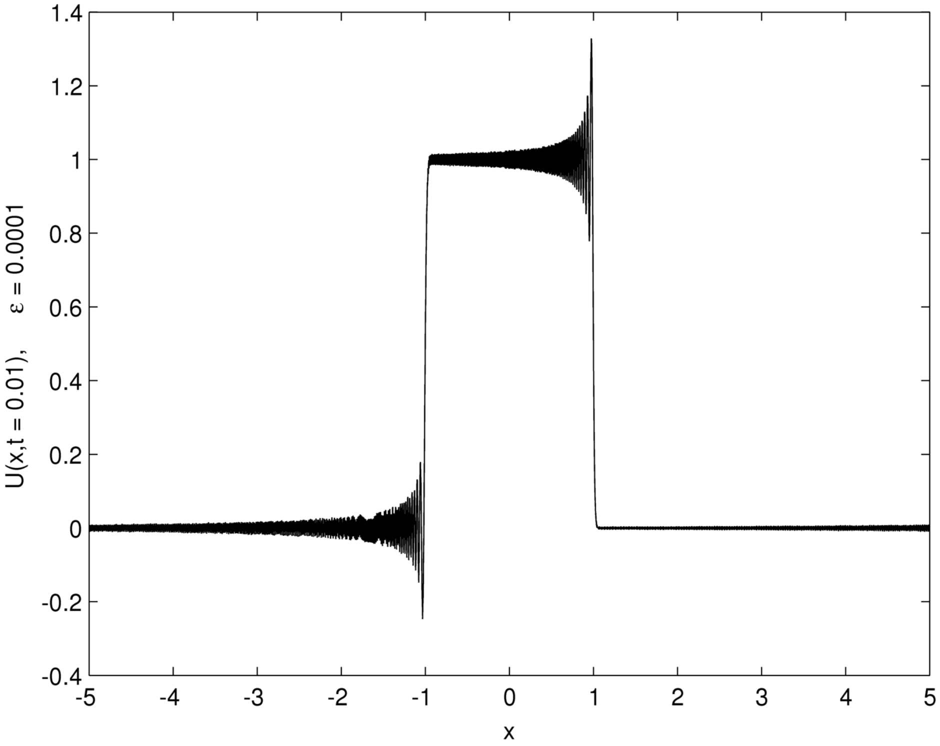


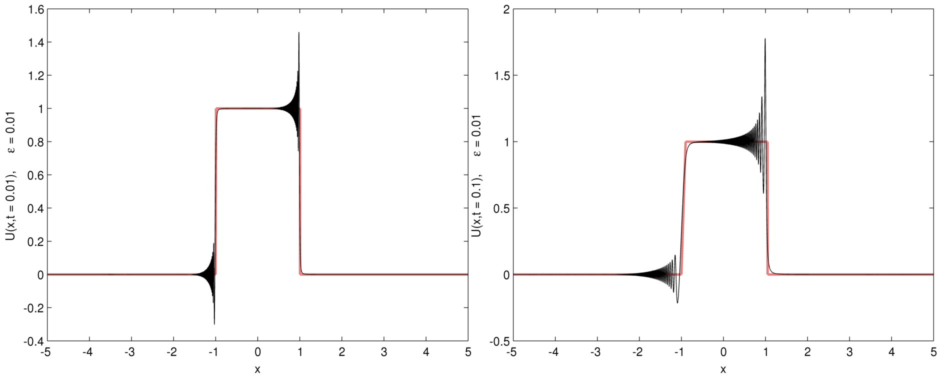
Figure 6.6: Solution to the KdV equation with $\epsilon = 0.0001$, after $t = 0.01$.

We can see in the three Figures 6.3a), 6.4 and 6.6, at time $t = 0.01$, that the amplitude of the waves is much the same, thus it seems independent of ϵ. The amount of waves is increasing with smaller ϵ, i.e. the wavelength is getting shorter. The wavelength is proportional to the size of ϵ. This is in accordance with previous results [LL05a]. According to the theory on the zero dispersion limit of the KdV equation, will the solution after the breaking time be characterised by an "interval of rapid modulated oscillations". The interval is independent of ϵ [GK07].

6.5 Zero Dispersion Limit of the Benjamin-Ono Equation

We now try to verify that the small dispersion term forces the shock and rarefaction to become travelling waves for the Benjamin-Ono equation. This is known for the Benjamin-Ono equation.

We use the same grid size as for the Burgers and KdV equation, $N = 100000$ and $k = 5L/N = 0.00025$. The two plots in Figure 6.7 show the solution with $\epsilon = 0.01$. Both in the shock and in the rarefaction do we get a lot of waves. Compared to the solution to the KdV equation, we have a lot less noise, and a higher density of waves in the shock and to the left of the rarefaction. The amplitude of the first wave in the shock is also slightly larger than in the KdV equation.

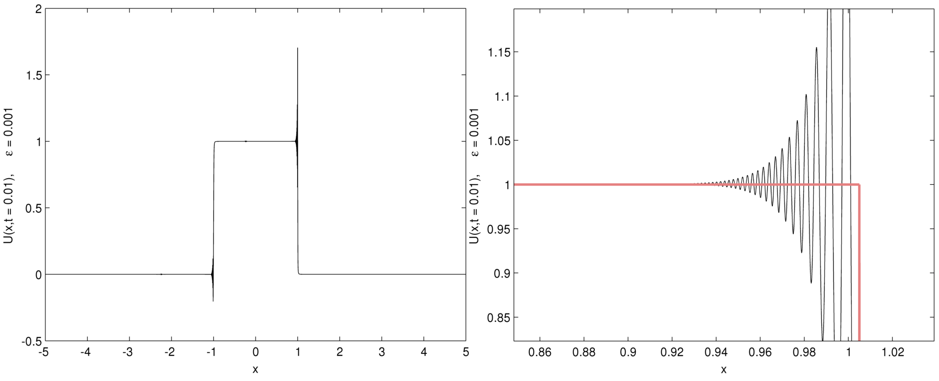


(a) $t = 0.01$

(b) $t = 0.1$

Figure 6.7: Solution to the BO equation with $\epsilon = 0.01$ in black, and the weak solution to the inviscid Burgers equation in red.

For $\epsilon = 0.001$ do we also get waves in the shock and rarefaction. But this time the waves are concentrated in a much smaller region around the critical points. The amplitude of the highest wave is also higher, and the wavelength is shorter. The plots follow in Figure 6.8.

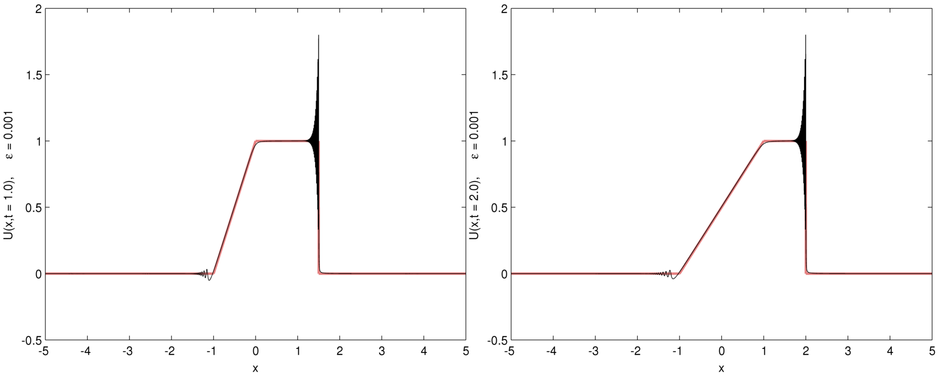


(a)

(b) Plot in a) around the shock.

Figure 6.8: Solution to the BO equation with $\epsilon = 0.001$, after $t = 0.01$.

The solutions after $t = 1$ and $t = 2$ are in Figure 6.9. The solutions are much closer to the weak solution of the inviscid Burgers equation than what the solutions of the KdV equation are. However, we still have large rapid changing waves at the shock, and some smaller waves at the left end of the rarefaction.

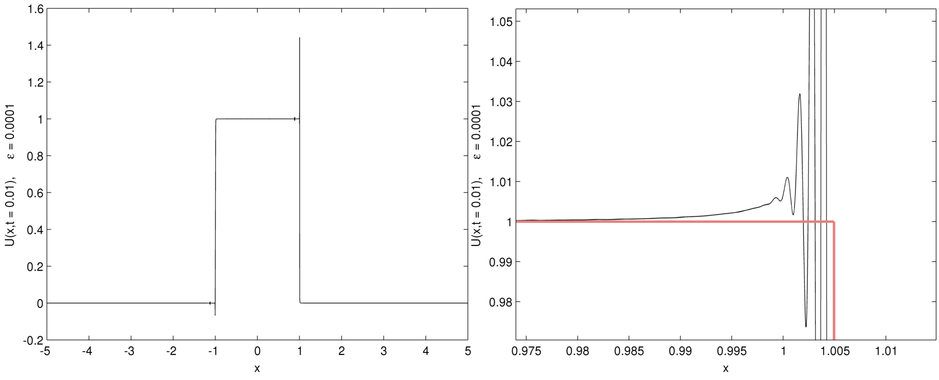


(a) $t = 1$

(b) $t = 2$

Figure 6.9: Solution to the BO equation, with $\epsilon = 0.001$, in black, and the weak solution to the inviscid Burgers equation in red.

Lastly, we plot the solutions to the Benjamin-Ono equation in the case of $\epsilon = 0.0001$. Figure 6.10 shows the plot of the solution at $t = 0.01$. As expected, the waves have even shorter wavelength. This is the same as we see for the KdV equation. Very few waves have broken out, both at the shock, and at the left of the rarefaction.

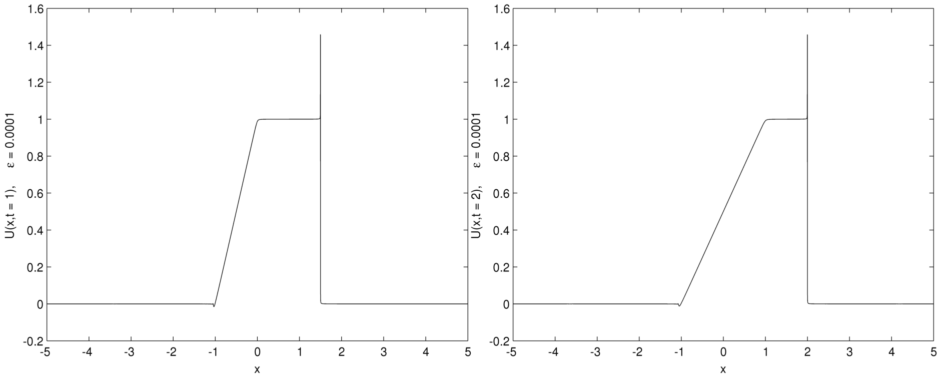


(a)

(b) Plot in a) around the shock.

Figure 6.10: Solution to the BO equation with $\epsilon = 0.0001$, after $t = 0.01$. Weak solution to the inviscid Burgers equation in red.

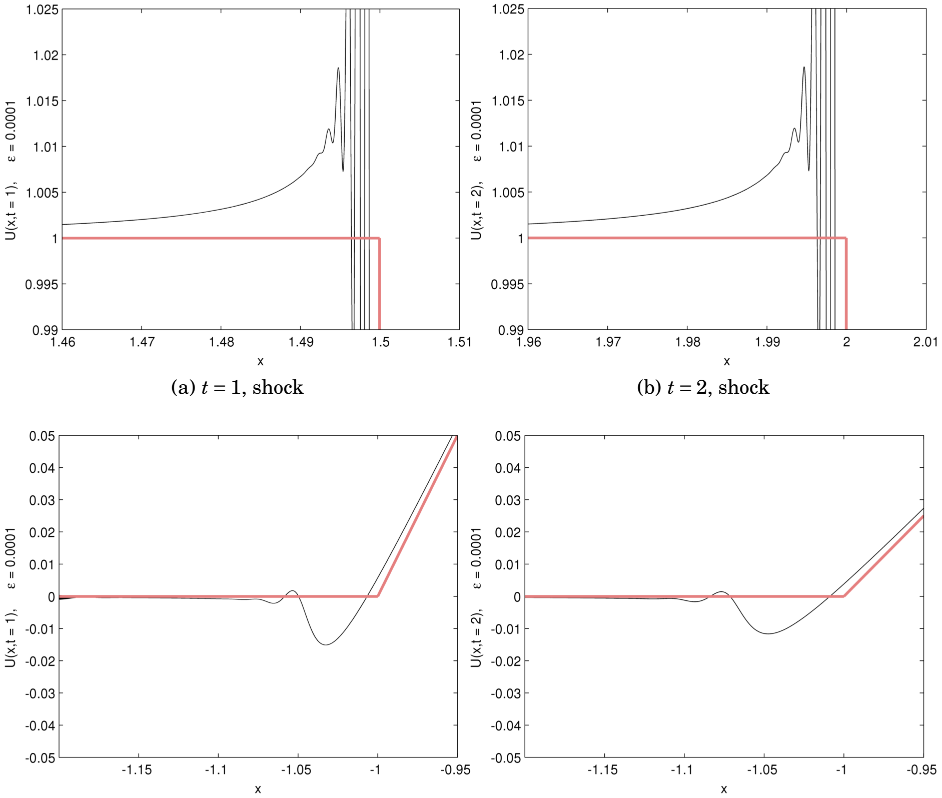
At $t = 1$ and $t = 2$ we have the following solutions in Figure 6.11. We plot the same solutions magnified around the shock and the left of the rarefaction in Figure 6.12.



(a) $t = 1$

(b) $t = 2$

Figure 6.11: Solution to the BO equation, with $\epsilon = 0.0001$, in black.



(a) $t = 1$, shock

(b) $t = 2$, shock

(c) $t = 1$, rarefaction

(d) $t = 2$, rarefaction

Figure 6.12: Solutions to the BO equation, with $\epsilon = 0.0001$, around the shock and to the left of the rarefaction.

Although the solution is much closer to the weak solution than what the solution to the KdV equation is, does the small dispersion term force the shock to become travelling waves also in the BO equation. For smaller ϵ we get smaller wavelength. This is in accordance with works on the zero dispersion limit for the BO equation [MX11, Xu10].

6.6 Comparison to Previous Numerical Results

In this last section, we want to see if we can reproduce some results previously found in the zero dispersion limit for the KdV and BO equation. We will use the Operator Splitting with Taylor Expansion method to obtain our solutions.

We will first look at a result by Grava and Klein [GK07], and try to reproduce one of their solutions to the KdV equation. They solve a differently scaled equation, so we will have to multiply the initial condition by 6. When we compare the solutions, we will downscale the solution by 6 again, so we can compare the plots. They solve the KdV equation on $[-5,5]$, with $\epsilon = 0.001$, and with the following initial condition (scaled to our equation)

$$u(x,0) = u_0(x) = -\frac{6}{\cosh^2(x)}. \tag{6.9}$$

The result of Grava and Klein for different time steps is presented in Figure 6.13.

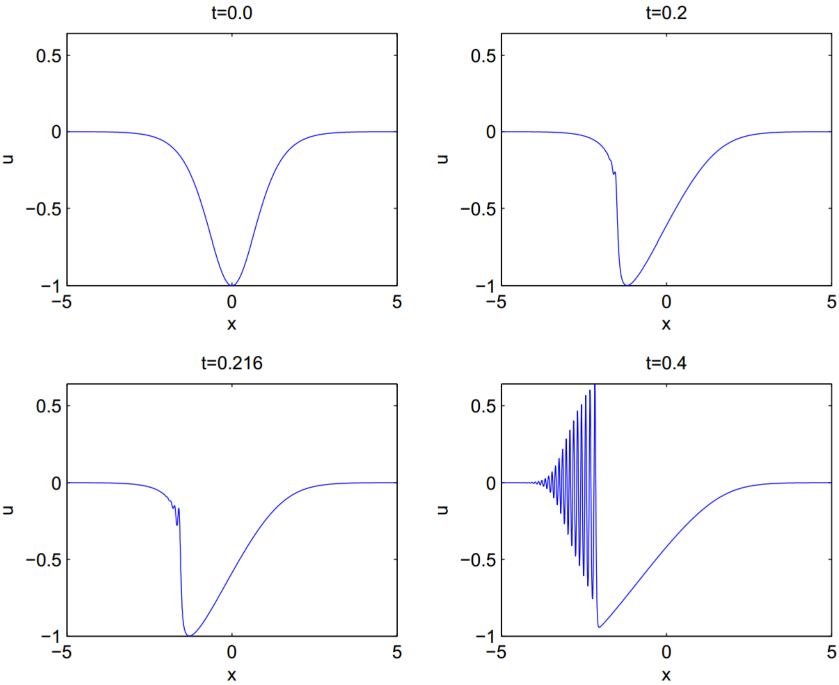
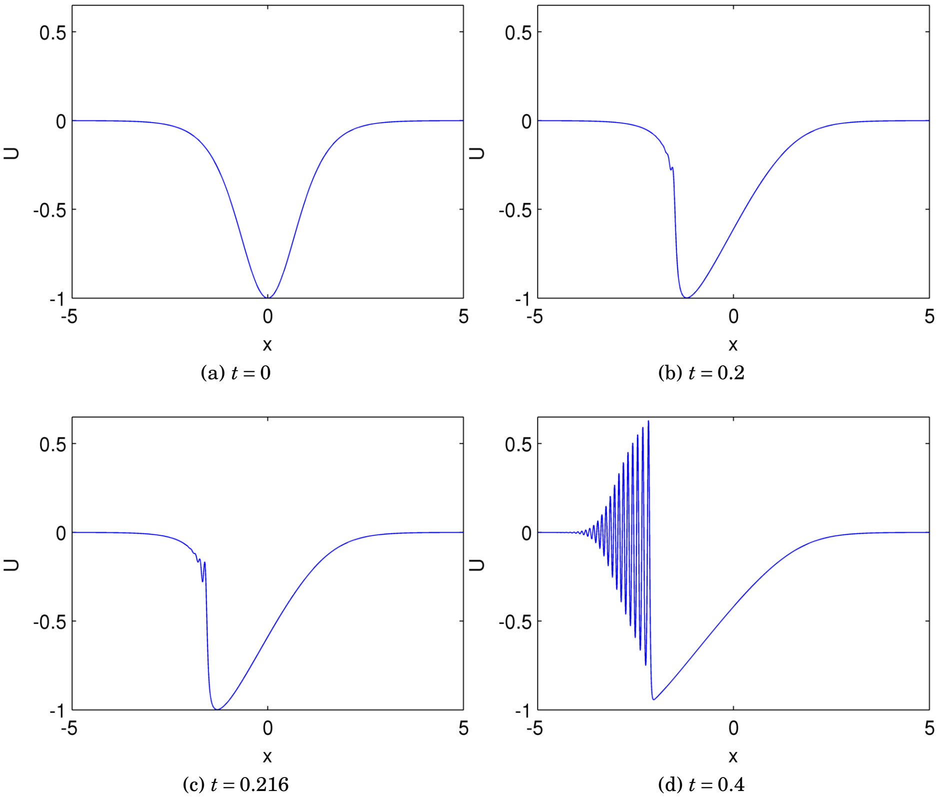


Figure 6.13: The solution found by Grava and Klein with $\epsilon = 0.001$. The plot is taken from [GK07].

Our results for the same times are presented in Figure 6.14. The results were produced with the Operator Splitting With Lax-Friedrichs method, with $N = 100000$ and $k = 5L/N = 0.00025$.



(a) $t = 0$

(b) $t = 0.2$

(c) $t = 0.216$

(d) $t = 0.4$

Figure 6.14: The solution found by the Operator Splitting with Lax-Friedrichs method with $\epsilon = 0.001$ and $N = 100000$ and $k = 5L/N$.

Qualitatively, we see the same results as obtained by Grava and Klein. It is hard to tell, but it also looks like we have the same amount of waves.

We now consider a solution to the BO equation found by Miller and Xu [MX11]. They solve a differently scaled equation, so we have to multiply the initial condition by 2. We will downscale the solution by 2 in the plots, so we can easily compare them. The initial condition is (scaled to our equation)

$$u(x,0) = u_0(x) = \frac{4}{1+x^2}, \tag{6.10}$$

and they solve the equation with $\epsilon = 0.04$ and $\epsilon = 0.02$. The solutions by Miller and Xu are found in Figure 6.15.

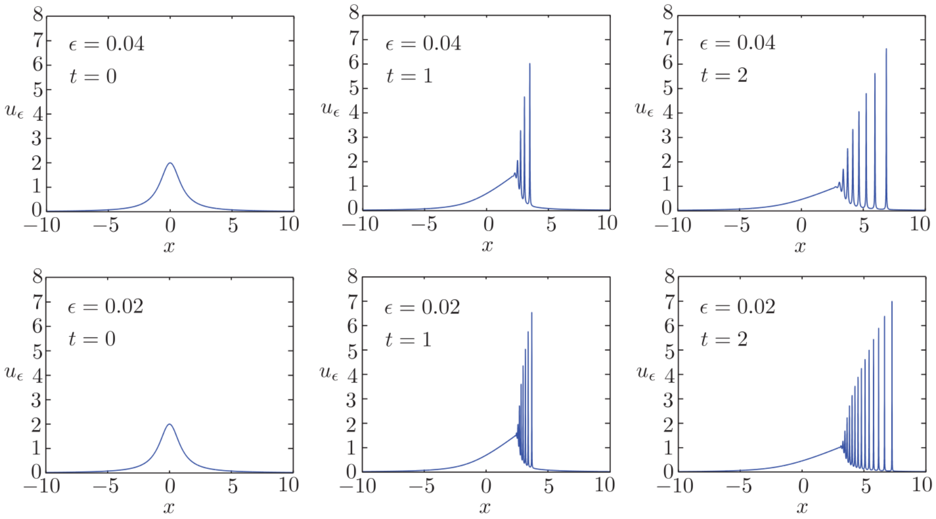
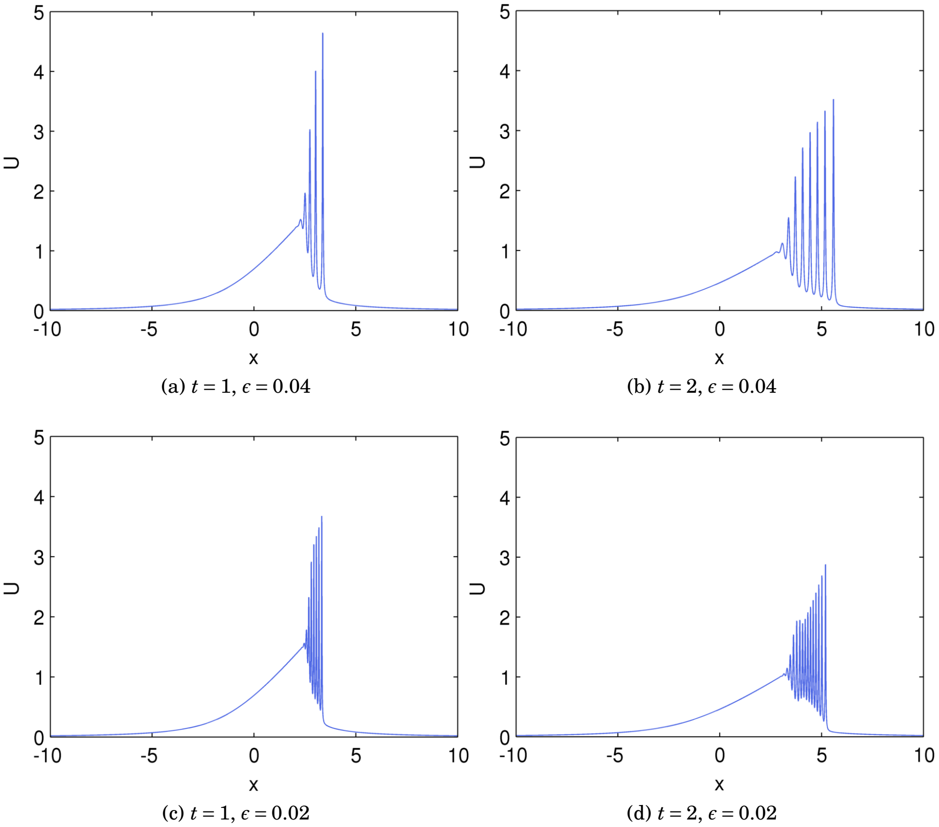


Figure 6.15: The solution found by Miller and Xu with $\epsilon = 0.04$ and $\epsilon = 0.02$. The plot is taken from [MX11].

Our results for the same times are presented in Figure 6.16. The results were produced with the Operator Splitting With Lax-Friedrichs method, with $N = 100000$ and $k = 5L/N = 0.00025$.



(a) $t = 1$, $\epsilon = 0.04$

(b) $t = 2$, $\epsilon = 0.04$

(c) $t = 1$, $\epsilon = 0.02$

(d) $t = 2$, $\epsilon = 0.02$

Figure 6.16: The solution found by the Operator Splitting with Lax-Friedrichs method with $N = 100000$ and $k = 5L/N$.

The results are not as good as for the KdV equation, and the waves are much lower than in the solution by Miller and Xu (note the axes). Although it is hard to tell from these plots, we do have exactly the same number of peaks. And we see the same behaviour as before, that the wavelength decreases with ϵ.

Chapter 7

Concluding Remarks

We have put four different numerical methods for solving the Benjamin-Ono equation to the test. In all the test problems in Chapter 5, the Semi-Implicit method by Chan and Kerkhoven proved to be the fastest in reaching the prescribed accuracies. The method was originally intended for the KdV equation, but has proved to work well also for the BO equation. For the 1-periodic test problem, the method of Thomée and Vasudeva Murthy was the second fastest to reach the accuracy constraint. However, as the difficulty of the test problems increased, the Operator Splitting with Taylor Expansion method outperformed the method of Thomée and Vasudeva Murthy. The two Fourier pseudospectral methods require far less steps in space than the other two methods. So, even with very fine time steps these two methods will perform very well. Whereas implicit time stepping often is absolutely necessary in finite difference methods, explicit time stepping can be acceptable in pseudospectral methods. So, the FPS methods were able to reach high accuracies in short amounts of time. The Operator Splitting with Lax-Friedrichs method was the slowest method in all test problems. It did reach the accuracy constraints in the first and third test problems, but the Lax-Friedrichs step requires too many steps in space to compete with the other methods.

In Chapter 6 we investigated the zero dispersion limit for the KdV and BO equation. We tried to verify that the small dispersion term, both in the KdV and BO equation, will force the shock to become travelling waves. The solutions did exhibit this behaviour, but the problem is difficult to solve numerically, and it is hard to determine the quality of the results. Qualitatively, we got the same results when we compared our solutions to previous ones. We also saw that the travelling waves got shorter wavelength as we decreased ϵ, which is in accordance with former results.

We discovered that the Fourier pseudospectral methods implemented could not cope very well with the discontinuities, and the solutions blew up quickly. Since the Lax-Friedrichs step is dissipative, and smooths out discontinuities, we chose to use the Operator Splitting with Lax-Friedrichs method to solve the zero dispersion limit problems. We did, however, experience a lot of numerical noise as small, high frequency oscillations, and the calculations took a considerable amount of time. For the same step sizes and same ϵ, we got less noise in the solutions to the BO equation than to the KdV equation. FPS methods have been used for zero dispersion limit problems previously. Grava and Klein [GK07] implemented a FPS method based on the method proposed by Trefethen [Tre00] for the KdV equation, in which a forth order Runge-Kutta (RK4) time stepping scheme was used. There is obviously a lot more work to be done on the zero dispersion limit for these equations, both theoretically and numerically.

Bibliography

[Ben67] T. Brooke Benjamin, *Internal waves of permanent form in fluids of great depth*, Journal of Fluid Mechanics **29** (1967), 559–592.

[BN71] P.L. Butzer and R.J. Nessel, *Fourier analysis and approximation*, Pure and Applied Mathematics, no. v. 1, Elsevier Science, 1971.

[Boy01] J.P. Boyd, *Chebyshev and Fourier Spectral Methods: Second Revised Edition*, Dover Books on Mathematics Series, Dover Publ., 2001.

[CK85] Tony F. Chan and Tom Kerkhoven, *Fourier Methods with Extended Stability Intervals for the Korteweg-de Vries Equation*, SIAM Journal on Numerical Analysis **22** (1985), no. 3, pp. 441–454 (English).

[DVZ97] P. Deift, S. Venakides, and X. Zhou, *New results in small dispersion KdV by an extension of the steepest descent method for Riemann-Hilbert problems*, International Mathematics Research Notices **1997** (1997), no. 6, 285–299.

[For98] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 1998.

[GK07] Tamara Grava and Christian Klein, *Numerical solution of the small dispersion limit of Korteweg-de Vries and Whitham equations*, Communications on Pure and Applied Mathematics **60** (2007), no. 11, 1623–1664.

[Hil73] E. Hille, *Analytic function theory. 1*, Analytic Function Theory, Chelsea, 1973.

[HKLR10] H. Holden, K. H. Karlsen, K. A. Lie, and N. H. Risebro, *Splitting Methods for Partial Differential Equations with Rough Solutions: Analysis* *and Matlab Programs*, EMS Series of Lectures in Mathematics, Amer Mathematical Society, 2010.

[KdV95] D. J. Korteweg and G. de Vries, *XLI. On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves*, Philosophical Magazine Series 5 **39** (1895), no. 240, 422–443.

[LL05a] Peter D. Lax and C. David Levermore, *The Small Dispersion Limit of the Korteweg-de Vries Equation. I*, Selected Papers Volume I (Peter Sarnak and Andrew Majda, eds.), Springer New York, 2005, pp. 463–500.

[LL05b] ______, *The Small Dispersion Limit of the Korteweg-de Vries Equation. II*, Selected Papers Volume I (Peter Sarnak and Andrew Majda, eds.), Springer New York, 2005, pp. 501–523.

[LL05c] ______, *The Small Dispersion Limit of the Korteweg-deVries Equation. III*, Selected Papers Volume I (Peter Sarnak and Andrew Majda, eds.), Springer New York, 2005, pp. 524–544.

[Mat98] Yoshimasa Matsuno, *The Small Dispersion Limit of the Benjamin-Ono Equation and the Evolution of a Step Initial Condition*, Journal of the Physical Society of Japan **67** (1998), no. 6, 1814–1817.

[MX11] Peter D. Miller and Zhengjie Xu, *On the zero-dispersion limit of the Benjamin-Ono Cauchy problem for positive initial data*, Communications on Pure and Applied Mathematics **64** (2011), no. 2, 205–270.

[NS89] F. Z. Nouri and D. M. Sloan, *A comparison of Fourier pseudospectral methods for the solution of the Korteweg-de Vries equation*, Journal of Computational Physics **83** (1989), no. 2, 324–344.

[Ono75] Hiroaki Ono, *Algebraic Solitary Waves in Stratified Fluids*, Journal of the Physical Society of Japan **39** (1975), no. 4, 1082–1091.

[PS02] Anne Porter and Noel F. Smyth, *Modelling the Morning Glory of the Gulf of Carpentaria*, Journal of Fluid Mechanics **454** (2002), 1–20.

[SI79] Junkichi Satsuma and Yūji Ishimori, *Periodic Wave and Rational Soliton Solutions of the Benjamin-Ono Equation*, Journal of the Physical Society of Japan **46** (1979), no. 2, 681–687.

[Tre00] L.N. Trefethen, *Spectral Methods in MATLAB*, Software, Environments, Tools, Society for Industrial and Applied Mathematics, 2000.

[TV98] V. Thomée and A. Vasudeva Murthy, *A numerical method for the Benjamin-Ono equation*, BIT Numerical Mathematics **38** (1998), 597–611, 10.1007/BF02510262.

[Xu10] Zhengjie Xu, *Asymptotic Analysis and Numerical Analysis of the Benjamin-Ono Equation*, Ph.D. thesis, The University of Michigan, 2010.

[ZK65] N. J. Zabusky and M. D. Kruskal, *Interaction of "Solitons" in a Collisionless Plasma and the Recurrence of Initial States*, Phys. Rev. Lett. **15** (1965), 240–243.