Nonlocal Models for Undular Bores

Marius Nevland

Master Thesis in Applied and Computational Mathematics

Department of Mathematics
University of Bergen

June 1, 2022

Abstract

In this thesis we have studied various equations that are potential models of the *undular bore*, which is a specific kind of water wave, usually caused by tidal forces, that can be observed in various rivers around the world. We start by considering the well-known Korteweg-de Vries equation, but with an additional double-derivative term that models dissipation. It has been proven that this particular equation admits traveling-wave solutions in the shape of undular bores. We solve the equation numerically by a spectral method, and numerical approximations of these undular bores are obtained.

Next, we attempt to generalize this equation, inspired by how the Korteweg-de Vries equation has been generalized into a nonlocal equation by Whitham [32]. Our numerical experiments indicate that this generalized equation also admits undular-bore solutions.

Finally, we briefly consider an equation with a nonlocal term that models dissipation in a small boundary layer near the bottom of the channel. This equation might be the most realistic model of the undular bore considered here. However, since this equation did not produce as interesting results as the previous equation, we have not given it too much focus.

Acknowledgements

I would like to thank my supervisor, Henrik Kalisch, for providing me with a very interesting project for my thesis, and for all his help and guidance along the way.

I would also like to thank my family for all of their support and encouragement. Finally, thanks to all of the wonderful people I have met in Bergen over the years, who have made my time as a student here a truly memorable experience.

Contents

Chapter 1

Introduction

The central object of study in this thesis is the *undular bore*. This is a special case of a *tidal bore*, which are waves caused by tidal forces that propagate upstream (in other words, against the usual flow direction). Tidal bores occur regularly in various rivers around the world, such as the Severn river in England, the Dordogne river in France and the Qiantang river in China [1].

The experiments of Favre [10] showed that tidal bores come in two characteristic forms. If the ratio $\frac{a}{h_0}$ between the undisturbed water depth and the amplitude of the incoming wave profile is greater than 0.28, then the bore develops a sharp, turbulent increase in elevation at the bore front, which is called a *hydraulic jump*. However, if the ratio is less than 0.28, the increase in elevation is more gradual, and oscillations behind the bore front start to develop. This is what is called the *undular bore*.

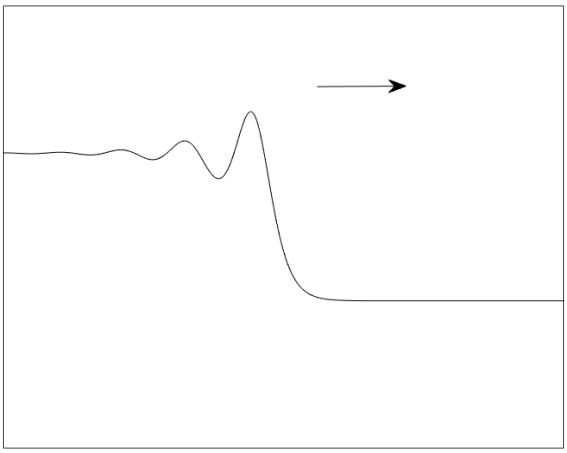


Figure 1.1: Sketch of a right-moving undular bore. This plot was made using the numerical scheme in section 6.2.

It is generally accepted that for hydraulic jumps, energy is "lost" due to the turbulence at the bore front [13, 17]. The turbulence initiates processes that convert mechanical energy into thermal energy. By the second law of thermodynamics, there is less capacity to do mechanical work after this conversion, and hence we say that energy is lost, or *dissipated*.

Energy loss occurs at the front of an undular bore as well, and it is believed that most of this energy is carried away by the wave train behind the front, rather than dissipated; this was proposed by Lemoine [18]. However, Benjamin and Lighthill [2] argued that a small amount of energy must still be dissipated in the undular bore. Moreover, they argued that the train of waves behind the front are approximately cnoidal waves, i.e., approximate solutions to the well-known Korteweg-de Vries (KdV) equation:

$$\eta_t + c_0\eta_x + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + \frac{1}{6}c_0h_0^2\eta_{xxx} = 0 \tag{1.1}$$

Calculations by Sturtevant [29] further expands on the theory of the undular bore. Sturtevant uses the results of Favre's experiments, together with cnoidal wave theory, to show that the dissipation in the undular bore is not due to turbulence, but rather due to frictional effects in a thin boundary layer near the bottom of the channel.

Hence, one would expect that a realistic model of the undular bore would involve an equation similar to the Korteweg-de Vries equation, but with an additional term modelling the dissipation in the boundary layer. Several authors have attempted to derive exactly this type of equation, and they all arrived at very similar equations [6, 13, 14].

In this thesis we will investigate several KdV-type equations with an additional dissipative term, and search for numerical solutions in the shape of undular bores. We will start quite simple, and consider the KdV equation with an additional double-derivative term that models energy loss. We will then attempt to generalize this equation in the same way the KdV equation was generalized into a nonlocal equation by Whitham in [31]. This generalized equation turned out to produce some mathematically interesting results, so we have chosen to focus the most on this equation.

Finally, we will attempt to generalize the boundary-layer equation from [14], again by using the theory of Whitham, and perform a numerical experiment on the generalized equation.

1.1 Outline of the Thesis

The structure of this thesis is as follows:

In chapter 2, some basic theory of fluid mechanics will be reviewed. This includes derivations of the continuity equation, the Navier-Stokes equations and a Bernoulli equation.

In chapter 3 we derive the water-wave problem for gravity waves, which are a set of equations that describe the propagation of water waves under the influence of gravity. The solution of the linearized problem will be derived, and we will arrive at the well-known *dispersion relation* for water waves.

In chapter 4 we consider equations for water waves that incorporate nonlinear effects. This includes the famous Korteweg-de Vries (KdV) equation, which can be derived from the water-wave problem with some additional assumptions. We also present a nonlocal equation called the Whitham equation, which can be seen as a generalization of the KdV equation. Finally, we consider how dissipative effects can be incorporated into these equations.

In chapter 5, we present the Fourier collocation method for numerically solving partial differential equations (which in this thesis will also be used to solve nonlocal integro-differential equations).

In chapter 6, we present our numerical experiments. We begin by numerically solving the KdV equation. Since this equation has known exact solutions, it can be used to verify that our numerical method works as intended. We then proceed by numerically solving the three dissipative equations that were mentioned in the introduction, and we do comparisons between the numerical solutions obtained in the three cases.

Note that chapters 2 through 5 are generally reviews of background theory that is needed for the experiments in chapter 6 (possibly with the exception of section 4.4, where a new equation is presented).

1.2 Notation and Abbreviations

The following abbreviations will be used throughout this thesis:

ODE - Ordinary Differential Equation

PDE - Partial Differential Equation

KdV Equation - Korteweg-de Vries Equation

DFT - Discrete Fourier Transform

IDFT - Inverse Discrete Fourier Transform

DCT - Discrete Cosine Transform

IDCT - Inverse Discrete Cosine Transform

We denote by $L^1(a,b)$ the space of all functions that are absolutely integrable (in the Lebesgue sense) over $[a,b] \subset \mathbb{R}$, and by $L^2(a,b)$ the space of functions that are square-integrable over $[a,b]$:

$$L^1(a,b) = \left\{ u : \int_a^b |u(x)|dx < \infty \right\} \tag{1.2}$$

$$L^2(a,b) = \left\{ u : \int_a^b |u(x)|^2 dx < \infty \right\} \tag{1.3}$$

In particular, $L^1(\mathbb{R})$ is the set of functions that are absolutely integrable over the whole real line, and $L^2(\mathbb{R})$ the set of functions that are square-integrable over the whole real line.

We denote by $L^\infty(\mathbb{R})$ the space of functions that are bounded almost everywhere:

$$L^\infty(a,b) = \{u : |u(x)| \leq M \ \text{ almost everywhere}\} \tag{1.4}$$

However, most of the time when necessary, we will simply assume that a function is bounded everywhere, i.e., that $|u(x)| \leq M$ for all $x \in \mathbb{R}$.

The Fourier transform of a function f will be denoted by $\hat{f}$ or $\mathcal{F}(f)$, and the inverse Fourier transform will be denoted by $\mathcal{F}^{-1}(f)$. Furthermore, the Fourier and inverse Fourier transform are used with the following conventions:

$$\mathcal{F}(f) = \hat{f}(k) = \int_{-\infty}^{\infty} f(x) e^{-ikx} dx \tag{1.5}$$

$$\mathcal{F}^{-1}(f) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(k) e^{ikx} dk \tag{1.6}$$

The partial derivative of a function f with respesct to some variable x is written as $\frac{\partial f}{\partial x}$ or f_x.

Chapter 2

Basic Theory of Fluid Mechanics

In this chapter we will review some of the most general equations of fluid mechanics. We start by considering an important quantity in fluid mechanics, called the *material derivative*.

2.1 Material Derivative

Fluid flows may be described in terms of a *velocity field* $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, which gives the velocity of the fluid at some point $\mathbf{x}$ in three-dimensional space, at time t. The time rate of change of some quantity F, subject to this velocity field, is called the material derivative, and is given by

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F \tag{2.1}$$

This formula shows that the time rate of change of F occurs in two different ways. The first is the local rate of change of F at some point $\mathbf{x}$, as given by the first term of (2.1). The second way is the rate of change of F that happens due to a particle moving through the velocity field, which is what the second term of (2.1) models.

The left-hand side of (2.1) can be interpreted as the derivative one would compute in a frame of reference that is following a fluid particle.

2.2 The Governing Equations of Fluid Mechanics

In this section we will derive the continuity equation and the Navier-Stokes equations, which are the equations that govern how a fluid moves. They are based on the familiar physical

laws of conservation of mass and momentum. These derivations can be found, with more details, in chapter 4 of [17].

2.2.1 Conservation of Mass

We define a *material volume* $V(t)$ as a region that always contains the same fluid particles (the region will move and deform in order to fulfill this). The total mass within this region must be constant in time. In other words, the time derivative of the total mass must be zero:

$$\frac{d}{dt}\int_{V(t)} \rho dV = 0 \tag{2.2}$$

Where ρ is the density of the fluid (which at the moment may vary in space and time). The left-hand side can be rewritten using the *Reynolds transport theorem*, which is a multi-dimensional generalization of Leibniz' rule for differentiating under the integral sign. It states that:

$$\frac{d}{dt}\int_{V(t)} \mathbf{F} dV = \int_{V(t)} \frac{\partial \mathbf{F}}{\partial t} dV + \int_{A(t)} \mathbf{F}(\mathbf{b}\cdot\mathbf{n}) dA \tag{2.3}$$

Where $A(t)$ is the surface of $V(t)$, and $\mathbf{b}$ is the surface velocity. For a material volume, the velocity of the surface is just the local flow velocity, that is, $\mathbf{b} = \mathbf{u}$. Applying the Reynolds transport theorem on (2.2), we get

$$\int_{V(t)} \frac{\partial \rho}{\partial t} dV + \int_{A(t)} \rho(\mathbf{u}\cdot\mathbf{n}) dA = 0 \tag{2.4}$$

Physically, this states that density changes within $V(t)$ are due to the movement of the boundary of $V(t)$. We can use the divergence theorem to convert the surface integral into a volume integral, and we obtain

$$\int_{V(t)} \left[\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u})\right] dV = 0 \tag{2.5}$$

Since this must hold for an arbitrary material volume, the only way for this to be satisfied is if the integrand is zero at every point in space. Thus, we obtain what is called the *continuity equation*:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0 \tag{2.6}$$

It can alternatively be written using the material derivative:

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \nabla \cdot \mathbf{u} = 0 \tag{2.7}$$

Now, a fluid is said to be *incompressible* if the density does not change while following a fluid particle, that is, $\frac{D\rho}{Dt} = 0$. In this case, we see from (2.7) that the continuity equation can be simplified to

$$\nabla \cdot \mathbf{u} = 0 \tag{2.8}$$

Note that a fluid whose density is constant everywhere is a special case of an incompressible fluid.

2.2.2 Conservation of Momentum

Newton's second law states that the sum of all forces acting on a body is equal to the time rate of change of the momentum of the body. We would like to apply this law to a material volume of a fluid. In that case, there are two relevant types of forces: *body forces* and *surface forces*. Body forces are long-range forces, and we will consider gravity as the only body force. Surface forces are intermolecular forces between the surface of the material volume and the fluid outside the volume. Note that by Newton's third law, all internal forces within the material volume cancel out, and they are therefore not considered.

Newton's second law applied on the material volume is then

$$\frac{d}{dt}\int_{V(t)} \rho\mathbf{u}dV = \int_{V(t)} \rho\mathbf{g}dV + \int_{A(t)} \mathbf{f}dA \tag{2.9}$$

Where $\mathbf{g}$ is the body force per unit mass (which in this case is just gravity), and $\mathbf{f}$ is the surface force per unit area. Note that this is a system of three equations, one for each spatial coordinate.

Applying Reynolds transport theorem on the left-hand side, we obtain

$$\int_{V(t)} \frac{\partial}{\partial t}(\rho\mathbf{u})dV + \int_{A(t)} \rho\mathbf{u}(\mathbf{u}\cdot\mathbf{n})dA = \int_{V(t)} \rho\mathbf{g}dV + \int_{A(t)} \mathbf{f}dA \tag{2.10}$$

We then use the divergence theorem to convert the surface integral on the left-hand side into a volume integral:

$$\int_{V(t)} \frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) dV = \int_{V(t)} \rho \mathbf{g} dV + \int_{A(t)} \mathbf{f} dA \tag{2.11}$$

Where $\otimes$ denotes the outer product of two vectors. Finally, we use the divergence theorem again to convert the surface integral on the right-hand side into a volume integral:

$$\int_{V(t)} \left[\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \rho \mathbf{g} - \nabla \cdot \mathbf{T} \right] dV = 0 \tag{2.12}$$

Where $\mathbf{T}$ is the second-order stress tensor (by definition, $\mathbf{f} = \mathbf{T} \cdot \mathbf{n}$). As before, these equations are only true if the integrands vanish at every point in space, yielding the system of differential equations:

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} + \nabla \cdot \mathbf{T} \tag{2.13}$$

The left-hand side of this can be rewritten by using the product rule on both terms:

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} \right) + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} \tag{2.14}$$

Where the continuity equation (2.6) has been used for the final equality. Thus, equation (2.13) can be rewritten as

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \rho \mathbf{g} + \nabla \cdot \mathbf{T} \tag{2.15}$$

At the moment, equation (2.15) together with (2.6) contain more unknowns than equations, and we need to specify the form of the stress tensor in order to close the system. It can be shown that for incompressible flow, the stress tensor takes the form (see chapter 4 of [17] for a derivation):

$$\mathbf{T} = -p\mathbf{I} + \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \tag{2.16}$$

Where $\mathbf{I}$ is the identity matrix, p is the pressure and μ is the dynamic viscosity. The first term of (2.16) are the normal stresses present in any fluid at rest. For moving fluids, the action of viscosity produces additional stresses, which is what the second term of (2.16) models.

By substituting (2.16) into (2.15), and using the incompressible continuity equation (2.8), together with the vector calculus identities

$$\nabla \cdot (\nabla \mathbf{u}) = \nabla^2 \mathbf{u} \tag{2.17}$$

$$\nabla \cdot (\nabla \mathbf{u})^T = \nabla(\nabla \cdot \mathbf{u}) \tag{2.18}$$

Equation (2.15) can be written as

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \tag{2.19}$$

These are commonly called the *incompressible Navier-Stokes equations*. Together with the incompressible continuity equation (2.8), they constitute a set of four equations with five unknowns (ρ,p and the three components of $\mathbf{u}$). If, in addition, we assume either that ρ is a known constant, or that ρ is a known function of p (called *barotropic flow*), we have a complete system of equations.

If we in addition assume that the fluid is inviscid, i.e., that $\mu = 0$, then the Navier-Stokes equations simplify to the *Euler equations*:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \rho \mathbf{g} \tag{2.20}$$

2.3 Bernoulli Equations

There are several equations that can be derived from the Navier-Stokes equations with some additional assumptions on the flow. All of them are closely related and are collectively called *Bernoulli equations*. In this section, we will derive a form of the Bernoulli equation that will be relevant in chapter 3, where gravity waves are studied. The derivation can be found in chapter 4 of [17].

We consider inviscid flow, so the relevant equations are the Euler equations. It can be shown, using vector calculus identities, that the Euler equations can be rewritten in the form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \left(\frac{1}{2} |\mathbf{u}|^2 \right) + \boldsymbol{\omega} \times \mathbf{u} = -\frac{1}{\rho} \nabla p - g\mathbf{j} \tag{2.21}$$

Where $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity, which can be interpreted as the tendency of fluid particles in an infinitesimally small region around a point to start rotating. If we assume that the fluid is barotropic, then the pressure gradient term can be rewritten as an integral:

$$\nabla\left(\int_{p_0}^{p}\frac{dp'}{\rho(p')}\right)=\frac{\partial}{\partial p}\left(\int_{p_0}^{p}\frac{dp'}{\rho(p')}\right)\nabla p=\frac{1}{\rho}\nabla p \tag{2.22}$$

Here p_0 is an arbitrary reference pressure. The first equality is due to the chain rule, as the integral is a function of p, which is in turn a function of space. The second equality comes from Leibniz' rule for differentiation under the integral sign. Equation (2.21) can now be rewritten as

$$\frac{\partial\mathbf{u}}{\partial t}+\nabla\left(\frac{1}{2}|\mathbf{u}|^2+\int_{p_0}^{p}\frac{dp'}{\rho(p')}+gz\right)=\mathbf{u}\times\boldsymbol{\omega} \tag{2.23}$$

Next, we assume that the fluid is *irrotational*[1], meaning that $\boldsymbol{\omega}=\nabla\times\mathbf{u}=0$. Irrotationality implies that there exists a function ϕ, called the *velocity potential*, such that

$$\mathbf{u}=\nabla\phi \tag{2.24}$$

With the assumption of irrotationality, equation (2.23) becomes

$$\nabla\left(\frac{\partial\phi}{\partial t}+\frac{1}{2}|\nabla\phi|^2+\int_{p_0}^{p}\frac{dp'}{\rho(p')}+gz\right)=0 \tag{2.25}$$

Integrating in space, we obtain

$$\frac{\partial\phi}{\partial t}+\frac{1}{2}|\nabla\phi|^2+\int_{p_0}^{p}\frac{dp'}{\rho(p')}+gz=B(t) \tag{2.26}$$

Where $B(t)$ is a function of integration. This function can be removed by redfining ϕ, for instance $\phi\to\phi+\int_{t_0}^{t}B(t')dt'$. Note that this new potential does not affect the velocity field, as $\nabla\phi$ stays the same. Finally, we get

$$\frac{\partial\phi}{\partial t}+\frac{1}{2}|\nabla\phi|^2+\int_{p_0}^{p}\frac{dp'}{\rho(p')}+gz=0 \tag{2.27}$$

This is the form of the Bernoulli equation that will be used in the study of gravity waves.

[1]This claim of irrotationality will be justified in chapter 3.

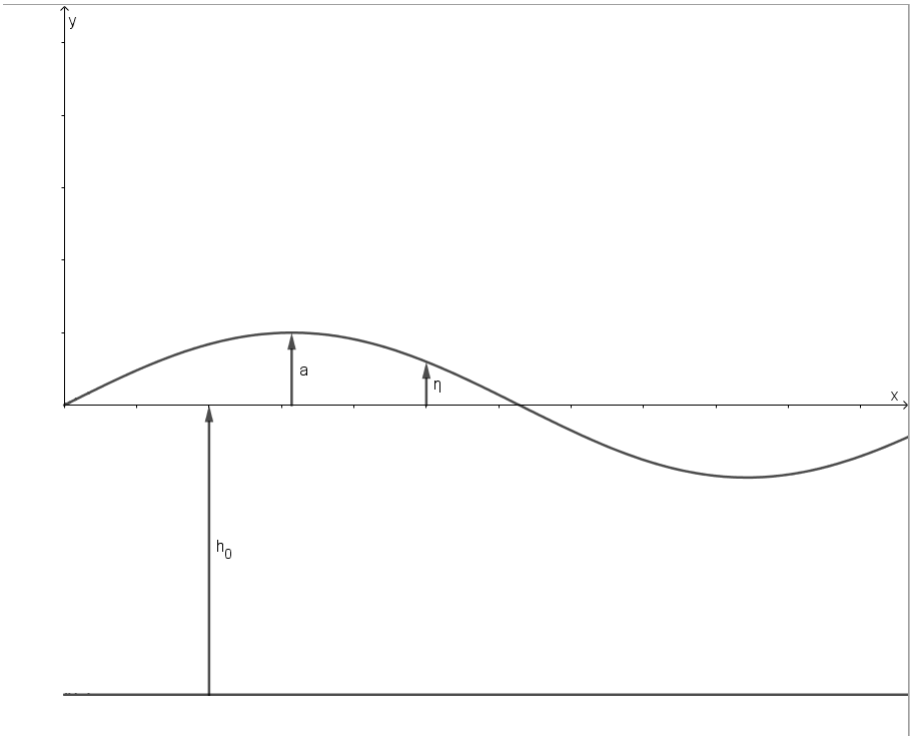
Chapter 3

Gravity Waves

In this chapter we derive equations for gravity waves, that is, waves where the restoring force is gravity. These derivations can be found in chapter 8 of [17], as well as chapter 13 of [31].

3.1 The Water-Wave Problem

We consider the familiar case of surface gravity waves on an air-water interface, or just simply *water waves*. We restrict ourselves to two spatial dimensions, and we use an x-y coordinate system. Let $y = 0$ be the location of the undisturbed surface, and let $\eta(x, t)$ denote the disturbance of the surface at position x and time t. Finally, let a denote the amplitude of the wave and h_0 the distance from the channel bottom to the undisturbed surface (we assume this distance to be constant). The density ρ is assumed to be constant, which is a good approximation as long as there are not any dramatic changes in temperature and pressure. The setup is sketched below.



Since we assume constant density, the fluid is incompressible. We will also assume irrotationality, and justify this claim in section 3.1.1. This allows us to combine equations (2.8) and (2.24), and as a result the governing equation is the *Laplace equation*:

$$\phi_{xx} + \phi_{yy} = 0 \tag{3.1}$$

Next, we formulate the boundary conditions for this problem. The first condition is that no fluid can flow through the bottom floor. In other words, the velocity in the y-direction must be zero there. Thus, by equation (2.24), we have:

$$\phi_y = 0 \quad on \quad y = -h_0 \tag{3.2}$$

At the surface we have a *kinematic boundary condition*. The surface is defined by the fact that fluid particles do not cross it. Therefore, we must require that the fluid particles always follow the surface. Mathematically, this is equivalent to requiring the material derivative of the surface to be zero. By writing the equation for the surface as $f(x, y, t) = y - \eta(x, t) = 0$, and using equation (2.24), we get:

$$\frac{Df}{Dt} = -\eta_x \phi_x + \phi_y - \eta_t = 0 \tag{3.3}$$

From which we obtain the condition:

$$\phi_y = \eta_t + \eta_x \phi_x \quad on \quad y = \eta \tag{3.4}$$

The final boundary condition is a *dynamic surface condition*. We require the pressure of the air just above the surface to be equal to the pressure of the water just below the surface. The Bernoulli equation (2.27) can be used here. Taking p_0 to be the constant value of the undisturbed air, and requiring that $p = p_0$, the integral in (2.27) becomes zero and we get:

$$\phi_t + \frac{1}{2}(\phi_x^2 + \phi_y^2) + g\eta = 0 \quad on \quad y = \eta \tag{3.5}$$

The full problem with all boundary conditions is then:

$$\phi_{xx} + \phi_{yy} = 0 \tag{3.6a}$$
$$\phi_y = 0 \quad on \quad y = -h_0 \tag{3.6b}$$
$$\phi_y = \eta_t + \eta_x \phi_x \quad on \quad y = \eta \tag{3.6c}$$
$$\phi_t + \frac{1}{2}(\phi_x^2 + \phi_y^2) + g\eta = 0 \quad on \quad y = \eta \tag{3.6d}$$

Equations (3.6a)-(3.6d) are often collectively called the *water-wave problem*.

3.1.1 Justification of Irrotational Flow

Here we will show how the claim of irrotationality in the previous section can be justified. A corollary to Kelvin's theorem [17, ch.5] is that there are three ways to create or destroy vorticity in a flow: Nonconservative body forces, nonbarotropic flow, and viscous forces. Now, we have assumed constant density, which is a special case of barotropic flow, and we consider gravity as the only body force, which is a conservative force.

The assumption of inviscid flow can be justified by considering the *Reynolds number*, which is a dimensionless number relating the intertial and viscous forces in a fluid flow. It is defined as

$$Re = \frac{UL}{\nu} \tag{3.7}$$

Where $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity, U is a typical velocity scale of the problem, and L is a typical length scale of the problem (for instance the wavelength of the gravity waves). Now, if the Reynolds number is sufficiently large (typically $Re \approx 10^3$ or higher), then viscous effects are only significant in thin layers around solid objects in the flow (called *boundary*

layers). For water at atmospheric pressure, the kinematic viscosity is very low, around 10^{-6} [17]. Therefore, as long as the speed and wavelength of the water waves are not extremely small, the Reynolds number should be sufficiently large for inviscid flow to be a very good approximation everywhere, except in a thin layer near the channel bottom[1].

Hence, there are no ways to create or destroy vorticity, according to the corollary to Kelvin's theorem, so if the fluid is initially irrotational, it will remain so for all time. It is natural here to assume the initial state to be the undisturbed water at rest, which is irrotational. The assumption of irrotationality has thus been justified.

3.2 Linearized problem

The problem (3.6a)-(3.6d) is very difficult to solve, due to the nonlinear terms in the boundary conditions. We will therefore *linearize* the equations - that is, we will neglect the nonlinear terms. This can be done if we assume η, ϕ and their derivatives to be much smaller than unity, which makes the nonlinear terms in (3.6a)-(3.6d) negligibly small compared to the linear terms.

We can simplify even further by evaluating the boundary conditions at $y = 0$ instead of $y = \eta$. This can be justified by Taylor expanding the left-hand sides of (3.6c) and (3.6d) around $y = \eta$. For instance, ϕ_y has the Taylor expansion:

$$(\phi_y)_{y=\eta} = (\phi_y)_{y=0} + \eta(\phi_{yy})_{y=0} + ... \tag{3.8}$$

Due to our assumptions on η and ϕ, the first term of this expansion is much larger than the remaining terms, and hence we keep only the first term on the right-hand side. The same reasoning applies to ϕ_t.

The full linearized problem is then

$$\phi_{xx} + \phi_{yy} = 0 \tag{3.9a}$$
$$\phi_y = 0 \qquad on \quad y = -h_0 \tag{3.9b}$$
$$\phi_y = \eta_t \qquad on \quad y = 0 \tag{3.9c}$$
$$\phi_t + g\eta = 0 \qquad on \quad y = 0 \tag{3.9d}$$

This problem can only be solved if the form of the surface η is known. Consequently, we assume the surface to be a simple sinusoidal wave

[1] As mentioned in the introduction, equations which take into account the viscous effects in the boundary layer have been derived, and we will present one of them in section 4.4.

$$\eta(x,t) = a\cos(kx - \omega t) \tag{3.10}$$

Where $k = \frac{2\pi}{\lambda}$ is the spatial frequency, or *wavenumber*, and $\omega = 2\pi\nu$ is the *angular frequency* ($\nu = 1/T$ is the ordinary temporal frequency). If we can construct a solution using a sinusoidal wave, we can also construct arbitrary waveforms by Fourier superposition of the sinusoidal solutions, since the problem is linear. With the sinusoidal surface assumption, the boundary conditions (3.9c) and (3.9d) imply that ϕ must be a sine function of $(kx - \omega t)$, and we make the ansatz:

$$\phi(x,y,t) = f(y)\sin(kx - \omega t) \tag{3.11}$$

Substituting this into the Laplace equation (3.9a), we get $f'' - k^2 f = 0$, which has the general solution $f(y) = Ae^{ky} + Be^{-ky}$. The velocity potential then has the form

$$\phi(x,y,t) = (Ae^{ky} + Be^{-ky})\sin(kx - \omega t) \tag{3.12}$$

The constants A and B can be determined using the boundary conditions. Substituting (3.12) into (3.9b) gives $k(Ae^{-kh_0} - Be^{kh_0})sin(kx - \omega t) = 0$, implying:

$$B = Ae^{-2kh_0} \tag{3.13}$$

Substituting (3.10) and (3.11) into the boundary condition (3.9c), we obtain $k(A-B)\sin(kx - \omega t) = a\omega\sin(kx - \omega t)$, implying:

$$k(A - B) = a\omega \tag{3.14}$$

Solving (3.13) and (3.14) for A and B, we get:

$$A = \frac{a\omega}{k(1 - e^{-2kh_0})} \quad , \quad B = \frac{a\omega e^{-2kh_0}}{k(1 - e^{-2kh_0})}$$

The velocity potential (3.12) then becomes:

$$\phi(x,y,t) = \frac{a\omega}{k}\frac{\cosh(k(y + h_0))}{\sinh(kh_0)}\sin(kx - \omega t) \tag{3.15}$$

The velocity components can easily be derived from this. Finally, we can use the dynamic boundary condition (3.9d) to derive a relationship between ω and k. Substituting the velocity potential (3.15), as well as the sinusoidal waveform (3.10) into (3.9d), we obtain:

$$\omega^2 = gk \tanh(kh_0) \tag{3.16}$$

This kind of relationship between ω and k is known as a *dispersion relation*, and contains valuable information about the behavior of the waves under study. For instance, the phase speed of the waves, $c = \omega/k$, is easily derived from the dispersion relation. Keeping in mind that we have two solutions for ω (plus or minus the square root), the phase speed is

$$c = \pm\sqrt{\frac{g}{k}\tanh(kh_0)} \tag{3.17}$$

The plus and minus sign correspond to right-traveling and left-traveling waves, respectively. This result shows that the phase speed of a linear water wave depends on its wavenumber, and hence on its wavelength. Thus, for a wave that is initially a superposition of different sinusoidal waves, the different wavelengths will travel at different speeds, and the intial wave will disperse, hence the name *dispersion relation*.

Remark. It can be argued that there are two kinds of dispersion contained in the full, nonlinear water-wave problem. The first kind is the property that waves of different wavelengths travel at different speeds, which may be called *frequency dispersion*. Another kind is *amplitude dispersion*, in which points at different heights on the wave travel at different speeds, which may cause the wave to steepen and eventually break. This is an inherently nonlinear phenomenon, and is hence not contained in the linear dispersion relation (3.16). In this thesis, the term dispersion will always refer to frequency dispersion.

Chapter 4

Nonlinear Theory

We have seen in the previous section that one way to circumvent the difficulty of the full water-wave problem (3.6a)-(3.6d) is to linearize the problem. Nonlinear effects are important in many cases, however, and the solution is then to search for a simpler, nonlinear equation that can hopefully still capture some of the properties of the full water-wave problem.

4.1 The Shallow Water Equations

We start by considering the well-known *Shallow Water Equations*. These are a set of nonlinear equations that are valid for either relatively shallow water or relatively long waves, and they can be derived from the full Euler equations (see chapter 13 of [31]). For waves propagating purely in the horizontal direction, the equations are:

$$\begin{aligned} \eta_t + (uh)_x &= 0 \\ u_t + uu_x + g\eta_x &= 0 \end{aligned} \tag{4.1}$$

Were u is the horizontal flow velocity, and $h = h_0 + \eta$ is the distance from the channel bottom to the surface (again it is assumed that h_0 is constant). As before, we can gain information about these equations by finding the dispersion relation of the linearized form of the equations. If we neglect all the nonlinear terms in (4.1), we get

$$\begin{aligned} \eta_t + h_0 u_x &= 0 \\ u_t + g\eta_x &= 0 \end{aligned} \tag{4.2}$$

Differentiating the first equation with respect to t, and the second one with respect to x, the two equations can be combined into

$$\eta_{tt} - c_0^2 \eta_{xx} = 0 \tag{4.3}$$

Where $c_0 = \sqrt{gh_0}$. This is the wave equation. Its dispersion relation can be found in the same way as in the previous section, namely by substituting $\eta = a\cos(kx - \omega t)$ into the equation, and solving for ω. By doing this, we obtain:

$$\omega = c_0 k \tag{4.4}$$

From this we get phase speed $c_0 = \sqrt{gh_0}$. This is in fact the phase speed of the linearized water-wave problem (3.17) in the limit $kh_0 \to 0$, and thus it can be interpreted as the limiting phase speed for linear water waves as the water becomes shallower. Notice that the phase speed is independent of k, which means that the Shallow Water Equations model non-dispersive waves. Since in addition viscosity is neglected, there is no mechanism to balance out the nonlinear steepening, and as a result, the Shallow Water Equations predict that all waves carrying an increase of elevation will break. This is a shortcoming of these equations, as it is not a very realistic property. [31]

4.2 The Korteweg-deVries Equation

In view of the discussion above, we would like to find a nonlinear equation that models dispersive waves. This leads us to the Korteweg-deVries equation. It is well known that this equation admits traveling-wave solutions, which is possible due to the nonlinear and dispersive effects cancelling each other out. It is especially remarkable that these solutions can be written down exactly.

The Korteweg-deVries (KdV) equation can be derived from the full water-wave problem by setting up an asymptotic expansion, and keeping the terms that give the desired balance between nonlinear and dispersive effects. Mathematically, this is done by considering the parameters $\alpha = \frac{a}{h_0}$ and $\beta = \frac{h_0^2}{l^2}$, where a is a typical amplitude, and l a typical wavelength. We require these parameters to be small and approximately of the same order [5].

The full derivation of the KdV equation will be given here. This derivation is based on [31]. We use an x-Y coordinate system, with the x-axis aligned with the flat bottom. Starting from the water-wave problem, we would like to solve the Laplace equation

$$\phi_{xx} + \phi_{YY} = 0 \tag{4.5}$$

With the following boundary conditions

$$\phi_Y = 0 \quad on \quad Y = 0 \tag{4.6}$$

$$\eta_t + \phi_x \eta_x - \phi_Y = 0 \quad on \quad Y = h_0 + \eta \tag{4.7}$$

$$\phi_t + \frac{1}{2}(\phi_x^2 + \phi_Y^2) + g\eta = 0 \quad on \quad Y = h_0 + \eta \tag{4.8}$$

The derivation is best carried out using non-dimensionalized variables. Denoting the original variables by a prime, we introduce the following scalings:

$$x' = lx, \quad Y' = h_0 Y, \quad t' = \frac{lt}{c_0}, \quad \eta' = a\eta, \quad \phi' = \frac{gla\phi}{c_0} \tag{4.9}$$

The new variables will all be of order one, and the relative sizes of the terms can be directly compared. By substituting these new variables into equations (4.5)-(4.8), and using the chain rule for derivatives, we obtain the normalized equations:

$$\beta\phi_{xx} + \phi_{YY} = 0, \quad 0 < Y < 1 + \alpha\eta \tag{4.10}$$

$$\phi_Y = 0 \quad on \quad Y = 0 \tag{4.11}$$

$$\eta_t + \alpha\phi_x\eta_x - \frac{1}{\beta}\phi_Y = 0 \quad on \quad Y = 1 + \alpha\eta \tag{4.12}$$

$$\eta + \phi_t + \frac{1}{2}\alpha\phi_x^2 + \frac{1}{2}\frac{\alpha}{\beta}\phi_Y^2 = 0 \quad on \quad Y = 1 + \alpha\eta \tag{4.13}$$

Here we have used the previously defined parameters α and β. We now make the ansatz that ϕ can be expanded as the asymptotic series

$$\phi = \sum_{n=0}^{\infty} Y^n f_n(x,t) \tag{4.14}$$

Inserting this into (4.10) and using (4.11), we get, after some algebra:

$$\phi = \sum_{m=0}^{\infty} (-1)^m \frac{Y^{2m}}{(2m)!} \frac{\partial^{2m} f}{\partial x^{2m}} \beta^m \tag{4.15}$$

Here we have defined $f \equiv f_0$. Substituting this into the surface conditions (4.12), (4.13), we get

$$\begin{aligned}\eta_t + [(1+\alpha\eta)f_x]_x - \left[\frac{1}{6}(1+\alpha\eta)^3 f_{xxxx} + \frac{1}{2}\alpha(1+\alpha\eta)^3\eta_x f_{xxx}\right]\beta + O(\beta^2) = 0 \\ \eta + f_t + \frac{1}{2}\alpha f_x^2 - \frac{1}{2}(1+\alpha\eta)^2(f_{xxt} + \alpha f_x f_{xxx} - \alpha f_{xx}^2)\beta + O(\beta^2) = 0\end{aligned} \tag{4.16}$$

If we now drop all $O(\beta)$ terms, and differentiate the second equation with respect to x, we obtain the shallow water equations (define $w = f_x$):

$$\begin{aligned}\eta_t + [(1+\alpha\eta)w]_x = 0 \\ w_t + \alpha w w_x + \eta_x = 0\end{aligned}$$

However, as mentioned earlier, we would like α and β to be approximately of the same order. Thus, we keep the $O(\beta)$ terms, but drop the $O(\alpha\beta)$ and higher order terms. By doing this, and again differentiating the second equation of (4.16) with respect to x, we get

$$\begin{aligned}\eta_t + [(1+\alpha\eta)w]_x - \frac{1}{6}\beta w_{xxx} + O(\alpha\beta, \beta^2) = 0 \\ w_t + \alpha w w_x + \eta_x - \frac{1}{2}\beta w_{xxt} + O(\alpha\beta, \beta^2) = 0\end{aligned} \tag{4.17}$$

Here we have also defined $w = f_x$. The Korteweg-deVries equation can be derived from (4.17) by looking for solutions that are right-moving waves. If all $O(\alpha)$ and $O(\beta)$ terms are dropped from (4.17), we get:

$$\begin{aligned}\eta_t + w_x = 0 \\ w_t + \eta_x = 0\end{aligned}$$

One solution to these equations can be found by setting $w = \eta$, in which case the equations reduce to $\eta_t + \eta_x = 0$. This is the transport equation, which is well known for having a right-moving wave as the solution. This suggests making the ansatz that w can be expanded in an asymptotic series, having η and the leading order term, with corrections of order α and β:

$$w = \eta + \alpha A + \beta B + O(\alpha^2 + \beta^2)$$

Where A and B are functions of η and its spatial derivatives. Substituting this into (4.17), we get:

$$\begin{aligned} \eta_t + \eta_x + \alpha(A_x + 2\eta\eta_x) + \beta(B_x - \frac{1}{6}\eta_{xxx}) + O(\alpha^2 + \beta^2) = 0 \\ \eta_t + \eta_x + \alpha(A_t + \eta\eta_x) + \beta(B_t - \frac{1}{2}\eta_{xxt}) + O(\alpha^2 + \beta^2) = 0 \end{aligned} \tag{4.18}$$

Since we seek a solution satisfying the transport equation to first order, we have $\eta_t = -\eta_x + O(\alpha, \beta)$. Thus, we can replace every t-derivative in the $O(\alpha, \beta)$ terms by minus the x-derivative, without changing the equations. It is then easily shown that the two equations are equal, up to $O(\alpha^2 + \beta^2)$, if we define:

$$A = -\frac{1}{4}\eta^2 \quad , \quad B = \frac{1}{3}\eta_{xx}$$

Thus, if we neglect the $O(\alpha^2 + \beta^2)$ terms, the two equations (4.18) become

$$\eta_t + \eta_x + \frac{3}{2}\alpha\eta\eta_x + \frac{1}{6}\beta\eta_{xxx} = 0 \tag{4.19}$$

This is the Korteweg-deVries equation in nondimensional form. By reverting to the dimensional variables, using the relationships of (4.9), we obtain the Kortweg-deVries equation in dimensional form:

$$\eta_t + c_0\eta_x + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + \frac{1}{6}c_0 h_0^2 \eta_{xxx} = 0 \tag{4.20}$$

The dispersion relation of the linearized form of (4.20) is again found by assuming η to be a sinusoidal wave, and the result is

$$\omega = c_0 k - \frac{1}{6}c_0 h_0^2 k^3 \tag{4.21}$$

From this we get the phase speed $c(k) = c_0 - \frac{1}{6}c_0 h_0^2 k^2$. We notice this as being the dispersion relation of the linearized Shallow Water Equations, but with an additional dispersive term.

As mentioned, the Korteweg-de Vries equation is known for having solutions that can be written down exactly. These solutions can be found by searching for solutions in the form of *traveling waves*, which are solutions of the form

$$\eta(x, t) = \phi(x - ct) \tag{4.22}$$

This describes a solution moving with speed c to the left or right (depending on the sign of c), without changing its shape[1]. Substituting (4.22) into the KdV equation turns it into an ODE, and two specific solutions of this ODE can be found (see [31, ch. 13] for details). The first type of solution is a *solitary wave*, which consists of just a single wave crest. The second type of solution is the *cnoidal wave*, which is a periodic wave that can be written in terms of the Jacobi elliptic function *cn*. The solitary wave will be considered in more detail in chapter 6, where we solve the KdV equation numerically.

4.3 The Whitham Equation

In this section we consider the Whitham equation, which was first proposed by Whitham in [32] as an attempt to address some of the shortcomings of the KdV equation. The equation can also be found in chapter 13 of [31].

The main problem with the KdV equation is that it is a poor approximation for water waves of short wavelengths. This can be seen by noting that its linear phase speed is a second order Taylor series expansion of the exact linear phase speed (3.17) for right-moving waves, about $k = 0$:

$$\sqrt{\frac{g}{k}\tanh(kh_0)} = c_0 - \frac{1}{6}c_0 h_0^2 k^2 + ... \tag{4.23}$$

Thus, for large k (or equivalently, short wavelengths), the second order approximation of the linear phase speed becomes inaccurate, as shown in the plot below.

[1] Of course, since the KdV equation only models right-moving waves, c must be positive in this case.

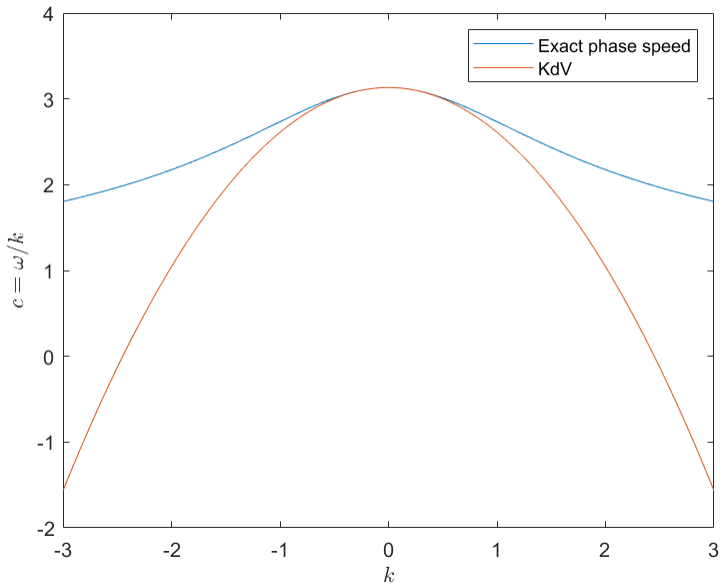


Figure 4.1: Comparison of the exact linear phase speed and the linear phase speed of the KdV equation.

One possible remedy for this would be to find an equation whose linearized version gives the phase speed of the full water-wave problem (3.17). This can be achieved by noting that arbitrary relations for the phase speed can be constructed from the following integro-differential equation [31, ch. 11]:

$$\frac{\partial \eta}{\partial t}(x,t) + \int_{-\infty}^{\infty} K_{h_0}(x-\xi)\frac{\partial \eta}{\partial \xi}(\xi,t)d\xi = 0 \tag{4.24}$$

For some specified kernel K_{h_0} [2]. This equation can more conveniently be written using convolution products:

$$\eta_t + K_{h_0} * \eta_x = 0 \tag{4.25}$$

The dispersion relation is found as before; we assume a sinusoidal waveform. In this case, it is more convenient to work with complex exponentials, so we assume $\eta = Ae^{i(kx-\omega t)}$. Substituting this into (4.24), we get that any non-trivial solution must satisfy

$$-i\omega e^{ikx} + ik\int_{-\infty}^{\infty} K_{h_0}(x-\xi)e^{ik\xi}d\xi = 0 \tag{4.26}$$

[2]We use the subscript h_0 on the kernel because, as we will see, it will depend on h_0, and later we will denote a nondimensional version of this kernel by simply K.

Making the change of variable $\zeta = x - \xi$, we obtain

$$\frac{\omega}{k} = \int_{-\infty}^{\infty} K_{h_0}(\zeta) e^{-ik\zeta} d\zeta \tag{4.27}$$

We notice the right-hand side as being the Fourier transform of the kernel K_{h_0}. Denote the Fourier transform by $\hat{K}_{h_0}(k)$. Then (4.27) can be written very elegantly as

$$c(k) = \hat{K}_{h_0}(k) \tag{4.28}$$

It is now clear how equation (4.24) can produce an arbitrary phase speed relation; simply choose K_{h_0} such that its Fourier transform is the desired phase speed.

The idea is then to combine the nonlinear term of the KdV equation with (4.25), yielding:

$$\eta_t + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + K_{h_0} * \eta_x = 0 \tag{4.29}$$

The KdV equation can be retrieved from (4.29) by choosing K_{h_0} such that:

$$\hat{K}_{h_0} = c_0 - \frac{1}{6} c_0 h_0^2 k^2 \tag{4.30}$$

If we now consider equation (4.29) with

$$\hat{K}_{h_0} = \sqrt{\frac{g}{k}\tanh(kh_0)} \tag{4.31}$$

We get what is called the *Whitham equation*. By the Fourier inversion theorem [3], we can equivalently write (4.31) as

$$K_{h_0} = \mathcal{F}^{-1}\left(\sqrt{\frac{g}{k}\tanh(kh_0)}\right) \tag{4.32}$$

Henceforth, the kernel K_{h_0} will always be given by (4.32). Note that the dispersion relation of the Whitham equation is equal to

[3] We interpret the Fourier and inverse Fourier transforms in terms of distributions here. Since $\hat{K}_{h_0} \in L^\infty(\mathbb{R})$, it is a tempered distribution, and for any tempered distribution T we have that $\mathcal{F}^{-1}\mathcal{F}T = \mathcal{F}\mathcal{F}^{-1}T = T$. [12, ch. 31]

$$\omega(k) = k\hat{K}_{h_0} = k\sqrt{\frac{g}{k}\tanh(kh_0)} \tag{4.33}$$

which is equivalent to the dispersion relation of the full water-wave problem (3.16) if $k > 0$. For $k < 0$, (3.16) and (4.33) are equal in absolute value but have opposite signs. This is of no great concern, however, since the dispersion relation (3.16) is in theory only defined for $k > 0$, as these are the wavenumbers that make sense from a physical standpoint.

Since the phase speed of the Whitham equation agrees with that of the full water-wave problem (at least for $k > 0$), one might expect that this equation will give a more faithful description of waves with shorter wavelengths. Indeed, it was shown in [23] that the Whitham equation gives closer approximations to the full water-wave problem (equations 3.6a-3.6d) than the KdV equation, except for certain kinds of long waves.

In addition, the existence of periodic, traveling-wave solutions to the Whitham equation was proven in [8, 9], and a proof of wave breaking for the Whitham equation was given in [24].

Another prominent feature of the Whitham equation is that it has weaker dispersion than the KdV-equation, which is apparent from a comparison of the slopes of the two functions in figure 4.1 (larger slope means larger variations in phase speeds, i.e., stronger dispersion).

Note that the Whitham kernel (4.32) blows up at the origin, but it is still in $L^1(\mathbb{R})$ [9]. Note also that the function $\sqrt{\frac{g}{k}\tanh(kh_0)}$ has a singularity at the origin, although it is a removable singularity. Since $K_{h_0} \in L^1(\mathbb{R})$, $\hat{K}_{h_0}$ must be continuous on $\mathbb{R}$ [12, ch. 17], and it follows that $\hat{K}_{h_0}(0) = \lim_{k\to 0} \hat{K}_{h_0}(k) = \sqrt{gh_0}$.

Remark. Equations such as the Whitham equation are often called *non-local*, because the convolution product at a point depends on the values of η at all other points in $\mathbb{R}$. By contrast, a purely differential equation is a local equation, since the derivative of η at a point only depends on the values of η in a local neighborhood around that point.

4.4 Dissipative Equations

As mentioned in the introduction, a central feature of the undular bore is a loss of energy, and this should accordingly be a feature of any equation attempting to describe this phenomenon. It is very likely that the undular bore can be described by the Navier-Stokes equations (2.19), as they are supposed to model all kinds of incompressible fluids, and energy loss is accounted for in the viscous term of these equations.

However, as the Navier-Stokes equations are way too difficult to solve numerically, a simpler equation that includes energy loss must be sought. Mathematically, an equation which models energy loss is often called a *disspative* equation, and a central feature of dissipative

equations is that if one considers the linearized form of the equation, then sinusoidal solutions should decay with time [13].

Recall that the wavetrain in an undular bore can be considered cnoidal to a good approximation, and hence the equation we seek should be similar to the KdV equation. One might ask whether the equation we seek is just simply the KdV or Whitham equation. These equations are not dissipative, however, and this is because their dispersion relations are real-valued. If a complex-exponential solution $\eta = Ae^{i(kx-\omega t)}$ to an equation should decay with time, for an arbitrary wavenumber k, then the dispersion relation of the equation must be complex, and of the form:

$$\omega(k) = a(k) - ib(k) \tag{4.34}$$

Where $a(k)$ and $b(k)$ are real-valued, and $b(k) > 0$ for all values of k considered (we might consider all values of k, or perhaps just positive values). This is easily shown by substituting (4.34) back into the complex exponential:

$$\eta = Ae^{i(kx-\omega t)} = Ae^{i(kx-a(k)t+ib(k)t)} = Ae^{i(kx-a(k)t)}e^{-b(k)t} \tag{4.35}$$

From this we see that the real part of the dispersion relation determines the phase speed, while the imaginary part determines the damping rate.

Perhaps the simplest KdV-type equation which is also dissipative is the *Korteweg-de Vries-Burgers (KdV-Burgers) equation*. This equation can be formed by subtracting a double-derivative term from the left-hand side of the dimensional KdV equation (4.20):

$$\eta_t + c_0\eta_x + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + \frac{1}{6}c_0h_0^2\eta_{xxx} - \alpha\eta_{xx} = 0 \tag{4.36}$$

Where $\alpha > 0$ is a constant that determines the strength of the dissipation. If we assume $\eta = Ae^{i(kx-\omega t)}$, and substitute this into the linearized version of (4.36), we obtain the dispersion relation

$$\omega = c_0k - \frac{1}{6}c_0h_0^2k^3 - i\alpha k^2 \tag{4.37}$$

which is of the form (4.34). The KdV-Burgers equation was proposed in [31, ch. 13.15] as a potential model for the undular bore. It has in fact been proven by Bona and Schonbek [3] that the KdV-Burgers equation features solutions in the shape of undular bores. We will attempt to find these solutions numerically in chapter 6.

Now, in section 4.3 we explained how the Whitham equation can be seen as a more accurate version of the KdV equation. One might wonder whether the same procedure can make the

KdV-Burgers equation more accurate[4], i.e., replacing the dispersive term of (4.36) with the convlution product from the Whitham equation

$$\eta_t + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + K_{h_0} * \eta_x - \alpha\eta_{xx} = 0 \tag{4.38}$$

where $K_{h_0} = \mathcal{F}^{-1}\left(\sqrt{\frac{g}{k}\tanh(kh_0)}\right)$ is the previously defined Whitham kernel. Equation (4.38) will be investigated in detail in chapter 6. We will call it the *Whitham-Burgers* equation, as it is a combination of the Whitham equation and the viscous Burgers equation.

It might be the case, however, that neither the KdV-Burgers equation nor the Whitham-Burgers equation are very realistic models of undular bores. The reason is that the dissipative term η_{xx} is difficult to justify from a physical perspective[5]. As shown in [6, 13, 14], the assumption that viscous dissipation happens in a boundary layer near the bottom, as indicated by Sturtevant's calculations [29], results in a non-local integral term, rather than the double-derivative term. The equations obtained in these articles are all very similar, as previously mentioned. We choose to focus on the equations obtained in the article by Kakutani and Matsuuchi [14].

In this article, Kakutani and Matsuuchi consider the Navier-Stokes equations in nondimensional form, and impose similar boundary conditions to those used in the water-wave problem in chapter 3. By expanding the unknown variables in asymptotic series, and matching an outer solution with an inner solution in the boundary layer, they are able to derive a dissipative equation for η. It is derived under the assumption of weak nonlinearity and long wavelengths. They arrive at the equation

$$\eta_t + \frac{3}{2}\eta\eta_x + \frac{1}{6}\eta_{xxx} = \frac{1}{4\sqrt{\pi R^*}}\left(L * \eta_x\right) \tag{4.39}$$

Where R^* is a scaled Reynolds number, and the right-hand side is a convolution product with

$$L = \frac{1 - \operatorname{sgn} x}{\sqrt{|x|}} \tag{4.40}$$

Equation (4.39) can be interpreted as a nondimensional KdV-equation with a dissipative boundary-layer term. In chapter 6, we will attempt to form a new equation by replacing the dispersive term of (4.39) with the convolution term of the Whitham equation, just as we did

[4]By *more accurate* we mean that its solutions are closer to the solutions (presumably) described by the Navier-Stokes equations.

[5]In [13], Johnson is able to derive the KdV-Burgers equation from the Navier-Stokes equations, but under the assumption that the undisturbed flow is already moving due to an inclined bottom. We are, however, assuming a flat bottom throughout the whole thesis.

with the KdV-Burgers equation in order to obtain the Whitham-Burgers equation. We will then solve this new equation numerically, and compare the solutions to the Whitham-Burgers equation.

Chapter 5

Numerical Methods

Most partial differential equations do not possess an analytical solution that can be written in terms of elementary functions, and usually we have to resort to solving the equations numerically on a computer. As computers can only work with finite arithmetic, the continuous problem of solving a PDE must be turned into a discrete problem, or *discretized*. Assuming the solution is a function of space and time, one must find a way to approximate the solution at discrete points in space (*discretizing in space*) and at discrete points in time (*discretizing in time*).

There exist many different algorithms for discretizing in space and time. We will see that for the equations considered in this thesis, it is natural to use a *spectral method* for discretizing in space. Unless otherwise specified, the theory presented in this chapter is based on [7], with some additional insights from [4] regarding the discrete and inverse discrete Fourier transforms.

5.1 Spectral Methods

Spectral methods are a class of different numerical methods for solving PDE's. Let us illustrate their design principles by considering an arbitrary differential equation of the form

$$\frac{\partial u}{\partial t} = F(u) \tag{5.1}$$

To be solved for some function $u(x,t)$. Here $F(u)$ is an operator containing the spatial derivatives of u. Spectral methods involve representing the function u as a series expansion:

$$u(x,t) = \sum_{k=-\infty}^{\infty} \hat{u}_k(t)\phi_k(x) \tag{5.2}$$

However, to be able to implement this on a computer, the series must be truncated:

$$u^N(x,t) = \sum_{k=-N/2}^{N/2} \hat{u}_k(t)\phi_k(x) \tag{5.3}$$

This will clearly make u^N only an approximate solution of the PDE, and so we need a way to minimize the residual (in some norm)

$$\frac{\partial u^N}{\partial t} - F(u^N) \tag{5.4}$$

There are several ways to do this, giving rise to different types of spectral methods. For our computations, we will use the *collocation method*. Suppose we have a discrete spatial grid $\{x_j\}$. The collocation method then requires the differential equation to be satisfied exactly at the grid points:

$$\frac{\partial u^N}{\partial t} - F(u^N)\bigg|_{x=x_j} = 0 \tag{5.5}$$

Different types of spectral methods are also obtained by choosing different kinds of basis functions $\{\phi_k\}$ in the series expansion (5.2). Commonly used basis functions are trigonometric polynomials, Chebyshev polynomials and Legendre polynomials. We will use trigonometric polynomials for our computations, which in combination with (5.5) yields the *Fourier collocation method*.

5.2 The Fourier Collocation Method

The Fourier Collocation Method is obtained by choosing the trigonometric basis functions $\phi_k(x) = e^{ikx}$ in the series expansion (5.2), resulting in a *Fourier series*, which is formally defined as follows

Definition 5.2.1. *If a function $u(x,t)$ is 2π-periodic and integrable on $x \in [0, 2\pi]$, then the* Fourier series *of u is defined to be*

$$\sum_{k=-\infty}^{\infty} \hat{u}_k(t)e^{ikx} \tag{5.6}$$

Where the Fourier coefficients $\hat{u}_k$ *are defined as*

$$\hat{u}_k(t) = \frac{1}{2\pi} \int_0^{2\pi} u(x,t) e^{-ikx} dx \tag{5.7}$$

Remark 1. The Fourier series can be defined for a periodic function of any period P, but the resulting formulas become simpler if we specifically assume 2π-periodicity. We will however see that any P-periodic function can be made 2π-periodic through a scaling of variables.

Remark 2. Note that we treat the variable t as a constant in this definition, so formula (5.6) is in fact the Fourier series of a function $u(x)$ of one variable.

We truncate the Fourier series as follows

$$P_N u(x,t) = \sum_{k=-N/2}^{N/2-1} \hat{u}_k(t) e^{ikx} \tag{5.8}$$

Note that this particular truncation requires N to be even, which we will assume is the case throughout the whole thesis.

In definition 5.2.1, we have not set u to be equal to its Fourier series. This is because it is not automatically given that a function converges to its Fourier series. There are various results about this, regarding different kinds of convergence. One well-known result is that of functions in $L^2(0, 2\pi)$. For these functions, we have the following theorem about convergence of their Fourier series [7, 11]:

Theorem 5.2.1. *If u is 2π-periodic, and $u \in L^2(0, 2\pi)$, then*

$$u = \sum_{k=-\infty}^{\infty} \hat{u}_k e^{ikx} \tag{5.9}$$

Where the series converges in norm, i.e.

$$\int_0^{2\pi} |u(x) - P_N u(x)|^2 dx \to 0 \quad as \quad N \to \infty \tag{5.10}$$

Henceforth, we will assume all of the functions considered to be in $L^2(0, 2\pi)$. This will certainly be the case for all the numerical solutions found in chapter 6.

Note that *any* function $u \in L^2(0, 2\pi)$ can be written as a Fourier series on $[0, 2\pi)$, not just functions that are 2π-periodic on the whole line. We can define a *periodic extension* of u (call it u_{per}) to the whole line by the formula:

$$u_{per}(x + 2n\pi) = u(x) \quad \text{for all} \quad x \in [0, 2\pi) \quad \text{for all} \quad n \in \mathbb{Z} \tag{5.11}$$

Since $u = u_{per}$ on $[0, 2\pi)$, the Fourier series of u_{per} also converges to u on this interval. Fourier collocation can accordingly be used to approximate any L^2-function on $[0, 2\pi)$; it is implied by the method that the function is periodically extended outside this interval.

5.2.1 The Discrete Fourier Transform

We cannot implement the truncated Fourier series (5.8) directly on a computer, as in general the Fourier coefficients cannot be computed exactly and must be approximated. This can for instance be done by the trapezoidal rule, which leads us to the *discrete Fourier transform*. We discretize the interval $[0, 2\pi]$ by using the following grid points, for some even integer $N > 0$:

$$x_j = \frac{2\pi j}{N} \quad , \quad j = 0, ..., N \tag{5.12}$$

Applying the trapezoidal rule to the continuous Fourier coefficients (5.7), using the grid points above, we get

$$\hat{u}_k \approx \frac{1}{2\pi}\frac{2\pi}{N}\left(\frac{1}{2}u(x_0) + \sum_{j=1}^{N-1} u(x_j)e^{-ikx_j} + \frac{1}{2}u(x_N)\right) \tag{5.13}$$

Where we have used that $e^{-i2\pi k} = 1$ for any integer k. Since we assume that u is 2π-periodic, we can set $u(x_0) = u(x_n)$, and (5.13) becomes (we define $u_j \equiv u(x_j)$):

$$\hat{u}_k \approx \frac{1}{N}\sum_{j=0}^{N-1} u_j e^{-ikx_j} \tag{5.14}$$

Note that the final grid point from (5.12) has been removed, due to the periodicity. This formula computes what are called the *discrete Fourier coefficients* of the function u, and we denote them by $\tilde{u}_k$, to distinguish them from the exact Fourier coefficients. In view of (5.8), we would like to compute the coefficients for $-N/2 \leq k \leq N/2 - 1$:

$$\tilde{u}_k = \frac{1}{N}\sum_{j=0}^{N-1} u_j e^{-ikx_j} \quad , \quad -N/2 \leq k \leq N/2 - 1 \tag{5.15}$$

The function values u_j can be retrieved from the discrete Fourier coefficients by the inversion formula (see [4, p. 29-30] for a proof):

$$u_j = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k e^{ikx_j} \quad , \quad j = 0, ..., N-1 \tag{5.16}$$

The mapping from the sequence $\{u_j\}$ to the sequence $\{\tilde{u}_k\}$ is called the *discrete Fourier transform* (DFT), and denoted by the symbol $\mathcal{D}$. The *inverse discrete Fourier transform* (IDFT), denoted by $\mathcal{D}^{-1}$, is the mapping from $\{\tilde{u}_k\}$ to $\{u_j\}$:

$$\mathcal{D}\{u_j\} = \{\tilde{u}_k\} \quad , \quad \mathcal{D}^{-1}\{\tilde{u}_k\} = \{u_j\} \tag{5.17}$$

As indicated by their names, and from the formulas (5.15) and (5.16), the DFT and IDFT are inverses of each other, i.e., $\mathcal{D}^{-1}(\mathcal{D}\{u_j\}) = \{u_j\}$. Below are a couple of other important properties of the DFT/IDFT that will be of use.

Theorem 5.2.2.
a) If $\{u_j\} = \{v_j\}$, then $\mathcal{D}\{u_j\} = \mathcal{D}\{v_j\}$
b) $\mathcal{D}\{\alpha u_j + \beta v_j\} = \alpha\mathcal{D}\{u_j\} + \beta\mathcal{D}\{v_j\}$ (Linearity)
c) The discrete and inverse discrete Fourier coefficients are both N-periodic, i.e., $u_{j+N} = u_j$ and $\tilde{u}_{k+N} = \tilde{u}_k$ for all integers j and k.

Properties a) and b) follow easily from the definition of the DFT. Property c) follows from the fact that $e^{-i(k+N)x_j} = e^{-2\pi i(k+N)j/N} = e^{-2\pi ikj/N} = e^{-ikx_j}$, and $e^{ikx_{j+N}} = e^{2\pi i(j+N)k/N} = e^{2\pi ijk/N} = e^{ikx_j}$.

Finally, we define the interpolation operator

$$I_N u(x) = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k e^{ikx} \tag{5.18}$$

Which has the property that $I_N u_j = u_j$. This operator forms the basis of the Fourier collocation method, as it is an approximation of u that can be directly implemented on a computer.

5.2.2 Implementing the DFT in MATLAB

In this section we make some remarks about how to implement the DFT in MATLAB. The DFT-IDFT pair comes in many different forms, corresponding to various conventions, and so it is important to know which convention is used by a specific software.

In MATLAB, the DFT and IDFT are computed by calling the functions **fft** and **ifft**, respectively. These are based on a famous algorithm called the Fast Fourier Transform (FFT),

which computes the discrete Fourier coefficients very quickly. The formulas used by MATLAB to compute the **fft** and **ifft** are as follows [21]:

$$Y(k) = \mathbf{fft}(X) = \sum_{j=1}^{N} X(j) e^{-2\pi i (j-1)(k-1)/N} \tag{5.19}$$

$$X(j) = \mathbf{ifft}(Y) = \frac{1}{N} \sum_{k=1}^{N} Y(k) e^{2\pi i (j-1)(k-1)/N} \tag{5.20}$$

These formulas differ from (5.15) and (5.16) in two ways. First, the factor $1/N$ is placed on the IDFT instead of the DFT. This is of no concern, as the combined effect of taking the DFT and IDFT will be the same, no matter where the constant is placed.

Another more significant difference is that the discrete Fourier coefficients are computed for $k = 0, ..., N-1$ instead of $k = -N/2, ..., N/2-1$. However, these two sets of wavenumbers will produce the exact same discrete Fourier coefficients, but in different orders. This is due to the discrete Fourier coefficients $Y(k)$ being N-periodic. Consequently, MATLAB's **fft** computes the discrete Fourier coefficients in the equivalent order $k = 0, ...N/2 - 1, -N/2, -N/2 + 1, ..., -1$. This order is very important to keep in mind, as pointwise multiplication by k will be done when using spectral methods. Note also that using a different set of wavenumbers does not affect the **ifft** at all, as the sums in (5.16) and (5.20) will be the exact same; the only difference is that the terms are summed in a different order.

By the same argument as above, we can justify using Fourier collocation on a function sampled at the grid points $\frac{2\pi j}{N}$ for any N consecutive integers j. Since $X(j)$ is N-periodic, $Y(k)$ will always be the same, just summed in different orders, depending on the values of j. Similarly, taking the **ifft** produces the same set of inverse discrete Fourier coefficients, just ordered differently depending on the values of j. This fact will be used in some of our numerical experiments, where the indices $j = -N/2, ..., N/2 - 1$ will be used, corresponding to the interval $[-\pi, \pi]$.

5.3 Sources of Error

One type of error that is clearly introduced by the Fourier collocation method is a *truncation error*, i.e., the error introduced by truncating the Fourier series. The significance of this error depends on how quickly the Fourier coefficients decay for large values of k. A result along these lines, proven in [7], is the following:

Theorem 5.3.1. *If u is m-times continuously differentiable in $[0, 2\pi]$, and the j-th derivative $u^{(j)}$ is periodic for all $j \leq m - 2$, then*

$$\hat{u}_k = O(k^{-m}) \tag{5.21}$$

This shows that the more derivatives the function has, the more quickly its Fourier coefficients decay. In particular, the Fourier coefficients of an infinitely differentiable function decay faster than any negative power of k. This is called *spectral accuracy*, and is one of the reasons why spectral methods are so effective.

An additional source of error is introduced by working with the interpolation operator (5.18) instead of the truncated Fourier series. It can be shown, by inserting the Fourier series into the discrete Fourier coefficients (5.15), that the following formula holds (see [4, p. 181-182] for details):

$$\tilde{u}_k = \hat{u}_k + \sum_{\substack{m=-\infty \\ m\neq 0}}^{\infty} \hat{u}_{k+Nm} \tag{5.22}$$

This formula tells us that the k-th discrete Fourier coefficient is equal to the exact Fourier coefficient plus the sum of all Fourier coefficients that are "indistinguishable" from $\hat{u}_k$ on the grid. For each j, we have that $e^{ikx_j} = e^{i(k+Nm)x_j}$, and hence these frequencies are indistinguishable on the grid (this phenomenon is known as *aliasing*).

It follows immediately from (5.22) that

$$I_N u = P_N u + R_N u \tag{5.23}$$

Where

$$R_N u = \sum_{k=-N/2}^{N/2-1} \left(\sum_{\substack{m=-\infty \\ m\neq 0}}^{\infty} \hat{u}_{k+Nm} \right) e^{ikx} \tag{5.24}$$

Hence, $R_N u$ is the error introduced due to aliasing, and it is consequently called the *aliasing error*. One might expect that the interpolation error, $||u - I_N u||$, will be much worse than the truncation error, $||u - P_N u||$, since the interpolation involves both truncation and alisaing effects. However, it is proven in chapter 9 of [7] that the two errors are asymptotically of the same order. As a result, the interpolation operator preserves the spectral accuracy.

5.4 Differentiation

Fourier collocation enables us to approximate derivatives in a simple and accurate way. It can be shown that if $\sum_{k=-\infty}^{\infty} \hat{u}_k e^{ikx}$ is the Fourier series of u, and the derivative u' satisfies

sufficient conditions for convergence of its Fourier series (for instance if $u' \in L^2(0, 2\pi)$, which we will assume is the case), then u' is equal to its Fourier series given by [7, 11]

$$u' = \sum_{k=-\infty}^{\infty} ik\hat{u}_k e^{ikx} \tag{5.25}$$

Hence, we can truncate this Fourier series as before to obtain an approximation of the derivative:

$$u' \approx P_N u' = \sum_{k=-N/2}^{N/2-1} ik\hat{u}_k e^{ikx} \tag{5.26}$$

Note that this is equivalent to differentiating the truncated series (5.8), i.e., $(P_N u)' = P_N u'$. By theorem 5.3.1, this approximation is spectrally accurate. However, as before, we can not implement (5.26) directly on a computer, due to the continuous Fourier coefficients. The appropriate way to approximate the derivatives at the grid points is to instead differentiate the interpolation operator (5.18):

$$u'(x_j) \approx (I_N u)'(x_j) = \sum_{k=-N/2}^{N/2-1} ik\tilde{u}_k e^{ikx_j} \tag{5.27}$$

In contrast to the truncated Fourier series, $(I_N u)' \neq I_N(u')$ (unless $u = I_N u$), and as a result the approximation (5.27) is not exact at the grid points. However, it is once again proven in chapter 9 of [7] that the error $||u' - (I_N u)'||$ is asymptotically of the same order as $||u' - P_N u'||$, and it follows that approximating the derivative by (5.27) is spectrally accurate.

Clearly the approximation (5.27) can be generalized to any n-th derivative by just repeatedly multiplying by ik:

$$u^{(n)}(x_j) \approx (I_N u)^{(n)}(x_j) = \sum_{k=-N/2}^{N/2-1} (ik)^n \tilde{u}_k e^{ikx_j} \tag{5.28}$$

We can compute (5.28) by taking the DFT, multiplying pointwise by $(ik)^n$, and then taking the IDFT. Here special attention needs to be paid to the ordering of the wavenumbers, as explained in section 5.2.2.

5.5 The Discrete Cosine Transform

There is another discrete transform, with a corresponding inverse transform, that is closely related to the DFT. It is called the *Discrete Cosine Transform* (DCT), and as indicated by the name, it represents a set of data points as a sum of pure cosines, rather than complex exponentials as with the DFT.

The DCT can be derived in several ways and in different forms. Here we will show a derivation that, like our derivation for the DFT, seeks to approximate the continuous Fourier coefficients by the trapezoidal rule. This time we make the extra assumption that our function ϕ [1], in addition to being 2π-periodic, is *even* about π. In other words, $\phi(x) = \phi(2\pi - x)$. It can be shown that in this case, the Fourier series of the function becomes a pure cosine series:

$$\phi(x) = \sum_{k=0}^{\infty} a_k \cos(kx) \tag{5.29}$$

Where the *cosine coefficients* a_k are related to the regular Fourier coefficients $\hat{\phi}_k$ (5.7) by the formulas $a_0 = \hat{\phi}_0$ and $a_k = \hat{\phi}_k + \hat{\phi}_{-k}$ (see [11, ch. 2] for more details). We may then define the *cosine collocation method* as a special case of the Fourier collocation method, where we seek to approximate ϕ by a truncated form of its cosine series (5.29). This leads us to the DCT and IDCT.

We will sample the function at the points

$$x_n = \frac{(2n-1)\pi}{2N} \quad , \quad n = 1, ..., 2N+1 \tag{5.30}$$

Next, we use the following result about integrating periodic functions (for the proof, see [11, p. 21]):

Lemma. *If F is periodic with period P, then $\int_a^{a+P} F(x)dx$ is independent of a.*

Hence we can choose to integrate the Fourier coefficients (5.7) from $\frac{\pi}{2N}$ to $2\pi + \frac{\pi}{2N}$, in accordance with (5.30), and the result should be the same. Approximating the Fourier coefficients by the trapezoid rule then yields

$$\hat{\phi}_k \approx \frac{1}{2\pi}\frac{\pi}{N}\left(\frac{1}{2}\phi(x_1)e^{-ikx_1} + \sum_{n=2}^{2N}\phi(x_n)e^{-ikx_n} + \frac{1}{2}\phi(x_{2N+1})e^{-ikx_{2N+1}}\right) \tag{5.31}$$

[1] We will denote the function by ϕ instead of u in the derivation of the DCT, in order to distinguish it from the DFT. This is also the letter we will denote the function by when using the DCT in chapter 6.

Using the fact that $\phi(x_1)e^{-ikx_1} = \phi(x_{2N+1})e^{-ikx_{2N+1}}$, this can be simplified to

$$\hat{\phi}_k \approx \frac{1}{2N}\left(\sum_{n=1}^{2N}\phi(x_n)e^{-ikx_n}\right) \tag{5.32}$$

Since ϕ is even about π, we have that $\phi(x_n) = \phi(x_{2N-n+1})$, since $x_{2N-n+1} = 2\pi - x_n$. This, together with the identities $e^{ik(2\pi-x)} = e^{-ikx}$ and $e^{ikx} + e^{-ikx} = 2\cos(x)$ allows us to rewrite (5.32) to

$$\hat{\phi}_k \approx \frac{1}{N}\sum_{n=1}^{N}\phi(x_n)\cos(kx_n) \tag{5.33}$$

This is one version of the discrete cosine transform. In view of (5.29), we would like to compute the coefficients for $k = 0, ..., N-1$:

$$\tilde{\phi}_k = \frac{1}{N}\sum_{n=1}^{N}\phi(x_n)\cos(kx_n) \quad , \quad k = 0, ..., N-1 \tag{5.34}$$

The function values can be retrieved from (5.34) by the *Inverse Discrete Cosine Transform* (IDCT), which is given by

$$\phi(x_n) = \sum_{k=0}^{N-1} w(k)\tilde{\phi}_k\cos(kx_n) \quad , \quad n = 1, ..., N \tag{5.35}$$

Where

$$w(k) = \begin{cases} 1 & , \quad k = 0 \\ 2 & , \quad k \geq 1 \end{cases} \tag{5.36}$$

We omit the derivation of (5.35), but note that it can be derived from the IDFT[2]. We remark that half of the collocation points (5.30) have been removed in (5.34), and hence the actual grid used in the DCT is

$$x_n = \frac{(2n-1)\pi}{2N} \quad , \quad n = 1, ..., N \tag{5.37}$$

[2]The DCT can also be derived directly from the DFT; it is given as an exercise in [4, p. 141-142]

This new grid lies in the interval $[0, \pi]$. The DCT and IDCT can accordingly be used on *any* function sampled at (5.37), even if the function is neither even about π nor periodic. An even periodic extension about π is automatically implied when using the DCT and IDCT.

5.5.1 Implementing the DCT in MATLAB

Just like the DFT, the DCT and IDCT come in many different forms, and it is important to know which conventions a specific software uses. In MATLAB, the DCT and IDCT are computed by calling the functions **dct** and **idct**, respectively. Note that MATLAB can actually compute four different variants of the DCT/IDCT, but unless otherwise specified, the default variants computed are the ones given below [20, 22].

$$\mathbf{dct}(x) = y(k) = \sum_{n=1}^{N} v(k)x(n)\cos(kx_n) \tag{5.38}$$

$$\mathbf{idct}(y) = x(n) = \sum_{k=0}^{N-1} v(k)y(k)\cos(kx_n) \tag{5.39}$$

Where

$$v(k) = \begin{cases} \sqrt{1/N} & , \quad k = 0 \\ \sqrt{2/N} & , \quad k \geq 1 \end{cases} \tag{5.40}$$

We see that the only differences between MATLAB's conventions and our formulas (5.34) and (5.35) are the different sets of constants $v(k)$ and $w(k)$. It is easy to see, however, that the combined effect of taking the DCT and IDCT will be the same, regardless of which sets of constants are being used. We will, however, stick to the forms (5.38) and (5.39) in all of our numerical experiments, as we will at one point use MATLAB's **dct**, and afterwards compute the IDCT "manually".

Finally, we can define an interpolation operator in terms of the discrete cosine coefficients (using MATLAB's conventions), which we will denote by $I_N^{DCT}\phi$:

$$I_N^{DCT}\phi(x) = \sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \cos(kx) \tag{5.41}$$

It has the property that $I_N^{DCT}\phi(x_n) = \phi(x_n)$. This operator forms the basis of cosine collocation, as it is an approximation of the cosine series of ϕ that can be implemented on a computer.

Chapter 6

Numerical Experiments

We will now use the theory of chapter 5 to numerically solve the dissipative equations mentioned in section 4.4. We are especially interested in finding solutions that are traveling waves. Recall from earlier that these are solutions moving unchanged in shape with speed c, and they can be written in the form

$$\eta(x,t) = \phi(x - ct) \tag{6.1}$$

Before considering the dissipative equations, however, we will first solve the KdV equation. This is a good equation to test that our numerical method works as intended, since it has known exact solutions that we can directly compare our numerical solutions with.

6.1 The KdV Equation

Recall that the KdV equation is given in dimensional form as:

$$\eta_t + c_0\eta_x + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + \frac{1}{6}c_0h_0^2\eta_{xxx} = 0 \tag{6.2}$$

When designing a numerical scheme for a PDE, it is often convenient to consider the equation in the simplest form possible. This can be achieved by a scaling of the independent variables x and t, and of the function η. This procedure will be shown below.

We introduce the following scalings, which makes the quantities nondimensional[1]:

[1]Henceforth, we will denote the original and scaled variables by the same letter. For instance, the scaling $x \mapsto h_0x$ means that the original variable x' and the scaled variable x are related by the formula $x' = h_0x$.

$$x \mapsto h_0 x \quad , \quad t \mapsto \frac{h_0}{c_0} t \quad , \quad \eta \mapsto h_0 \eta \tag{6.3}$$

By using the chain rule for derivatives, we find that the scaled version of (6.2) becomes

$$\eta_t + \eta_x + \frac{3}{2}\eta\eta_x + \frac{1}{6}\eta_{xxx} = 0 \tag{6.4}$$

Furthermore, we can get rid of the η_x term by the transformation $\eta(x,t) \mapsto \eta(x-t,t)$. The physical interpretation of this transformation is that we regard the function in a reference frame moving to the right at a normalized speed of one. Finally, we can scale the factors $3/2$ and $1/6$ to unity by the transformations:

$$x \mapsto \frac{1}{\sqrt{6}} x \quad , \quad t \mapsto \frac{1}{\sqrt{6}} t \quad , \quad \eta \mapsto \frac{2}{3}\eta \tag{6.5}$$

Thus we obtain the KdV equation in the very simple form:

$$\eta_t + \eta\eta_x + \eta_{xxx} = 0 \tag{6.6}$$

One form of the exact solution to (6.6) is given by [28]:

$$\eta(x,t) = 12c^2 \operatorname{sech}^2\left(c(x - 4c^2 t - \alpha)\right) \tag{6.7}$$

Which can easily be verified by substituting this function into (6.6). This is a solitary-wave solution, where c is proportional to the speed of the wave, and α gives the phase.

We would like to solve the equation on the interval $[0, 2\pi]$, since this is the natural setting for the Fourier collocation method, as described in section 5.2. However, we see from (6.7) that the phase speed c needs to be quite large in order for the solitary wave to be contained in a small interval such as $[0, 2\pi]$. This is turn makes the amplitude of the wave very large, which makes the scheme unstable.

Therefore, we must solve the equation on a larger interval, and we choose $L = 40$, as done in [19]. However, we can employ yet another scaling in order to get the equation redefined on $[0, 2\pi]$. We define the original problem as

$$\eta_t + \eta\eta_x + \eta_{xxx} = 0 \quad , \quad 0 \leq x \leq L \tag{6.8}$$

and we introduce the same scalings as in [19]:

$$x \mapsto ax \quad , \quad t \mapsto t \quad , \quad \eta \mapsto a\eta \tag{6.9}$$

Where $a = \frac{L}{2\pi}$. By the chain rule, the terms transform as $\eta_t \mapsto a\eta_t$, $\eta\eta_x \mapsto a\eta\eta_x$ and $\eta_{xxx} \mapsto \frac{1}{a^2}\eta_{xxx}$. The rescaled problem then becomes

$$\eta_t + \eta\eta_x + \frac{1}{a^3}\eta_{xxx} = 0 \quad , \quad 0 \leq x \leq 2\pi \tag{6.10}$$

The re-scaled exact solution is

$$\eta(x,t) = \frac{12}{a}c^2 \operatorname{sech}^2\left(c(ax - 4c^2t - \alpha)\right) \tag{6.11}$$

Note that the scalings lower the amplitude of the wave, which helps with the stability of the scheme. Next, we discretize equation (6.10) in space by the Fourier collocation method. First, we rewrite the nonlinear term in the equation to a more convenient form:

$$\eta_t + \frac{1}{2}(\eta^2)_x + \frac{1}{a^3}\eta_{xxx} = 0 \quad , \quad 0 \leq x \leq 2\pi \tag{6.12}$$

The collocation points are, as defined in section 5.2:

$$x_j = \frac{2\pi j}{N} \quad , \quad j = 0, ..., N-1$$

As described in the same section, the Fourier collocation method consists of approximating η by its trigonometric interpolant (5.18):

$$\eta_N \equiv I_N\eta(x) = \sum_{k=-N/2}^{N/2-1} \tilde{\eta}_k(t)e^{ikx} \tag{6.13}$$

and enforcing equation (6.12) at the collocation points:

$$\frac{\partial \eta_N}{\partial t} + \frac{1}{2}\frac{\partial(\eta_N)^2}{\partial x} + \frac{1}{a^3}\frac{\partial^3 \eta_N}{\partial x^3}\bigg|_{x=x_j} = 0 \quad , \quad j = 0, 1, ..., N-1 \tag{6.14}$$

Equation (6.14) can equivalently be stated in Fourier space, which makes it easier to work with. Notice that (6.14) states that two vectors with $N-1$ components should be equal.

Therefore, by Theorem 5.2.2 (a), we can take the DFT of both sides of (6.14). The right-hand side stays the same, as $\mathcal{D}\{0\} = 0$. By Theorem 5.2.2 (b), the DFT on the left-hand side is the DFT of each individual term. Thus, we can take the appropriate derivatives of each term of (6.14), and then take the DFT of each term to obtain the equivalent equations:

$$\tilde{\eta}_t + \frac{1}{2} ik\widetilde{(\eta^2)} - ik^3 a^{-3}\tilde{\eta} = 0 \tag{6.15}$$

Where $k = (-N/2, ..., N/2-1)^T$ and $\tilde{\eta} = (\tilde{\eta}_{-N/2}, ..., \tilde{\eta}_{N/2-1})^T$ are to be interpreted as vectors, and all the vectors in the equation are multiplied pointwise.

Remark. The nonlinear term is evaluated in a pseudospectral way, which means that we take the IDFT of $\tilde{\eta}$ to obtain η, perform the multiplication η^2 in physical space, and then take the DFT. This is reflected in the notation $\widetilde{(\eta^2)}$. Note that evaluating the nonlinear term in this way introduces an aliasing error (see [7, ch. 3] for details). There are techniques that can be used to remove this aliasing error, although we will not use these techniques in our codes. The error analysis done below for the KdV equation indicate that the scheme is still sufficiently accurate with this aliasing error present, at least for large enough N.

For the discretization in time, we will use the Implicit Trapezoid method. If we have an ODE (or system of ODE's) of the form

$$y'(t) = f(t, y) \tag{6.16}$$

Then the Implicit Trapezoid Method is given by

$$y_{n+1} = y_n + \frac{\Delta t}{2}(f_{n+1} + f_n) \tag{6.17}$$

Where $\Delta t = t_{n+1} - t_n$ is the time step and $y_n = y(t_n)$ is the approximate value of y at the n-th step. Applying the Implicit Trapezoid Method to the semi-discrete equation (6.15), we obtain

$$\tilde{\eta}_{n+1} = \tilde{\eta}_n + \frac{\Delta t}{2}\left[ik^3 a^{-3}(\tilde{\eta}_{n+1} + \tilde{\eta}_n) - \frac{ik}{2}\widetilde{(\eta_n^2)} - \frac{ik}{2}\widetilde{(\eta_{n+1}^2)}\right] \tag{6.18}$$

Which, after some algebra, becomes

$$\tilde{\eta}_{n+1} = \frac{1 + \frac{\Delta t}{2} ik^3 a^{-3}}{1 - \frac{\Delta t}{2} ik^3 a^{-3}}\tilde{\eta}_n - \frac{\frac{\Delta t}{4} ik}{1 - \frac{\Delta t}{2} ik^3 a^{-3}}\left[\widetilde{(\eta_n^2)} + \widetilde{(\eta_{n+1}^2)}\right] \tag{6.19}$$

If we can solve this equation for $\tilde{\eta}_{n+1}$, we can take the IDFT and obtain the approximate value of η at the next time step. However, this equation cannot be solved exactly for $\tilde{\eta}_{n+1}$, so we use fixed-point iteration to solve it numerically, as done in [8, 19]. The iteration works as follows:

1. Use η_n as the initial guess for η_{n+1} on the right-hand side of (6.19). Take the DFT of both η_n and η_n^2, and compute the right-hand side. The IDFT of the computed right-hand side is the new guess for η_{n+1}.

2. Use the new guess from step 1 to compute $\widetilde{(\eta_{n+1}^2)}$ on the right-hand side again. The IDFT of the entire right-hand side is once again the new guess for η_{n+1},

3. Repeat step 2 until convergence is achieved.

We choose to let the iteration run until the difference in norm between two iterations is $< 10^{-12}$. The iteration will converge quite quickly if the time step is sufficiently small. This is due to the fact that for small time steps, η_n will already be very close to η_{n+1}.

The scheme is run with initial data equal to

$$\eta(x,0) = \frac{12}{a}\operatorname{sech}^2(ax - 20) \tag{6.20}$$

Which is the exact solution (6.11) with $c = 1$, $\alpha = 20 = L/2$. This choice of α ensures that the wave crest is located in the middle of the computational domain, which in turn makes the initial data periodic at the boundaries. When using the Fourier collocation method, it is important that the initial data is either periodic, or very close to being so, on the computational domain. If the initial data is non-periodic, the implied periodic extension will create jump discontinuities at the boundaries. We know from section 5.3 that this will make its Fourier series converge far more slowly, such that a truncated Fourier series would need far more terms in order to maintain sufficient accuracy. In addition, the truncated Fourier series of a discontinuous function will oscillate rapidly near the discontinuities, which is called the *Gibbs phenomenon* [7]. These oscillations will make the scheme highly inaccurate and unstable.

The scheme is run until $T = 0.1$, and compared with the exact solution at this time. The comparison is done by computing the discrete L^2-error of the approximate solution at time T, which is defined as

$$||e||_2 = \left(h \sum_{i=1}^{N} (\eta_i^{exact} - \eta_i^{approx})^2 \right)^{1/2} \tag{6.21}$$

Note that the L^2-error is computed for the solution found on $[0, 2\pi]$ (we do not scale it back to $[0, L]$ when computing the error). The convergence rate in space can be found by choosing

a small time step, say $\Delta t = 10^{-5}$, and computing the error at T for a progressively larger number of grid points N. The results are shown in the table below. The third column shows the ratio of the previous and current error.

N	$\|\|e\|\|_2$	Ratio
16	0.4074	
32	0.2528	1.6119
64	0.0131	19.2376
128	3.5715e-06	3.6791e+03
256	1.3997e-10	2.5516e+04
512	1.3991e-10	1.0004

We see that the errors drop at an exponential rate until $N = 512$, where presumably rounding errors become significant. This exponential convergence rate is precisely the spectral accuracy that was mentioned in section 5.3.

Next, we compute the convergence rate in time. This is done by choosing a fixed number of grid points, say $N = 256$, and computing the errors at T for progressively smaller time steps. The results are shown in the table below.

Δt	$\|\|e\|\|_2$	Ratio
0.005	3.4966e-05	
0.005/2	8.7420e-06	3.9999
0.005/4	2.1855e-06	4.0000
0.005/8	5.4638e-07	4.0000
0.005/16	1.3660e-07	4.0000
0.005/32	3.4149e-08	4.0000
0.005/64	8.5372e-09	4.0000
0.005/128	2.1344e-09	3.9999

We see that the ratio between consecutive errors is approximately 4, which is to be expected, as the Implicit Trapezoid Method is a second-order method [27]. The fact that the computed convergence rates match the theoretical ones is a very good indication that the numerical scheme works as intended. A plot of the exact and numerical solution at time $t = 0.1$, with time step $\Delta t = 0.05/4$, is included below (here we have scaled the solution back to $[0, L]$). We see that the difference between the exact and numerical solutions is barely noticeable.

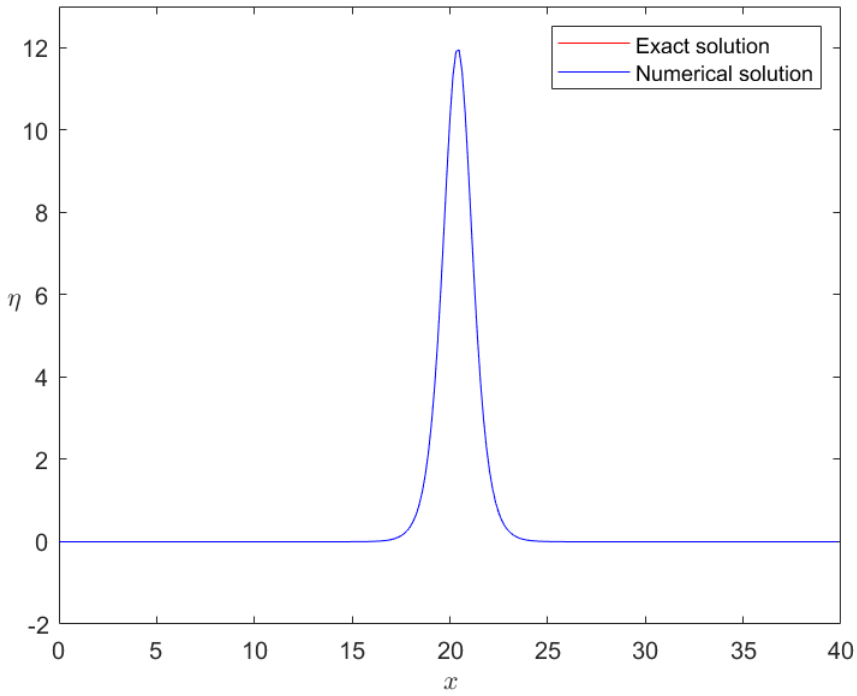


Figure 6.1: Comparison of exact and numerical solitary-wave solutions of the KdV equation. Here $t = 0.1$ and the time step is $\Delta t = 0.05/4$.

Remark. In the code, we set the Fourier mode $k = -N/2$ to zero. The reason is that for spectral differentiation of *odd* order, this mode will make the solution complex-valued, which is not desirable [30]. This can be illustrated by using formula (5.28) for the spectral n-th derivative, and insert the formula for the DFT:

$$\eta^{(n)}(x_m, t) \approx \sum_{k=-N/2}^{N/2-1} (ik)^n \tilde{\eta}_k e^{ikx_m} = \sum_{k=-N/2}^{N/2-1} (ik)^n \left(\frac{1}{N} \sum_{j=0}^{N-1} \eta(x_j, t) e^{-ikx_j} \right) e^{ikx_m} \tag{6.22}$$

Rearranging the sums, we obtain

$$\eta^{(n)}(x_m, t) \approx \frac{1}{N} \sum_{j=0}^{N-1} \eta(x_j, t) \sum_{k=-N/2}^{N/2-1} (ik)^n e^{ik(x_m - x_j)} \tag{6.23}$$

Now, for odd n we have that $(-ik)^n = -(ik)^n$, and we can use the well-known identity $e^{ix} - e^{-ix} = 2i \sin(x)$ to rewrite the inner sum as

$$\pm(-iN/2)^n + \sum_{k=1}^{N/2-1} 2i^{n+1}k^n \sin(k(x_m - x_j)) \tag{6.24}$$

Where it has been used that $e^{-iN/2(x_m-x_j)} = e^{-i\pi(m-j)} = \pm 1$. Since n is odd, i^{n+1} is real-valued, and we see that the $-N/2$ mode is the only mode contributing an imaginary part to (6.23). Hence, by truncating this Fourier mode the solution stays real, at the cost of some accuracy (which should be negligible for sufficiently smooth functions, due to the spectral accuracy). We will however still take the real part of the IDFT in the code, in case there are small rounding errors.

Note that if n is even, then $(-ik)^n = (ik)^n$, and we can instead use the identity $e^{ix} + e^{-ix} = 2\cos(x)$ to rewrite the inner sum of (6.23) as

$$\pm(-iN/2)^n + \sum_{k=1}^{N/2-1} 2(ik)^n \cos(k(x_m - x_j)) \tag{6.25}$$

which is entirely real-valued. Hence, the $-N/2$ mode only contributes imaginary parts to derivatives of odd order.

6.2 The KdV-Burgers Equation

We now move on to the main object under investigation in this thesis: The undular bore[2]. As mentioned in section 4.4, an equation modeling the undular bore should be a KdV-type equation with a dissipative term, and the KdV-Burgers equation is probably the simplest equation of this form. Hence we will start by investigating this equation. In [3], Bona and Schonbek established the existence of traveling-wave solutions to the KdV-Burgers equation in the form

$$\eta_t + \eta\eta_x + \delta\eta_{xxx} - \varepsilon\eta_{xx} = 0 \tag{6.26}$$

We will briefly summarize some of their results. Traveling-wave solutions for the KdV-Burgers equation are investigated by making the ansatz $\eta(x,t) = \phi(x - ct)$, and substituting this form into (6.26), which turns it into the ODE

$$-c\phi' + \phi\phi' + \delta\phi''' - \varepsilon\phi'' = 0 \tag{6.27}$$

[2]The term *undular bore* will specifically refer to the kind of wave shown in figure 1.1 in the introduction, where oscillations develop behind the bore front.

Where the derivatives are with respect to the variable $\xi = x - ct$. It is proven in the article that for fixed positive values of δ and ϵ, there exists a unique (up to translation in ξ), bounded solution to (6.27), such that the following limits exist:

$$\phi_L = \lim_{\xi \to -\infty} \phi(\xi) \quad , \quad \phi_R = \lim_{\xi \to \infty} \phi(\xi) \tag{6.28}$$

Furthermore, the solution satisfies the conditions

$$\lim_{|\xi| \to \infty} \phi^{(j)}(\xi) = 0 \quad , \quad j = 1, 2, 3, ... \tag{6.29}$$

Where $\phi^{(j)}(\xi)$ denotes the j-th derivative of ϕ. Additionally, the conditions $c > \phi_R$ and $\phi_L + \phi_R = 2c$ are satisfied. Moreover, the shape of the solution depends on the relative sizes of δ and ε. It is shown that if

$$\varepsilon^2 \geq 4(c - \phi_R)\delta \tag{6.30}$$

then the solution resembles a regular tanh-wave. On the other hand, $\varepsilon^2 < 4(c - \phi_R)\delta$, then oscillations will develop behind the leading wave, and the amplitude of the oscillations decrease with decreasing ξ. This is exactly the shape of an undular bore.

Finally, in the limit $\varepsilon \to 0$ the solution converges to the solitary wave solution

$$\eta(x,t) = \phi_R + 3(c - \phi_R)\,\mathrm{sech}^2\left[\left(\frac{c - \phi_R}{4\delta}\right)^{1/2}(x - x_0 - ct)\right] \tag{6.31}$$

of the KdV-equation (equation (6.26) with $\varepsilon = 0$), while in the limit $\delta \to 0$ it converges to the tanh-wave solution

$$\eta(x,t) = \phi_R + (c - \phi_R)\left(1 - \tanh\left[\frac{c - \phi_R}{2\varepsilon}(x - x_0 - ct)\right]\right) \tag{6.32}$$

of the Burgers equation (equation (6.26) with $\delta = 0$). Note that Bona and Schonbek initially made a typo in [3] when writing out the solution (6.32), but corrected it in section 5. These traveling-wave solutions to (6.26) will be sought by using the same procedure as [8], where approximate traveling-wave solutions to the Whitham equation are found by a cosine collocation method.

We will, however, consider the KdV-Burgers equation in a slightly different form than (6.26). Namely, we will start with the dimensional KdV-Burgers equation (4.36) and use the scalings

$$x \mapsto h_0 x \quad , \quad t \mapsto \frac{h_0}{c_0} t \quad , \quad \eta \mapsto \frac{2}{3} h_0 \eta \tag{6.33}$$

which turns it into

$$\eta_t + \eta_x + \eta\eta_x + \frac{1}{6}\eta_{xxx} - \varepsilon\eta_{xx} = 0 \tag{6.34}$$

where we have defined $\varepsilon \equiv \frac{\alpha}{h_0}$. The reason we will use this form of the equation is because the exact same scalings will be used on the dimensional Whitham-Burgers equation (4.38) in section 6.3, which make the two equations directly comparable. Note that by substituting the traveling-wave ansatz into (6.34), we obtain

$$(-c+1)\phi' + \phi\phi' + \frac{1}{6}\phi''' - \varepsilon\phi'' = 0 \tag{6.35}$$

This is just a special case of equation (6.27), and hence the results of [3] still apply to the equation in this form.

We can, without loss of generality, assume that $\phi_R = 0$. This is because we can employ the transformations

$$\phi \mapsto \phi + \phi_R \quad , \quad c \mapsto c + \phi_R \tag{6.36}$$

and the transformed function will still satisfy (6.35), in addition to satisfying $\phi_R = 0$. Using this assumption in combination with the conditions (6.29) ,we can integrate (6.35) from an arbitrary point x to ∞ to obtain

$$(-c+1)\phi + \frac{1}{2}\phi^2 + \frac{1}{6}\phi'' - \varepsilon\phi' = 0 \tag{6.37}$$

Hence we will now regard ϕ as a function of x. Assuming (6.37) is to be solved on the interval $[0, L]$, we should employ a scaling in order to redefine the equation on $[0, \pi]$, where it is most natural to use cosine collocation. We can use the scaling

$$\phi(x) \mapsto \phi(ax) \quad \text{with} \quad a = \frac{L}{\pi} \tag{6.38}$$

Which yields the scaled equation

$$(-c+1)\phi + \frac{1}{2}\phi^2 + \frac{1}{6}a^{-2}\phi'' - \varepsilon a^{-1}\phi' = 0 \quad , \quad 0 \le x \le \pi \tag{6.39}$$

We proceed by approximating (6.39) by the cosine collocation method. As described in section 5.5, this consists of approximating ϕ by its cosine interpolant (5.41), which we denote by ϕ_N:

$$\phi_N \equiv I_N^{DCT}\phi(x) = \sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \cos(kx) \tag{6.40}$$

Recall that this operator interpolates ϕ at the collocation points

$$x_n = \frac{(2n-1)\pi}{2N} \quad , \quad n = 1, ..., N \tag{6.41}$$

and hence these will be our grid points. By approximating ϕ by ϕ_N and enforcing (6.39) at the collocation points (6.41), we obtain the system of equations:

$$(-c+1)\phi_N + \frac{1}{2}\phi_N^2 + \frac{1}{6}a^{-2}\phi_N'' - \varepsilon a^{-1}\phi_N' = 0 \tag{6.42}$$

Where $\phi_N \equiv (\phi_N(x_1), ..., \phi_N(x_N))^T$ is to be interpreted as a vector. Note that ϕ_N^2 denotes component-wise multiplication, and the derivatives are also done component-wise. The derivative vectors ϕ_N' and ϕ_N'' can be written as matrix-vector products. By (6.40), we have

$$\begin{aligned}
\phi_N'(x_m) &= -\sum_{k=0}^{N-1} kv(k)\tilde{\phi}_k \sin(kx_m) \\
&= -\sum_{k=0}^{N-1} kv(k)\left(v(k)\sum_{n=1}^{N}\phi_N(x_n)\cos(kx_n)\right)\sin(kx_m) \\
&= -\sum_{n=1}^{N} D_1(m,n)\phi_N(x_n)
\end{aligned} \tag{6.43}$$

Where the matrix $D_1(m,n)$ is given by

$$D_1(m,n) = \sum_{k=0}^{N-1} kv^2(k)\cos(kx_n)\sin(kx_m) \tag{6.44}$$

Similarly, the second derivative may be written as

$$\begin{aligned}\phi_N''(x_m) &= -\sum_{k=0}^{N-1} k^2 v(k) \left(v(k) \sum_{n=1}^{N} \phi_N(x_n) \cos(kx_n) \right) \cos(kx_m) \\ &= -\sum_{n=1}^{N} D_2(m,n) \phi_N(x_n)\end{aligned} \tag{6.45}$$

Where

$$D_2(m,n) = \sum_{k=0}^{N-1} k^2 v^2(k) \cos(kx_n) \cos(kx_m) \tag{6.46}$$

Hence the system (6.42) can equivalently be written as

$$(-c+1)\phi_N + \frac{1}{2}\phi_N^2 + \frac{1}{6}a^{-2} D_2 \phi_N - \varepsilon a^{-1} D_1 \phi_N = 0 \tag{6.47}$$

Where $D_1\phi_N$ and $D_2\phi_N$ are matrix-vector products. This is a coupled, nonlinear system of N equations. Following [8], we solve the system by the multivariate Newton's method. If the system of equations to be solved is $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, for a function $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$, then the multivariate Newton's method is defined as the iteration:

$$\begin{cases} \mathbf{x}_0 = \text{Initial guess} \\ \text{Solve } DF(\mathbf{x}_k)\mathbf{s} = -\mathbf{F}(\mathbf{x_k}) \text{ for } \mathbf{s} \\ \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s} \end{cases} \tag{6.48}$$

Where $DF(\mathbf{x}_k)$ is the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{x}_k$. If the left-hand side of (6.47) is defined as the mapping $\mathbf{F} : \phi_N \mapsto (-c+1)\phi_N + \frac{1}{2}\phi_N^2 + \frac{1}{6}a^{-2}D_2\phi_N - \varepsilon a^{-1} D_1\phi_N$, then the Jacobian is

$$DF(\phi_N) = (-c+1)\mathbf{I} + \text{diag}(\phi_N) + \frac{1}{6}a^{-2}D_2 - \varepsilon a^{-1} D_1 \tag{6.49}$$

Where $\mathbf{I}$ is the identity matrix and $\text{diag}(\phi_N)$ is the $N \times N$ diagonal matrix whose diagonal elements are the components of ϕ_N. For initial guesses sufficiently close to a solution of (6.47), the scheme (6.48) should converge towards this solution. We will set the criterion for convergence to be when the change in ϕ_N between two iterations is less than 10^{-12} in the discrete L^∞-norm (the maximum component, in absolute value, of a vector). Note, however, that for certain combinations of c and ε, this tolerance must be raised in order to get convergence (the highest tolerance we had to use was $< 10^{-6}$).

A good way to test that the scheme works is to first seek the exact traveling-wave solutions to the Burgers and KdV equations. We will seek the traveling-wave solution to Burgers equation by solving (6.47) with the double-derivative term set to zero (which is equivalent to setting $\delta = 0$ in (6.26)) and setting $\varepsilon = 1$ and $c = 2$. Recall also that we have assumed $\phi_R = 0$. By (6.32), the exact solution is then [3]:

$$\phi(x) = 1 - \tanh\left(\frac{1}{2}(x - x_0)\right) \tag{6.50}$$

We let $L = 5\pi$ and choose $N = 256$ grid points. Experimenting with various tanh-waves as initial guesses, it is found that if the guess is sufficiently close, then the scheme always seems to converge towards (6.50) with $x_0 = L/2$ (even if the initial guess uses another value of x_0). A plot of this is shown below.

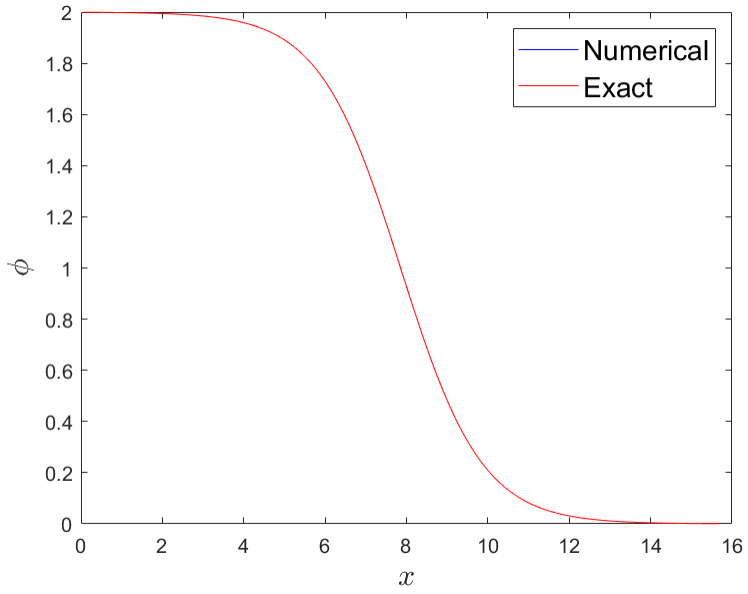


Figure 6.2: Comparison between the solution to Burgers equation found by the numerical scheme and the exact solution (6.50) with $x_0 = L/2$.

Computing the discrete L^2-error (6.21) between the numerical solution and the exact solution (6.50) with $x_0 = L/2$, we get a value of $1.0984e - 06$, providing strong evidence that the scheme indeed converges towards (6.50).

We will also seek the exact solution (6.31) of the KdV equation. To this end, we solve (6.47) with $\varepsilon = 0$ (corresponding to $\delta = \frac{1}{6}$ in the exact solution), and we choose $c = 1.5$ (corresponding to $c = 0.5$ in the exact solution). Various sech^2-waves were tried as initial

[3] Recall that x denotes the traveling-wave variable $x - ct$, and note that $c = 2$ corresponds to $c = 1$ in the exact solution, due to us using the KdV-Burgers equation in a different form.

conditions, and the results were quite interesting. If the crest of the initial wave is sufficiently close to zero, then the scheme converges towards figure 6.3a below, with the crest located at zero. For most other locations of the initial crest, however, the scheme tends to converge towards figure 6.3b, where the crest is located in the middle.

(a) (b)

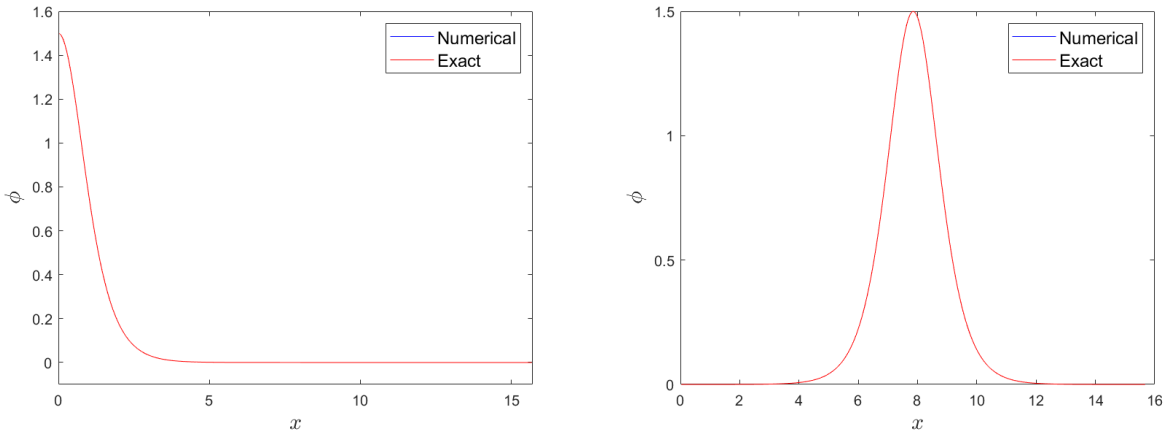


Figure 6.3: Comparison between the numerical and exact solution (6.31) to the KdV equation with $x_0 = 0$ (left figure) and with $x_0 = L/2$ (right figure)

Taking the discrete L^2-error in both of these cases, we get a value of $2.2007e-12$ for the wave with the crest at zero, which is extremely accurate. The wave with the crest in the middle is not as accurate, however, with an error of $1.770e-05$. The explanation for this probably lies in the even extension implied by the cosine collocation method. The even extension will in general not be differentiable at the boundaries, except if the derivative approaches zero at the boundaries. If the function is not differentiable at the boundaries, then we know that its Fourier series converges more slowly, and in addition, the Gibbs phenomenon will be experienced by the derivative near the boundaries. Since the solution with the crest at zero in fact has a zero derivative at the origin (and a derivative very close to zero at L), it makes sense that it should be a significantly more accurate approximation than the solution with the crest in the middle.

Next, we will solve the full KdV-Burgers equation, i.e., (6.47) with nonzero ε. We let $c = 2$, and experiment with various values of ε, and we choose various tanh-waves as initial data. It is found that the scheme converges towards either the shape of an undular bore, or to a function resembling a tanh-wave, depending on the size of ε. This is in accordance with what was proven in [3]. Below is a plot showing two of the obtained solutions.

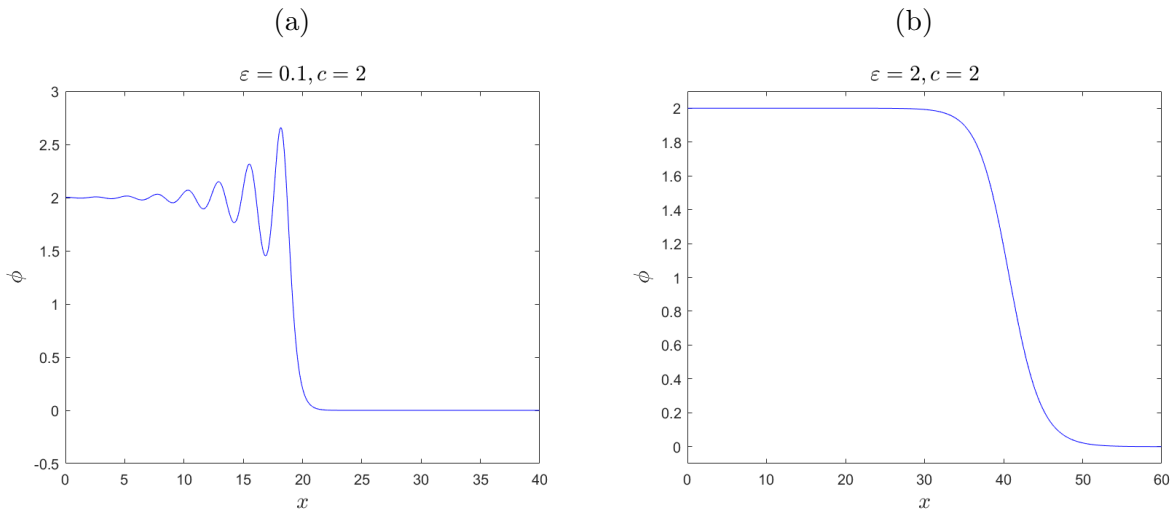


Figure 6.4: Two approximate traveling-wave solutions to the KdV-Burgers equation. In the left figure, $\varepsilon^2 < 4c\delta$, and we get oscillations behind the bore front, while in the right figure, $\varepsilon^2 \geq 4c\delta$, and there are no oscillations.

It is generally easier to obtain convergence towards a solution with oscillations (the tolerance for convergence had to be raised to 10^{-9} in order to obtain figure 6.4b). This is probably because in this case, the left boundary can coincide with one of the maxima or minima behind the bore front. This makes the derivative equal to zero on the left boundary, which as previously mentioned should make the solution more accurate. It seems to always be the case that the left boundary coincides with a minima or maxima, as shown in the plot below.

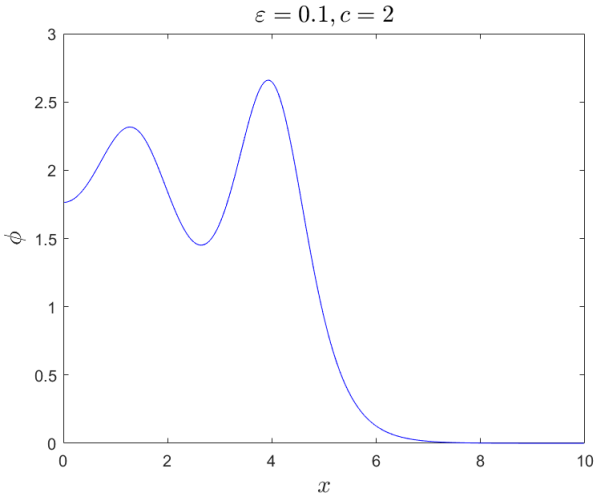


Figure 6.5: The same solution as in figure 6.4a, but calculated on a smaller interval. It seems like the code will always translate the solution such that one of its minima or maxima coincides with the left boundary.

We will also check how well the numerical solutions satisfy the condition (6.30) that governs

the existence of oscillations. Notice that due to our form of the traveling-wave equation (6.37), c should be replaced by $c-1$ in (6.30), and $\delta = \frac{1}{6}$, such that the criterion for no oscillations becomes

$$\varepsilon \geq \sqrt{\frac{2}{3}(c-1)} \tag{6.51}$$

where we have also used our assumption that $\phi_R = 0$. We test this condition by choosing some fixed values of c and solving the system (6.47) for increasing values of ε. At some point the oscillations behind the bore front should disappear. We can use MATLAB's built-in function **findpeaks** to check for oscillations. This is a function that locates all of the local maxima of an input sequence, where the criterion for a local maxima is that the point is larger than its two neighboring points (endpoints are not included). If this function does not find any peaks, we will consider the solution to be without oscillations (although we will sometimes disregard some very small peaks; see the discussion below).

The results of the experiments are shown in the table below. We have denoted the critical value of ε where the oscillations disappear by $\varepsilon_{\text{crit}}$, and the numerical values of $\varepsilon_{\text{crit}}$ obtained in the experiments are compared with the exact theoretical values.

c	ε_{crit}, numerical	ε_{crit}, exact
1.1	≈ 0.255	$\sqrt{15}/15 \approx 0.2582$
1.2	≈ 0.36	$\sqrt{30}/15 \approx 0.3651$
1.3	≈ 0.435	$\sqrt{5}/5 \approx 0.4472$
1.4	≈ 0.505	$2\sqrt{15}/15 \approx 0.5164$
2	≈ 0.805	$\sqrt{2/3} \approx 0.8165$

These computed values of ε_{crit} are admittedly quite rough estimates. As mentioned above, it is more difficult to obtain convergence to a solution without oscillations, and the length of the domain L and the tolerance for convergence of Newton's method sometimes had to be varied for different combinations of c and ε (L was anywhere between 10 and 90, and the tolerance usually around 10^{-8}). It was also sometimes the case that the solution had one extremely small oscillation, with an amplitude on the order of $\phi_L + 10^{-14}$. Such a small oscillation is not a very reliable result, and hence we disregard peaks found below a certain tolerance. It seemed natural to set the tolerance to be ϕ_L plus the final change in ϕ_N between iterations before the Newton scheme converged.

In any case, we see that the numerical values of ε_{crit} are quite close to the theoretical values, which is a good indication that the numerical solutions found are actually the solutions proposed in [3]. The numerical critical value seems to always be lower than the theoretical, which makes sense, since the oscillations should become infinitesimally small as ε approaches $\sqrt{\frac{2}{3}(c-1)}$, so at some point they should be small enough such that rounding errors dominate.

6.3 The Whitham-Burgers Equation

Next, we will investigate the possibility of the Whitham-Burgers equation having the undular bore as a traveling-wave solution. This equation was proposed in section 4.4 as a possible improvement on the KdV-Burgers equation. Recall that it is given by

$$\eta_t + \frac{3}{2}\frac{c_0}{h_0}\eta\eta_x + K_{h_0} * \eta_x - \alpha\eta_{xx} = 0 \tag{6.52}$$

Again we employ scalings in order to get the equation into a simpler form. We use the same scalings as for the KdV-Burgers equation, namely:

$$x \mapsto h_0 x \quad , \quad t \mapsto \frac{h_0}{c_0}t \quad , \quad \eta \mapsto \frac{2}{3}h_0\eta \tag{6.53}$$

The calculation of how the convolution term scales is a bit involved; see Appendix A for details. The result is

$$K_{h_0} * \eta_x \mapsto \frac{2}{3}c_0(K * \eta_x) \tag{6.54}$$

Where $K = \mathcal{F}^{-1}\left(\sqrt{\frac{\tanh(k)}{k}}\right)$ is the scaled kernel. The scalings of the other terms in (6.52) are easily calculated by the chain rule, and the result is the simpler equation:

$$\eta_t + \eta\eta_x + K * \eta_x = \varepsilon\eta_{xx} \tag{6.55}$$

Where $\varepsilon \equiv \frac{\alpha}{h_0}$.

6.3.1 The Traveling-Wave Equation

We make the traveling-wave ansatz $\eta(x,t) = \phi(x-ct)$ as before, which turns equation (6.55) into

$$-c\phi' + \frac{1}{2}(\phi^2)' + K * \phi' - \varepsilon\phi'' = 0 \tag{6.56}$$

Since we are still looking for a solution in the shape of an undular bore, we assume that the conditions (6.28) and (6.29) about the behavior at infinity still hold. Furthermore, we

will assume that $\phi_R = 0$, which is a valid assumption, as explained in section 6.2. We then integrate (6.56) from an arbitrary point x to ∞. In order to calculate the integral of the convolution term, we use the identity $K * \phi' = (K * \phi)'$ [11], which implies that

$$\int_x^\infty K * \phi' \, d\xi = \lim_{\xi \to \infty} (K * \phi)(\xi) - (K * \phi)(x) \tag{6.57}$$

It can be proven, with help of the Dominated Convergence Theorem, that

$$\lim_{\xi \to \infty} (K * \phi)(\xi) = \phi_R \tag{6.58}$$

and

$$\lim_{\xi \to -\infty} (K * \phi)(\xi) = \phi_L \tag{6.59}$$

given that ϕ is a bounded function on $\mathbb{R}$ (which is certainly true for an undular bore or tanh-wave). The details of this proof are given in Appendix B. Since we assume $\phi_R = 0$, the first term on the right-hand side of (6.57) disappears, and the result of integrating (6.56) from x to ∞ is the equation

$$-c\phi + \frac{1}{2}\phi^2 + K * \phi = \varepsilon \phi' \tag{6.60}$$

As before, we would like to find an approximate solution to (6.60) on the interval $[0, L]$, and employ the scaling (6.38) to get the equation redefined on $[0, \pi]$. The computation of the scaled convolution term follows the exact same procedure as for (6.54), so we omit the details. The result is

$$-c\phi + \frac{1}{2}\phi^2 + \sqrt{a} K_{1/a} * \phi = \varepsilon a^{-1} \phi' \quad , \quad 0 \leq x \leq \pi \tag{6.61}$$

Where we have defined $K_{1/a} \equiv \mathcal{F}^{-1}\left(\sqrt{\frac{\tanh(k/a)}{k}}\right)$. Equation (6.61) will be discretized by the cosine collocation method once again. In other words, we approximate ϕ by the cosine interpolant (6.40), and enforce (6.61) at the collocation points (6.41).

Approximating the convolution product in the equation is quite a delicate matter. It will be shown in section 6.3.2 that the convolution product of $K_{1/a}$ and ϕ_N can be written as

$$K_{1/a} * \phi_N = \sum_{k=0}^{N-1} v(k)\hat{K}_{1/a}(k)\tilde{\phi}_k \cos(kx) \tag{6.62}$$

There is however a problem with this approximation, which is that ϕ_N assumes an even periodic extension of ϕ outside $[0, \pi]$, even though ϕ itself may not be periodic. This is a problem, since the convolution product computes an integral over the entire real line, and hence we would expect $K_{1/a} * \phi_N$ to be a bad approximation of $K_{1/a} * \phi$. However, we will present a somewhat informal argument in section 6.3.2 that seems to indicate that (6.62) should still be a rather good approximation, due to the decay of K away from zero.

Using (6.62), the convolution product can be written as a matrix-vector product in the following way:

$$\begin{aligned}
(K_{1/a} * \phi_N)(x_m) &= \sum_{k=0}^{N-1} v(k)\hat{K}_{1/a}(k)\tilde{\phi}_k \cos(kx_m) \\
&= \sum_{k=0}^{N-1} v(k)\hat{K}_{1/a}(k) \left[v(k) \sum_{n=1}^{N} \phi_N(x_n) \cos(kx_n) \right] \cos(kx_m) \\
&= \sum_{n=1}^{N} K_{1/a}^N(m,n)\phi_N(x_n)
\end{aligned} \tag{6.63}$$

Where the matrix $K_{1/a}^N(m,n)$ is given by

$$K_{1/a}^N(m,n) = v^2(0)\frac{1}{\sqrt{a}} + \sum_{k=1}^{N-1} v^2(k)\sqrt{\frac{\tanh(k/a)}{k}} \cos(kx_n)\cos(kx_m) \tag{6.64}$$

where it has been used that

$$\begin{aligned}
\hat{K}_{1/a} &= \sqrt{\frac{\tanh(k/a)}{a}} \quad , \quad k \neq 0 \\
\hat{K}_{1/a}(0) &= \lim_{k\to 0} \hat{K}_{1/a}(k) = 1/\sqrt{a}
\end{aligned} \tag{6.65}$$

Hence, the system of equations to be solved is

$$-c\phi_N + \frac{1}{2}\phi_N^2 + \sqrt{a}K_{1/a}^N\phi_N - \varepsilon a^{-1} D_1 \phi_N = 0 \tag{6.66}$$

where $\phi_N \equiv (\phi_N(x_1), ..., \phi_N(x_N))^T$, and D_1 is the matrix (6.44). We use Newton's method once again to solve (6.66). The Jacobian of the left-hand side is

$$DF(\phi_N) = -c\mathbf{I} + \text{diag}(\phi_N) + \sqrt{a}K_{1/a}^N - \varepsilon a^{-1}D_1 \tag{6.67}$$

The system of equations (6.66) was solved for different values of ε, c and with various tanh-waves as initial data. The solutions found were similar to those of the KdV-Burgers equation; they have the shape of an undular bore, with oscillations behind the bore front, and the oscillations disappear if ε is large enough. Hence, it seems like the Whitham-Burgers equation admits traveling-wave solutions in the form of undular bores.

Below we have included plots of two of these approximate traveling-wave solutions to the Whitham-Burgers equation (6.55) that was found by the numerical scheme. In both cases ε is low enough such that oscillations are present. Since the exact same scalings used to obtain (6.55) was also used to obtain the scaled KdV-Burgers equation (6.34), we can directly compare solutions to these two equations. Hence for each traveling-wave solution to (6.55), we do a side-by-side comparison with the traveling-wave solution to (6.34), using the same values of ε and c.

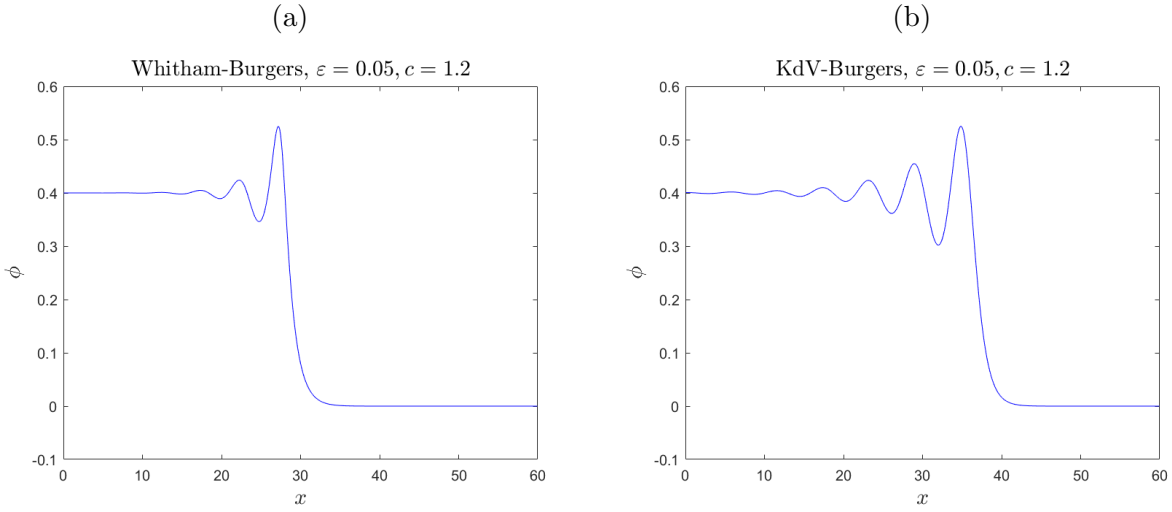


Figure 6.6: The figure on the left is an approximate traveling-wave solution to the scaled Whitham-Burgers equation (6.55) with $\varepsilon = 0.05$ and $c = 1.2$. The figure on the right is the approximate traveling-wave solution to the scaled KdV-Burgers equation (6.34) for the same values of ε and c.

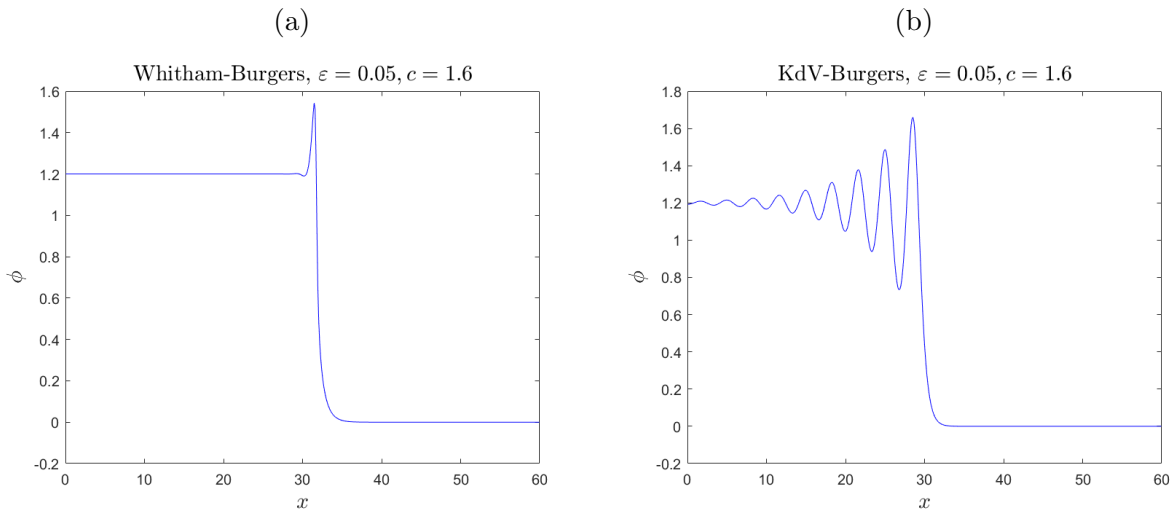


Figure 6.7: The figure on the left is an approximate traveling-wave solution to the scaled Whitham-Burgers equation (6.55) with $\varepsilon = 0.05$ and $c = 1.6$. The figure on the right is the approximate traveling-wave solution to the scaled KdV-Burgers equation (6.34) for the same values of ε and c.

From these comparisons it is clear that the solutions to the KdV-Burgers equation are more dispersive than the solutions to the Whitham-Burgers equation, in the sense that a larger selection of wavelengths are present. This makes sense, since the dispersive term of the KdV-Burgers equation is stronger.

The figures above also seem to suggest that for fixed values of c, the traveling-wave solutions to the Whitham-Burgers and KdV-Burgers equations have the same limit at negative infinity, ϕ_L (this was tested for many different values of c and seems to always be the case). Now, recalling the relationship $\phi_L + \phi_R = 2c$ that was derived in [3] for the KdV-Burgers equation, it is easily seen that this formula adjusted to our version of the KdV-Burgers equation (6.37) becomes

$$\phi_L = 2(c - 1) \tag{6.68}$$

There is in fact a way to derive the same relationship (6.68) for the Whitham-Burgers equation. If we let $x \to -\infty$ in equation (6.61), we get, using (6.28) and (6.29):

$$-c\phi_L + \frac{1}{2}\phi_L^2 + \sqrt{a} \lim_{x \to -\infty} (K_{1/a} * \phi) = 0 \tag{6.69}$$

It is easily shown that $\lim_{x \to -\infty}(K_{1/a} * \phi) = \frac{1}{\sqrt{a}}\phi_L$; this would just be a copy of the proof of (6.59), except for the fact that $\hat{K}_{1/a}(0) = \frac{1}{\sqrt{a}}$. Hence, (6.69) becomes

$$-c\phi_L + \frac{1}{2}\phi_L^2 + \phi_L = 0 \tag{6.70}$$

This has two solutions, $\phi_L = 0$ and $\phi_L = 2(c-1)$. It might be the case that the solution $\phi_L = 0$ corresponds to the trivial solution $\phi = 0$, but we were not able to prove this fact[4]. Nevertheless, it is a good sign that the relation (6.68) that is found empirically can also be derived from the traveling-wave equation. This provides evidence that the numerical solutions found are actually approximations of exact solutions to the Whitham-Burgers equation.

Next, we investigate how much dissipation is needed for the oscillations to disappear. Inspired by the criterion (6.30) of the KdV-Burgers equation, we will keep c constant and solve the system of equations (6.66) for increasing values of ε, until the point ε_{crit} is reached where the oscillations in the numerical solutions disappear. We will once again consider the solution to be without oscillations if MATLAB's function **findpeaks** does not locate any maxima above the given tolerance, which is set to be ϕ_L plus the change in ϕ_N during the last Newton iteration. Once again, the computed values of ε_{crit} are probably quite rough estimates, for the same reasons as mentioned in section 6.2.

The results are shown in the table below. We have also included the values of ε_{crit} computed for the KdV-Burgers equation in section 6.2, for comparison.

c	ε_{crit}, Whitham-Burgers	ε_{crit}, KdV-Burgers
1.1	≈ 0.265	≈ 0.255
1.2	≈ 0.395	≈ 0.36
1.3	≈ 0.5	≈ 0.435
1.4	≈ 0.6	≈ 0.505
2	≈ 1.15	≈ 0.805

Table 6.1: Relationships between c and the critical value ε_{crit} for which the oscillations to numerical solutions disappear.

This comparison suggests that a slightly higher value of ε is needed for the oscillations to disappear in the Whitham-Burgers solutions than in the KdV-Burgers solutions. See also the figure below for a specific example of a combination of c and ε where Whitham-Burgers gives oscillations but KdV-Burgers does not. This sounds a bit counter-intuitive at first, since the dispersion of Whitham-Burgers should be weaker than that of KdV-Burgers. However, this can possibly be explained by the fact that the traveling waves should be possible due to a balancing of four properties: Dispersive spreading, dissipative decay, nonlinear steepening

[4]Note that the same procedure was done in [3] in order to prove the relation $\phi_L + \phi_R = 2c$ for the KdV-Burgers equation. They also arrived at two solutions, the other one being $\phi_L = \phi_R$, and they were able to prove that this corresponded to a trivial, constant solution, by multiplying the equation by ϕ' and integrating. Attempting the same proof for the Whitham-Burgers equation would be very difficult, however, due to the convolution term.

and the "input of energy" from the left. If the dispersive term becomes weaker, then more dissipation is needed in order to balance out the nonlinear steepening and input of energy. This might be the reason for the larger value of ε_{crit} for the Whitham-Burgers equation.

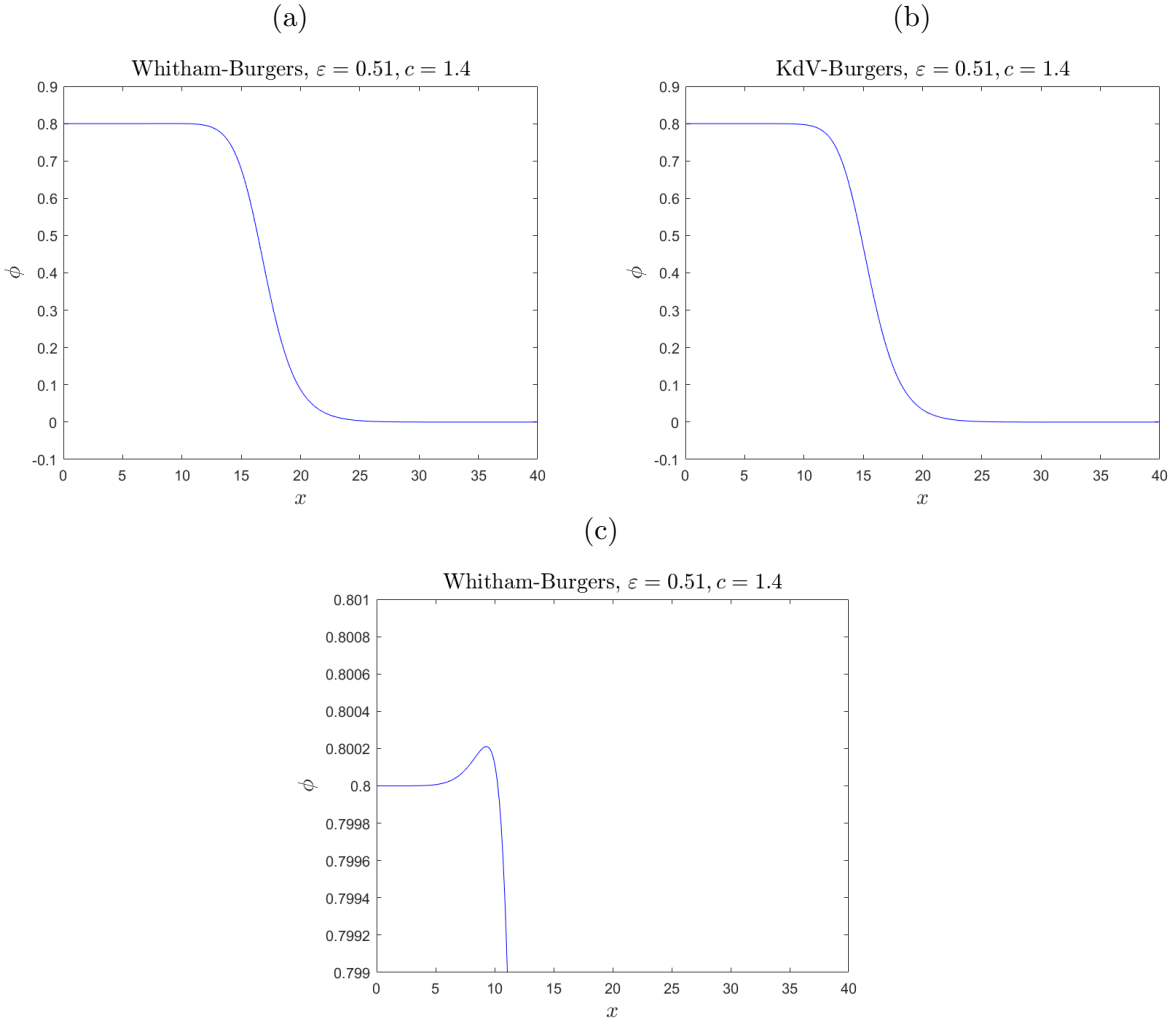


Figure 6.8: Figure (a) is a numerical solution to the Whitham-Burgers equation with $c = 1.4$ and $\varepsilon = 0.51$, and figure (b) is a numerical solution to the KdV-Burgers equation with the same values of c and ε. The Whitham-Burgers solution has a small maxima just behind the bore front (a closeup of this maxima is shown in figure (c)), while no maxima are found for the KdV-Burgers solution. Note however, that according to (6.30), the KdV-Burgers solution for this combination of c and ε should in theory have some very small oscillations.

6.3.2 Convolution With the Cosine Interpolation Operator

In this section we will prove formula (6.62) for the convolution of K with the cosine interpolant ϕ_N. The following proof can also be found in [26].

Proposition. $K_{1/a} * \phi_N = \sum_{k=0}^{N-1} v(k)\hat{K}_{1/a}(k)\tilde{\phi}_k \cos(kx)$

Proof. First of all, recall from section 4.3 that $K_{1/a} \in L^1(\mathbb{R})$. Since in addition ϕ_N is bounded on $\mathbb{R}$, the convolution product will exist for all x. [11, ch. 7] Substituting the definition of ϕ_N, we have

$$K_{1/a} * \phi_N = \int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \cos(k(x-y)) \right] dy$$

$$= \int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \frac{e^{ik(x-y)} + e^{-ik(x-y)}}{2} \right] dy$$

We can split this into two integrals to obtain

$$K_{1/a} * \phi_N = \int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \frac{e^{ik(x-y)}}{2} \right] dy + \int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \frac{e^{-ik(x-y)}}{2} \right] dy$$

On the second of these integrals, we use the substitution $y \mapsto -y$, and use the fact that $K_{1/a}$ is even[5], to rewrite the integral as

$$\int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \frac{e^{-ik(x+y)}}{2} \right] dy$$

If we then put the two integrals back together, we get

$$K_{1/a} * \phi_N = \int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \frac{e^{ik(x-y)} + e^{-ik(x+y)}}{2} \right] dy$$

$$= \sum_{k=0}^{N-1} v(k)\tilde{\phi}_k \frac{e^{ikx} + e^{-ikx}}{2} \int_{-\infty}^{\infty} K_{1/a}(y) e^{-iky} dy$$

$$= \sum_{k=0}^{N-1} v(k)\hat{K}_{1/a}(k)\tilde{\phi}_k \cos(kx)$$

[5]Since $K(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sqrt{\frac{\tanh k}{k}} e^{ikx} dk$, it is easily shown that $K(-x) = K(x)$ by making the substitution $k \mapsto -k$ and using the fact that $\sqrt{\frac{\tanh k}{k}}$ is even.

□

However, as mentioned in section 6.3.1, the problem with using $K_{1/a} * \phi_N$ as an approximation to $K_{1/a} * \phi$ is that ϕ_N assumes an even periodic extension outside $[0, \pi]$. Since the convolution is taken over the entire real line, and ϕ_N is presumably very different from ϕ outside $[0, \pi]$, due to the periodic extension, it seems like $K_{1/a} * \phi_N$ should also be very different from $K_{1/a} * \phi$.

We will try, however, to use a somewhat informal argument to show that $K_{1/a} * \phi_N$ might still be a quite accurate approximation. We will present the argument for the original, non-scaled convolution product $K * \phi_N$ defined on $[0, L]$; the scaling should not affect the accuracy of the convolution product.

As noted in [9], the kernel K has rapid decay away from zero, since the function $\sqrt{\tanh(k)/k}$ is analytic. Recall that K also has a singularity at the origin, and in [9] it is shown that the growth of K as $x \to 0$ is on the order of $x^{-1/2}$. Thus, if one also remembers that K is even, we can deduce that K should look similar to the sketch below.

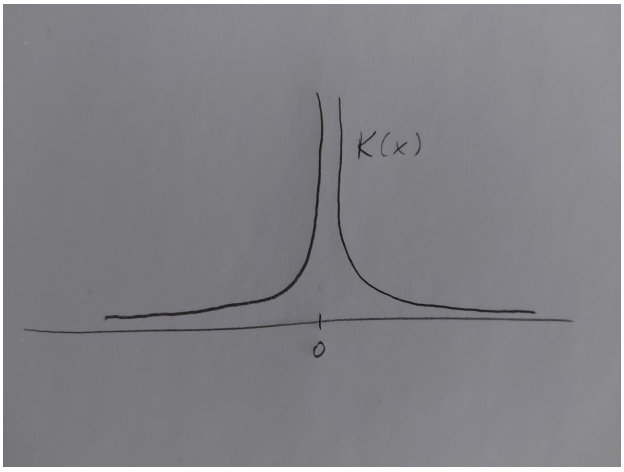


Figure 6.9: Sketch of the kernel K.

Let us consider the numerical solution obtained in figure 6.6a. The implied even extension is clearly seen if one plots this solution on a larger interval (it will be shown in the next section how to acquire the plot below):

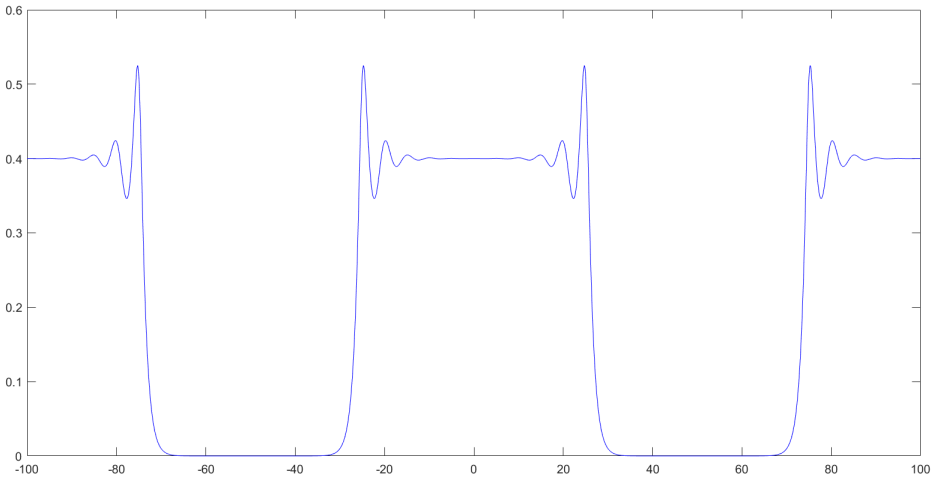


Figure 6.10: The even periodic extension of the solution from figure (6.6a), which is equivalent to ϕ_N.

The convolution product $(K * \phi_N)(x)$ at the point x is then the integral of the function in figure 6.9 multiplied with the function in figure 6.10, where K has been shifted such that the singularity is at x. Now, assuming that the Whitham-Burgers equation indeed has an exact undular-bore solution ϕ, then this solution should converge towards the limits ϕ_L and ϕ_R at negative and positive infinity, with $\phi_L > \phi_R$. We say that the solution evolves into a "plateau" for decreasing x and into a "valley" for increasing x (see the sketch below).

Then, in the interval $[0, L]$, $K * \phi_N$ should be quite close to $K * \phi$, as long as L is not too small, and a sufficient amount of the plateau and valley are contained in this interval. In that case, K should be very close to zero at the point where ϕ_N and ϕ start to significantly differ, and the contribution to the integral beyond this point should be negligibly small, both for the convolution with ϕ and with ϕ_N. A sketch of this idea is shown in the figure below, where we have chosen $x = L$.

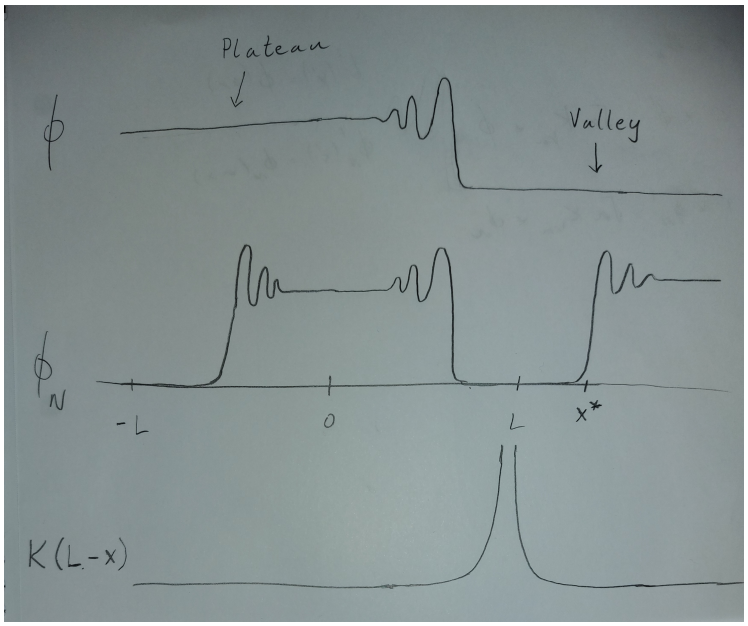


Figure 6.11: Sketch of the convolution product $(K * \phi_N)(L)$. From top to bottom: The (presumed) exact solution ϕ, the numerical solution ϕ_N with its implied even extension, and the kernel K evaluated at $L - x$. We have denoted the point where ϕ and ϕ_N start to significantly differ by x^*.

This is of course quite an informal "proof", and it would be desirable to get an actual estimate on the size of the error induced by computing $K * \phi_N$ instead of $K * \phi$, although this is likely very difficult.

6.3.3 Time-Dependent Numerical Scheme

We would like to test whether our computed solutions to (6.61) actually are approximate traveling-wave solutions to the time-dependent (scaled) Whitham-Burgers equation (6.55). Hence we will design a numerical scheme for (6.55) and use one of the computed solutions to (6.61) as the initial data. If the computed solution is indeed a traveling wave, then it should move to the right without changing its shape.

The natural choice for the spatial discretization is Fourier collocation, because as we have seen, it allows the convolution product to be written as a finite sum (a similar formula to (6.62) will be derived for regular Fourier collocation). However, there is one major problem with using this method, which is that the computed traveling-wave solutions are not periodic functions (they converge towards different limits ϕ_L and ϕ_R at negative and positive infinity). The Fourier collocation method would of course still assume a periodic extension outside the computational interval, but the discontinuities created at the boundaries would make the scheme unstable.

Ideally, we would like to design a numerical scheme for (6.55) that could handle non-periodic boundary conditions, but this would be very difficult due to the convolution term. Therefore, we will keep using Fourier collocation, and instead use a trick in order to force periodic boundary conditions. Namely, we will take one of the traveling-wave solutions and *reflect* it across the y-axis, which can conveniently be done by the cosine interpolation formula (5.41). Recall that our numerical traveling-wave solutions ϕ_N are simply (5.41) sampled at the collocation points $x_n = \frac{(2n-1)\pi}{2N}$, $n = 1, ..., N$. The numerical solution can then be reconstructed on a new grid $\{x_m\}$, $m = 1, .., M$, by evaluating (5.41) at the new grid points:

$$\phi_M(x_m) = \sum_{k=0}^{N-1} v(k)\tilde{\phi}_N(k)\cos(kx_m) \quad , \quad m = 1, ..., M \tag{6.71}$$

The algorithm for computing (6.71) is to first take the DCT of ϕ_N to obtain $\{\tilde{\phi}_N(k)\}_{k=0}^{N-1}$, and then evaluating the sum for each m. We now take advantage of the fact that the DCT assumes an even extension of the sampled function. If we use (6.71) to reconstruct the function on the interval $[-\pi, \pi]$, it will reconstruct the even extension of the function, which is equivalent to reflecting the function across the y-axis.

Our new grid will be the Fourier grid (5.12), but translated to the interval $[-\pi, \pi]$:

$$x_m = \frac{2\pi m}{M} \quad , \quad m = -M/2, ..., M/2 - 1 \tag{6.72}$$

Recall from section 5.2.2 that using Fourier collocation with these grid points is completely equivalent to Fourier collocation with the grid points $x_m = \frac{2\pi m}{M}$, $m = 0, ..., M - 1$, which are the grid points used in MATLAB's **fft** and **ifft**. We choose $M = 2N$, which seems like a natural choice since we double the spatial interval.

The result of reconstructing the solution in figure 6.6a on the new grid (6.72) is shown in the plot below. We see that the function indeed gets reflected across the y-axis. This will be our initial data when solving the time-dependent equation. Obviously, the left half of this function is not part of the traveling-wave solution, which is a major drawback of using this as the initial data. However, it can be argued that the right half should still, for some time, evolve as if a boundary condition was imposed at zero, since the long "plateau" essentially acts as a boundary condition for the right half.

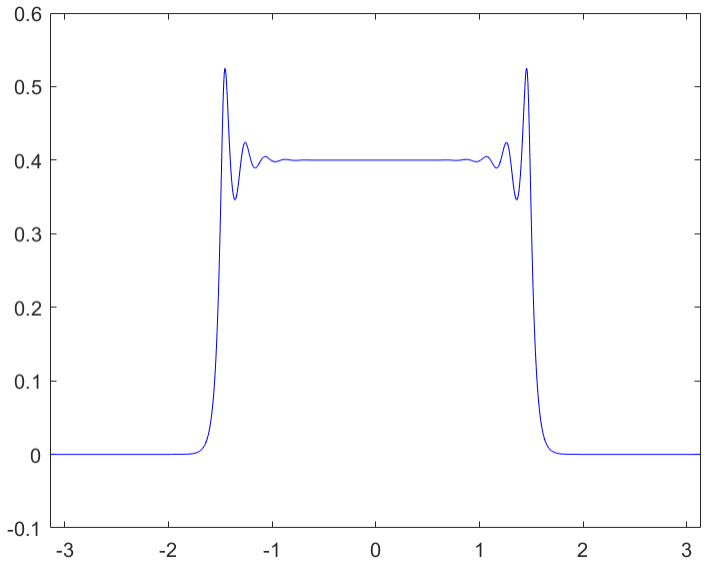


Figure 6.12: The numerical solution from figure 6.6a, reconstructed on the grid (6.72) by formula (6.71). Note that we have used $L = 80$ here, which ensures a sufficiently large plateau.

Since we doubled the size of the interval in order to reflect the function, the interval on which to solve (6.55) must also be doubled to $[-L, L]$. We then employ scalings to get (6.55) defined on $[-\pi, \pi]$. We use essentially the same scalings as for the traveling-wave equation, namely

$$x \mapsto ax \quad , \quad t \mapsto t \quad , \quad \eta \mapsto \eta \quad , \quad a = \frac{L}{\pi} \tag{6.73}$$

The scaling of the convolution term follows the same procedure as before. Equation (6.55) rescaled to $[-\pi, \pi]$ becomes

$$\eta_t + a^{-1}\eta\eta_x + a^{-1/2}K_{1/a} * \eta_x = \varepsilon a^{-2}\eta_{xx} \tag{6.74}$$

Where $K_{1/a}$ is the scaled kernel as previously defined. We approximate η by its trigonometric interpolant

$$\eta_M \equiv I_M\eta(x) = \sum_{k=-M/2}^{M/2-1} \tilde{\eta}_k(t)e^{ikx} \tag{6.75}$$

and we enforce (6.74) at the collocation points (6.72):

$$\frac{\partial \eta_M}{\partial t} + a^{-1}\frac{1}{2}\frac{\partial (\eta_M)^2}{\partial x} + a^{-1/2}K_{1/a} * \frac{\partial \eta_M}{\partial x} - \varepsilon a^{-2}\frac{\partial^2 \eta_M}{\partial x^2}\bigg|_{x=x_m} = 0 \quad , \quad m = -M/2, ..., M/2-1 \tag{6.76}$$

It will be proven in the next section that the convolution of $K_{1/a}$ and $\frac{\partial \eta_N}{\partial x}$ can be written as follows:

$$(K_{1/a} * \frac{\partial \eta_M}{\partial x})(x) = \sum_{k=-M/2}^{M/2-1} ik\tilde{\eta}_k \hat{K}_{1/a}(k) e^{ikx} \tag{6.77}$$

Recall that $\hat{K}_{1/a}$ is given by (6.65). The proof of (6.77) is very similar to that in section 6.3.2, but quite a bit easier. Once again we have the potential problem of (6.77) being an inaccurate approximation, due to the exact convolution product being with a non-periodic function. However, the arguments from section 6.3.2 should still apply here; as long as the plateau and valley of the numerical solution are large enough, then $K_{1/a}$ should be very close to zero at the point where $\partial\eta/\partial x$ and $\partial\eta_M/\partial x$ start to significantly differ. As a result, the approximation (6.77) should be sufficiently accurate for the right half of the initial data to approximately evolve like a traveling wave.

Just as with the scheme for the KdV equation, we will transform equation (6.76) into Fourier space. By differentiating η_N and taking the DFT of both sides, we obtain the equivalent system of equations

$$\tilde{\eta}_t + \frac{1}{2}ika^{-1}\widetilde{(\eta^2)} + a^{-1/2}ik\tilde{\eta}\hat{K}_{1/a} + \varepsilon a^{-2}k^2\tilde{\eta} = 0 \tag{6.78}$$

Where $k = (-M/2, ..., M/2-1)^T$, $\tilde{\eta} = (\tilde{\eta}_{-M/2}, ..., \tilde{\eta}_{M/2-1})^T$ and
$\hat{K}_{1/a} = (\hat{K}_{1/a}(-M/2), ..., \hat{K}_{1/a}(M/2-1))^T$, and all the vectors in the equation are multiplied pointwise.

For the discretization in time, the Implicit Trapezoid Method will be used once again. Applying this method to the semi-discrete equation (6.78), we obtain

$$\tilde{\eta}_{n+1} = \tilde{\eta}_n - m(\tilde{\eta}_{n+1} + \tilde{\eta}_n) - \frac{\Delta t}{4}ika^{-1}\left[\widetilde{(\eta_n^2)} + \widetilde{(\eta_{n+1}^2)}\right] \tag{6.79}$$

Where $m = \frac{\Delta t}{2}\left[\varepsilon a^{-2}k^2 + a^{-1/2}ik\hat{K}_{1/a}\right]$. This can be rearranged to

$$\tilde{\eta}_{n+1} = \frac{1-m}{1+m}\tilde{\eta}_n - \frac{\frac{\Delta t}{4}ika^{-1}}{1+m}\left[\widetilde{(\eta_n^2)} + \widetilde{(\eta_{n+1}^2)}\right] \tag{6.80}$$

This system of equations was solved by the same fixed-point iteration as for the KdV equation. The scheme was run until the time $T = \frac{100ah}{c} \approx 13.0208$, where $h = \frac{2\pi}{M}$ is the space step and c is the speed of the traveling-wave solution (the reason for this particular choice of T will be explained below). In order for the scheme to stop exactly at T, we must choose a time step Δt such that T is divisible by Δt. Hence, we choose $\Delta t = T/(2 \cdot 10^4) \approx 0.00065$. The numerical solution at T is plotted below, together with the initial data.

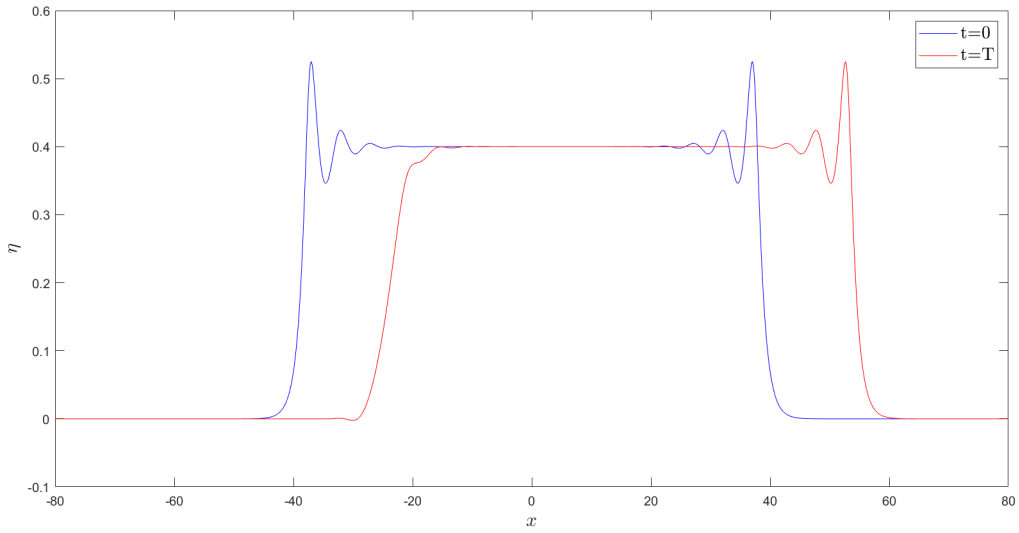


Figure 6.13: Numerical solution to the time-dependent Whitham-Burgers equation (6.55).

We see that the right half of the initial data indeed seems to evolve like a traveling wave. A close-up of the right half of the above plot is also included below.

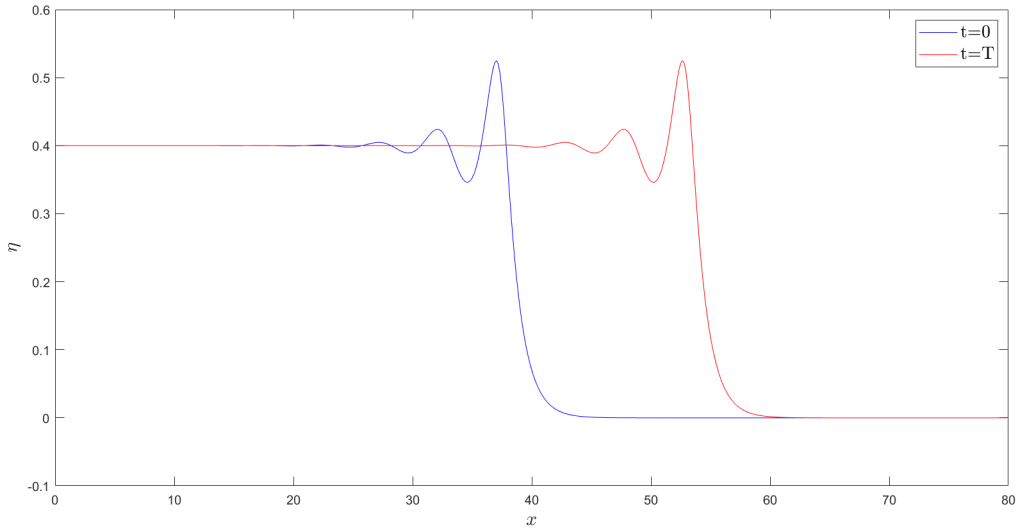


Figure 6.14: Close-up of the right half of figure 6.13.

The two shapes in figure 6.14 seem almost identical, which is a good indication that our numerical solutions are actually traveling waves. An explicit way to check the closeness of the two shapes is to translate the solution at T back to the initial data. Since the solution is supposed to travel at speed c, we can translate the function back by the amount $Tc = 100ah$, and it should more or less coincide with the initial data. The simplest way to achieve this is to define a new grid x_m^* by

$$x_m^* = ax_m - Tc \quad , \quad m = -M/2, ..., M/2 - 1 \tag{6.81}$$

where x_m is the grid (6.72) (we scale it back to $[0, L]$), and plot the solution at T on this new grid. The plot of this translated solution is shown below, together with the initial data. Note that we do not plot the spatial domain past $x^*_{M/2-1} \approx 64$. We see that the difference between the initial data and the translated solution is barely noticeable.

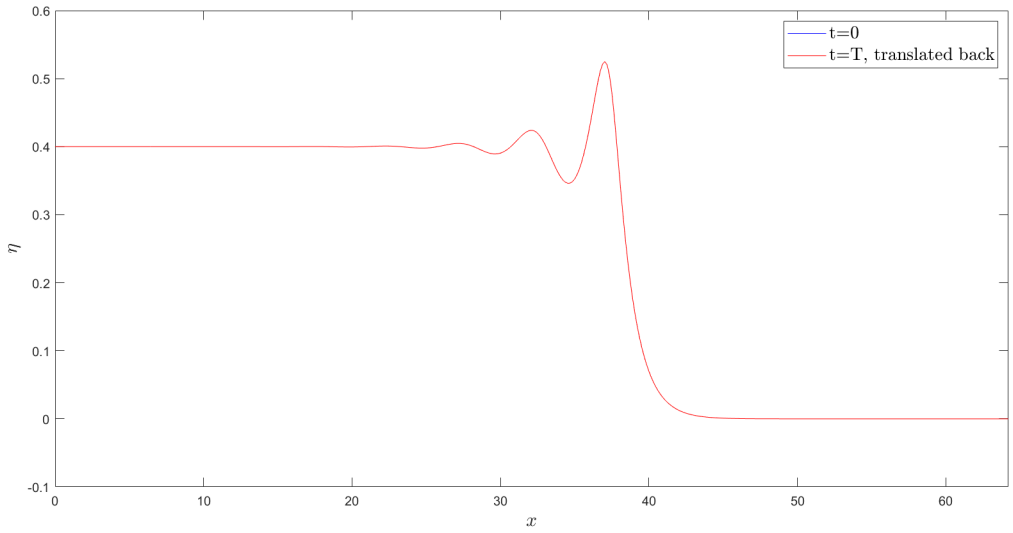


Figure 6.15: Comparison of the initial data and the solution at T translated to the left by the amount $T \cdot c$.

The closeness of the two shapes in the above plot can be quantified by taking the norm of their difference. In this case, it is important for the grid points x_m and x_m^* to match exactly in the interval where we are comparing the two functions. This is achieved when Tc is a multiple of ah, and this is the reason for our particular choice of T.

The initial data (call it η_0) and solution at T (call it η_T) are then compared at the grid points $(0 : h : x^*_{M/2-1})$, which are the grid points in $[0, L]$ that are common to both grids. Taking the discrete L^∞-error of $\eta_T - \eta_0$ at these grid points, we get a value of $1.6932 \cdot 10^{-7}$. This is a very small error, especially considering the fact that the left half of the initial data is not part of the traveling-wave solution.

6.3.4 Convolution With the Complex-Exponential Interpolation Operator

Here we will prove formula (6.77) for the convolution of K and the complex-exponential interpolation operator η_M.

Proposition. $K_{1/a} * \frac{\partial \eta_M}{\partial x} = \sum_{k=-M/2}^{M/2-1} ik\tilde{\eta}_k \hat{K}_{1/a}(k) e^{ikx}$

Proof. For simplicity, we will assume that η_M is only a function of x. This is justified by the fact that this convolution product is computed for fixed values of t. We then denote the derivative of η_M with respect to x by η'_M.

$$\begin{aligned}
(K_{1/a} * \eta'_M)(x) &= \int_{-\infty}^{\infty} K_{1/a}(y)\eta'_M(x-y)dy \\
&= \int_{-\infty}^{\infty} K_{1/a}(y) \left[\sum_{k=-M/2}^{M/2-1} ik\tilde{\eta}_k e^{ik(x-y)} \right] dy \\
&= \sum_{k=-M/2}^{M/2-1} ik\tilde{\eta}_k e^{ikx} \int_{-\infty}^{\infty} K_{1/a}(y) e^{-iky} dy \\
&= \sum_{k=-M/2}^{M/2-1} ik\tilde{\eta}_k \hat{K}_{1/a}(k) e^{ikx}
\end{aligned}$$

□

6.4 The Whitham Equation With Boundary-Layer Dissipation

Our final numerical experiment of this thesis will be of the Whitham equation with an additional term that models dissipation in a boundary layer near the channel bottom. We have not focused too much on this equation, however, since the Whitham-Burgers equation produced more interesting results. Recall from section 4.4 that Kakutani and Matsuuchi [14] derived the following equation where boundary-layer dissipation is included:

$$\eta_t + \frac{3}{2}\eta\eta_x + \frac{1}{6}\eta_{xxx} = \frac{1}{4\sqrt{\pi R^*}} (L * \eta_x) \tag{6.82}$$

Where R^* was a scaled Reynolds number and $L = \frac{1-\operatorname{sgn} x}{\sqrt{|x|}}$. Since (6.82) is just the KdV equation with an additional term, we can attempt to replace its dispersive term with the convolution term of the Whitham equation, just like we did to form the Whitham-Burgers equation. Since (6.82) is already a nondimensional and scaled equation, we will use the nondimensional form of the Whitham kernel, $K = \mathcal{F}^{-1}\left(\sqrt{\frac{\tanh(k)}{k}}\right)$, that is arrived at in appendix A. We then propose the equation:

$$\eta_t + \eta\eta_x + K * \eta_x = \rho\,(L * \eta_x) \tag{6.83}$$

Where $\rho > 0$ is some nondimensional constant. We have set all other constants to unity, which should be possible through some scaling. Equation (6.83) will be called the Whitham-Kakutani-Matsuuchi equation.

The first question one might ask of the Whitham-Kakutani-Matsuuchi equation is whether it admits traveling-wave solutions in the shape of undular bores, like the KdV-Burgers and Whitham-Burgers equations. However, this seems to not be the case. In the article by Kakutani and Matsuuchi, they conclude that equation (6.82) does not admit traveling-wave solutions with different limits at negative and positive infinity (they are called shock-like solutions in the article). They used a result by Pfirsch and Sudan [25], which states that a necessary condition for a KdV equation with dissipation to admit shock-like solutions is that

$$\lim_{k\to 0} \frac{\gamma(k)}{|k|} \to 0 \tag{6.84}$$

Where $\gamma(k)$ is the linear damping rate. It can be shown that this criterion is not satisfied by the Whitham-Kakutani-Matsuuchi equation. Recall from section 4.4 that the linear damping rate is given by the imaginary part of the dispersion relation. In order to find the dispersion relation for (6.83), note that by linearity of the convolution product, the linearized equation can be written as

$$\eta_t + (K - \rho L) * \eta_x = 0 \tag{6.85}$$

This is of the form (4.25), from which we know that the phase speed of (6.85) is given by

$$c(k) = \hat{K} - \rho\hat{L} \tag{6.86}$$

Computing the Fourier transform of the operator L is a difficult matter. We resorted to finding $\hat{L}$ by using Wolfram Alpha [33], being careful to use our convention of the Fourier transform, and the result was

$$\hat{L} = \frac{\sqrt{2\pi}(1 + i \operatorname{sgn} k)}{\sqrt{|k|}} \tag{6.87}$$

We note that $\hat{L}$ has a singularity at the origin, and it is a non-removable singularity, i.e., $\hat{L}(0)$ can not be defined in terms of its limit, as this limit does not exist. However, the product $k\hat{L}$ does have a removable singularity. We can replace $k/\sqrt{|k|}$ by $\sqrt{|k|}\operatorname{sgn} k$, as these functions are identical on $\mathbb{R}\backslash\{0\}$, and they both have the same limit as $k \to 0$. We can also replace $\frac{k \operatorname{sgn} k}{\sqrt{|k|}}$ by $\sqrt{|k|}$ for the same reasons. Therefore, the product $k\hat{L}$ can be redefined to

$$k\hat{L} \equiv |2\pi k|^{1/2}(i + \operatorname{sgn} k) \tag{6.88}$$

which is identical to $k\hat{L}$ as defined by (6.87), except the singularity at the origin has been removed. The dispersion relation of (6.85) is then

$$\omega(k) = k \cdot c(k) = k\sqrt{\frac{\tanh(k)}{k}} - \rho|2\pi k|^{1/2}(i + \operatorname{sgn} k) \tag{6.89}$$

We notice that the imaginary part is negative for all k, confirming that the term $L * \eta_x$ indeed provides a loss of energy. Moreover, the linear damping rate is given by $\gamma(k) = \rho|2\pi k|^{1/2}$, which does not satisfy the criterion (6.84). It might be the case, however, that this criterion would need to be altered for the Whitham-Kakutani-Matsuuchi equation, as the criterion is derived specifically for the KdV equation with dissipation. This was not looked any further into.

If one tries to insert the traveling-wave ansatz into (6.83) and integrate the resulting equation once, as done in sections 6.2 and 6.3, then the convolution product on the right-hand side of (6.83) becomes $L * \phi$, and now the singularity of $\hat{L}$ at the origin becomes a problem. If one attempts to solve the equation by cosine collocation, then the convolution product $L * \phi_N$ can not be computed by the proposition in section 6.3.2, because $\hat{L}(0)$ is not defined. It might be the case that L decays too slowly away from zero for the convolution product $L * \phi$ to exist. It might be possible to construct a cosine collocation scheme for the non-integrated traveling-wave equation, but this has not been attempted, due to time constraints.

It is possible, however, to solve the time-dependent Whitham-Kakutani-Matsuuchi equation (6.83) by Fourier collocation, due to the singularity of $\hat{L}$ being removable when multiplied by k. We assume to the equation is to be solved on $[-L, L]$, and employ the scaling (6.73) to redefine it on $[-\pi, \pi]$. The scaling of the convolution term $L * \eta_x$ is once again computed by the exact same procedure as shown in appendix A. The result is

$$\eta_t + a^{-1}\eta\eta_x + a^{-1/2}K_{1/a} * \eta_x - a^{-1/2}\rho L * \eta_x = 0 \tag{6.90}$$

It can be shown, using the same procedure as in section 6.3.4, that the convolution of L with the trigonometric interpolant η_N can be written as

$$L * \frac{\partial \eta_N}{\partial x} = -\sum_{k=-N/2}^{N/2-1} |2\pi k|^{1/2}(1 - i\,\mathrm{sgn}\,k)\tilde{\eta}_k e^{ikx} \tag{6.91}$$

Approximating η by η_N, we obtain the following system of equations in Fourier space (define $\varepsilon = \sqrt{2\pi}\rho$ for simplicity):

$$\tilde{\eta}_t + \frac{1}{2}ika^{-1}\widetilde{(\eta^2)} + a^{-1/2}ik\tilde{\eta}\hat{K}_{1/a} + a^{-1/2}\varepsilon|k|^{1/2}(1 - i\,\mathrm{sgn}\,k)\tilde{\eta} = 0 \tag{6.92}$$

Where $k = (-N/2, ..., N/2 - 1)^T$, $\tilde{\eta} = (\tilde{\eta}_{-N/2}, ..., \tilde{\eta}_{N/2-1})^T$ and $\hat{K}_{1/a} = (\hat{K}_{1/a}(-N/2), ..., \hat{K}_{1/a}(N/2-1))^T$ as before. Applying the Implicit Trapezoid Method in time, we obtain the equations

$$\tilde{\eta}_{n+1} = \frac{1-m}{1+m}\tilde{\eta}_n - \frac{\frac{\Delta t}{4}ika^{-1}}{1+m}\left[\widetilde{(\eta_n^2)} + \widetilde{(\eta_{n+1}^2)}\right] \tag{6.93}$$

Where $m = \frac{a^{-1/2}\Delta t}{2}\left[ik\hat{K}_{1/a} + \varepsilon|k|^{1/2}(1 - i\,\mathrm{sgn}\,k)\right]$. Equation (6.93) is solved by fixed-point iteration as usual. Since we are interested in how the equation models the development of an undular bore, we solve (6.93) with a tanh-wave as the initial data, which can be thought of as the incoming tidal bore. We choose $L = 100$, use $N = 2048$ grid points and a time step $\Delta t = 0.0005$.

Of course, the boundary conditions must still be approximately periodic, and hence we have to use a "reflected" tanh-wave as before. We use as initial data the following sum of tanh-waves

$$\eta(x, 0) = 0.05\tanh\left(\frac{1}{2}(x + 50)\right) - 0.05\tanh\left(\frac{1}{2}(x - 10)\right) \tag{6.94}$$

which is essentially a tanh-wave reflected about $x = -20$. The scheme was run with various values of ε, and compared with the solution obtained by the Whitham-Burgers scheme (6.80), using the same initial data and the same value of ε [6]. Below we have plotted the results obtained with $\varepsilon = 0.01$ and with $\varepsilon = 0.5$.

[6]Note that in the Whitham-Burgers equation, ε is the coefficient in front of the double-derivative term, while for Whitham-Kakutani-Matsuuchi, $\varepsilon = \sqrt{2\pi}\rho$. We could also have compared ε in Whitham-Burgers to ρ in Whitham-Kakutani-Matsuuchi; the conclusion below would have been the same.

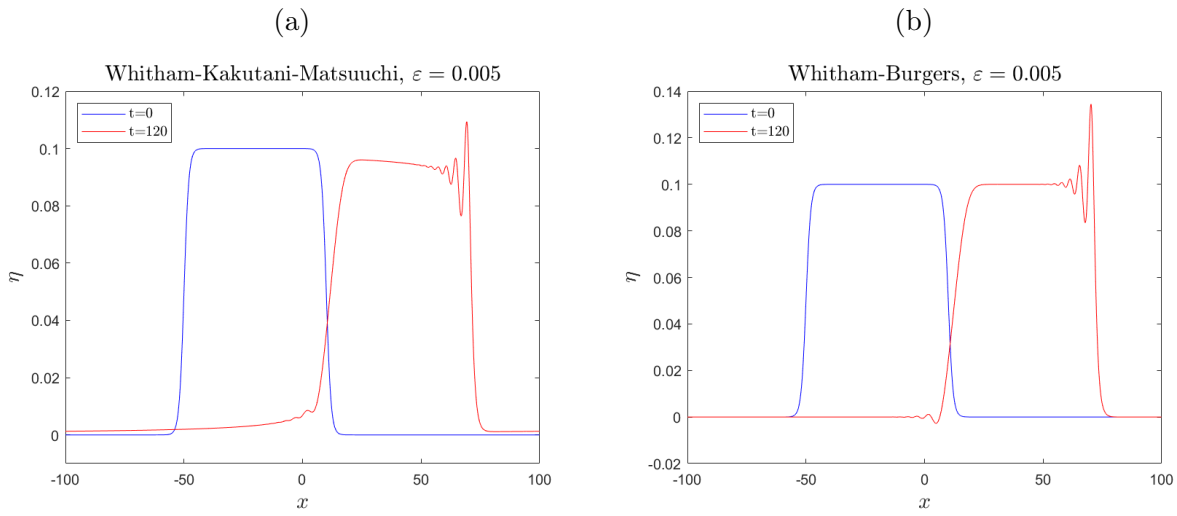


Figure 6.16: Left figure: The result of running the scheme (6.93) for the Whitham-Kakutani-Matsuuchi equation with $\varepsilon = 0.005$. Right figure: The result of running the scheme (6.80) for the Whitham-Burgers equation for the same value of ε. In both cases the schemes are run until $T = 120$.

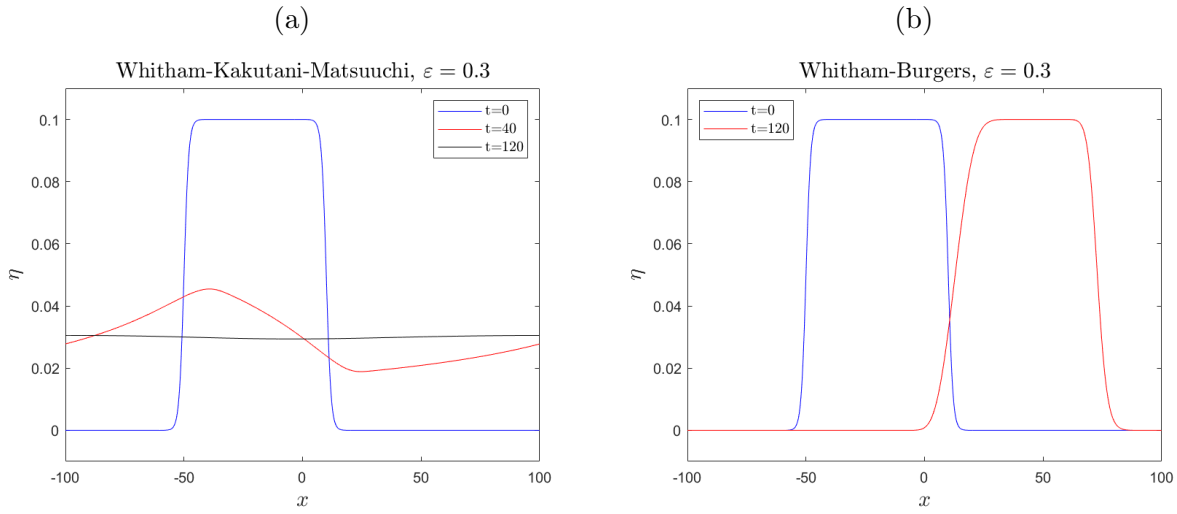


Figure 6.17: The same scenario as figure 6.16, but here dissipation is much stronger, with $\varepsilon = 0.3$.

From figure 6.16 we see that the initial wave evolves quite similarly for the two equations, with the only significant difference being the dissipation, which seems to be a lot stronger in the Whitham-Kakutani-Matsuuchi equation. This is even more apparent in 6.17, where the dissipation is completely dominant in the Whitham-Kakutani-Matsuuchi equation. This makes sense if one compares the imaginary part of the dispersion relations for the two equations, which is what gives the damping rate. For the Whitham-Burgers equation, it is proportional to k^2, while for the Whitham-Kakutani-Matsuuchi equation, it is proportional

to $|k|^{1/2}$. Recall that $k = 2\pi/\lambda$. Since our spatial interval is quite large, it should stand to reason that most of the wavelengths contained in the initial wave are larger than 2π, in which case $k < 1$ and $|k|^{1/2} > k^2$.

Finally, it should be noted that the coefficient of dissipation in a dimensional boundary-layer equation is likely to be very small, since the boundary layer is usually very thin in the case of water waves. For instance, the dimensional equation obtained by Byatt-Smith in [6] has a boundary-layer term that is proportional to $\sqrt{\nu/\pi}$, where ν is the kinematic viscosity. As mentioned in chapter 3, ν is very low for water at atmospheric pressure, and so the whole boundary-layer term should be very small in this equation. Hence, the solutions we found for the Whitham-Kakutani-Matsuuchi equation, where the dissipation was very dominant, are probably not very realistic.

Chapter 7

Conclusions and Further Work

In this thesis we have investigated three different equations that are potential models of the undular bore. We began by considering the KdV equation with an additional double-derivative term, i.e., the KdV-Burgers equation. For this equation, it had already been proven by Bona and Schonbek [3] that it admits traveling-wave solutions in the shape of undular bores. Hence, this equation served as a way to test that our numerical scheme worked, as we knew that theoretically, it should be possible to extract an undular bore from this equation. By making the traveling-wave ansatz, and solving the resulting ODE by cosine collocation, we found numerical solutions that indeed had the shape of undular bores.

Next, we generalized the KdV-Burgers equation by replacing its dispersive term with the nonlocal convolution term of the Whitham equation, and we named this new equation the Whitham-Burgers equation. We once again made the traveling-wave ansatz, and solved the resulting equation by cosine collocation. The numerical scheme found undular-bore solutions for this equation as well. These solutions were generally quite similar to the traveling-wave solutions to the KdV-Burgers equation, but the oscillating part was less dispersive. Somewhat surprisingly though, it seemed like a larger coefficient of dissipation was needed for the Whitham-Burgers equation in order for the oscillations to disappear.

Finally, we considered the Whitham equation with an additional term modelling dissipation in a boundary-layer near the bottom of the channel, which according to Sturtevant's article [29] should be a more realistic model of the undular bore. This equation was called the Whitham-Kakutani-Matsuuchi equation. However, we were not able to find any traveling-wave solutions to this equation, and a condition derived by Pfirsch and Sudan [25] might indicate that undular bores do not exist as traveling-wave solutions. We were able, however, to derive a numerical scheme for the time-dependent equation, and we ran tests comparing how this equation and the Whitham-Burgers equation model how an initial tanh-wave evolves. The effect of dissipation was much stronger in the Whitham-Kakutani-Matsuuchi equation.

Regarding possible further work, it would be very interesting if one could prove the existence

of traveling-wave solutions to the Whitham-Burgers equation, as the numerical experiments in section 6.3 seem to indicate that they do indeed exist. It would also be desirable to make the arguments in section 6.3.2 rigorous, i.e., to get an actual estimate of the error induced by taking the convolution product with ϕ_N instead of ϕ. Deriving a condition for the existence of oscillations in the Whitham-Burgers solutions, like the condition (6.30) for the KdV-Burgers equation, would also be interesting.

Another point for further study could be to model the interaction of bores. In [16] it was shown that the interaction of two solitary waves in the cubic Whitham equation[1] leaves behind a tail of oscillations after the interaction, while for the modified KdV equation[2] the interaction is completely clean. In light of these different interactions, it might be expected that the interaction of bores should produce different results for the KdV-Burgers and the Whitham-Burgers equations.

[1]The cubic Whitham equation is the Whitham equation with a slightly modified nonlinear term.

[2]The modified KdV equation is the KdV equation with the same change in the nonlinear term as in the cubic Whitham equation.

Appendix A

Scaling of the Convolution Terms

Here we provide explicit details of how the scaling of the convolution terms may be calculated. We will only show the calculation for equation (6.54); the other scalings used in equations (6.61), (6.74) and (6.90) can be proven by the exact same procedure.

Proposition. $K_{h_0} * \eta_x \mapsto \frac{2}{3} c_0 (K * \eta_x)$

Proof. It will be useful to denote the old variables by a prime. The scalings (6.33) can then be written

$$x' = h_0 x \quad , \quad t' = \frac{h_0}{c_0} t \quad , \quad \eta'(x', t') = \frac{2}{3} h_0 \eta(x, t) \tag{A.1}$$

Note also that since the wavenumber k is inversely proportional to the spatial coordinate, it scales as

$$k' = \frac{1}{h_0} k \tag{A.2}$$

By the convolution theorem, we have:

$$\begin{aligned} K_{h_0} * \eta'_{x'} &= \mathcal{F}^{-1}(\mathcal{F}(K_{h_0}) \cdot \mathcal{F}(\eta'_{x'})) \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \sqrt{\frac{g \tanh(k' h_0)}{k'}} i k' \hat{\eta}'(k', t') e^{i k' x'} dk' \\ &= \frac{c_0}{2\pi} \int_{-\infty}^{\infty} \sqrt{\frac{\tanh(k)}{k}} \frac{1}{h_0} i k \hat{\eta}'(k/h_0, t') e^{ikx} \frac{1}{h_0} dk \end{aligned} \tag{A.3}$$

Where the scalings for x and k have been used in the last equality. Note that the bounds of integration do not change. Moreover, we can use the following property of the Fourier transform:

$$\mathcal{F}[\eta(ax)] = \frac{1}{a}\hat{\eta}\left(\frac{k}{a}\right) \tag{A.4}$$

To show the following:

$$\mathcal{F}(\eta_x) = ik\mathcal{F}(\eta) = \frac{3}{2h_0}ik\mathcal{F}(\eta'(h_0x, t')) = \frac{3}{2h_0^2}ik\hat{\eta'}(k/h_0, t') \tag{A.5}$$

Substituting this into the final expression of (A.3), and using the convolution theorem again, we get

$$K_{h_0} * \eta'_{x'} = \frac{c_0}{2\pi}\frac{2}{3}\int_{-\infty}^{\infty}\sqrt{\frac{\tanh(k)}{k}}\mathcal{F}(\eta_x)e^{ikx}dk = \frac{2}{3}c_0(K * \eta_x) \tag{A.6}$$

□

Appendix B

The Limit of the Convolution

Here we will do a detailed proof of the following proposition:

Proposition. If ϕ is a bounded function on $\mathbb{R}$, then

$$\lim_{\xi \to \infty} (K * \phi)(\xi) = \phi_R \tag{B.1}$$

and

$$\lim_{\xi \to -\infty} (K * \phi)(\xi) = \phi_L \tag{B.2}$$

Proof. We will need the Dominated Convergence Theorem in order to prove this proposition. This theorem can be found in chapter 3 of [11], and is stated below (in the special case of functions of one variable).

Theorem B.0.1. *Let D be a region in $\mathbb{R}$. Suppose f_n $(n = 1, 2, 3, ...)$, f and g are functions on D, such that*

(a) $g(x) \geq 0$ and $\int_D g(x)dx < \infty$

(b) $|f_n(x)| \leq g(x)$ for all n and all $x \in D$

(c) $f_n(x) \to f(x)$ as $n \to \infty$ for all $x \in D$

Then $\int_D f_n(x)dx \to \int_D f(x)dx$ as $n \to \infty$.

We will show the proof in detail for the right-hand limit (B.1). In this case, $D = \mathbb{R}$, and the sequence of functions to consider is $f_n(y) = K(y)\phi(n - y)$. It is clear that

$$\lim_{n \to \infty} f_n(y) = \phi_R K(y) \quad \forall\ y \in \mathbb{R} \tag{B.3}$$

A slight caveat is that we consider $f_n(y)$ to be a continuous sequence of functions (so the limit as $n \to \infty$ is a continuous limit), whereas the Dominated Convergence Theorem assumes a discrete sequence of functions. However, we can just define some sequence $\{p_n\}$ such that $\lim_{n\to\infty} p_n = \infty$, and consider the discrete sequence of functions $f_{p_n}(y) = K(y)\phi(p_n - y)$. This sequence clearly still satisfies the limit condition (B.3).

We have that

$$|f_{p_n}(y)| = |K(y)||\phi(p_n - y)| \leq C|K(y)| \quad \forall\ n \quad \forall\ y$$

Where $C = \sup_{y\in\mathbb{R}}|\phi(y)|$. Hence condition (b) of the theorem is satisfied with $g(y) = C|K(y)|$. Moreover, $g(y) \geq 0$ and

$$\int_{-\infty}^{\infty} g(y)dy = C\int_{-\infty}^{\infty} |K(y)|dy < \infty$$

since $K \in L^1(\mathbb{R})$. Thus all conditions of the theorem are satisfied, and we get that

$$\lim_{n\to\infty} \int_{-\infty}^{\infty} K(y)\phi(p_n - y)dy = \phi_R \int_{-\infty}^{\infty} K(y)dy = \phi_R$$

Where it has been used that $\int_{-\infty}^{\infty} K(y)dy = \hat{K}(0) = \lim_{k\to 0}\sqrt{\tanh(k)/k} = 1$. Since this limit holds for any sequence $\{p_n\}$ going to infinity, it must also hold in the continuous limit to infinity, and we get

$$\lim_{\xi\to\infty}(K * \phi)(\xi) = \lim_{\xi\to\infty}\int_{-\infty}^{\infty} K(y)\phi(\xi - y)dy = \phi_R$$

The proof that $\lim_{\xi\to-\infty}(K*\phi)(\xi) = \phi_L$ is identical, except you define $f_n(y) = K(y)\phi(-n-y)$, and use that $\lim_{\xi\to-\infty}\int_{-\infty}^{\infty} K(y)\phi(\xi - y)dy = \lim_{\xi\to\infty}\int_{-\infty}^{\infty} K(y)\phi(-\xi - y)dy$.

□

Appendix C

Source Code

Here we have included two of the scripts that were used in chapter 6 of this thesis. The first script computes the approximate traveling-wave solutions that were found for the KdV-Burgers and Whitham-Burgers equations. The second script solves the time-dependent Whitham-Burgers equation numerically. The other scripts used in section 6.1 and 6.4 are very similar to the second script, but with minor changes. These scripts were inspired by existing codes made by Henrik Kalisch, and his codes have been further developed in [15].

```
% This program computes approximate traveling-wave solutions to the
% KdV-Burgers equation or the Whitham-Burgers equation by cosine ...
    collocation.
%
% KdV-Burgers traveling-wave equation:
%(-c+1)*u + 1/2*u^2 + 1/6*a^(-2)*u'' - epsilon*a^(-1)*u' = 0 on [0,pi]
%
% Whitham-Burgers traveling-wave equation:
% -c*u + 1/2*u^2 + a^(1/2)(K_1/a * u) - epsilon*a^(-1)*u' = 0 on [0,pi]
%
% We solve the equations on [0,L], but use the scaling u(x) -> u(ax),
% a=L/pi to redefine the equations on [0,pi], the interval on which the
% solutions are actually computed by the code.

clear

%%
%%%Basic parameters

L=60; %Spatial interval on which to solve the equation
N=512; %Number of grid points
h0=L/N; %Space step
x0 = (h0/2:h0:L-h0/2)'; %Cosine grid

c=1.2; %Speed of the traveling wave
uinit=0.2*(1-tanh(x0-L/2)); %Initial guess for the traveling-wave solution
```

```
26
27 %%
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 %%%Cosine collocation
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31
32 %First we scale the grid down to the interval [0,pi]
33
34 h=pi/N; %New space step on [0,pi]
35 a=L/pi; %Scaling parameter
36 x = (h/2:h:pi-h/2)'; %Spatial grid scaled down to [0,pi]
37
38 wav=(0:1:N-1)'; %Wavenumbers for cosine collocation
39
40 v = sqrt(2/N)*ones(N,1); %This is the weight used for the discrete ...
       cosine coefficients
41 v(1) = sqrt(1/N);
42
43 %First-derivative matrix
44 D1=zeros(N);
45 for m=1:N;
46    for n=1:N;
47       D1(m,n) = 0;
48       for k=2:N;
49       D1(m,n) = D1(m,n) + ...
             v(k)*v(k)*(wav(k))*cos(x(n)*wav(k))*sin(x(m)*wav(k));
50       end;
51    end;
52 end;
53 D1=-D1;
54
55 %Second-derivative matrix
56 D2=zeros(N);
57 for m=1:N;
58    for n=1:N;
59       D2(m,n) = 0;
60       for k=2:N;
61       D2(m,n) = D2(m,n) + ...
             v(k)*v(k)*(wav(k))^2*cos(x(n)*wav(k))*cos(x(m)*wav(k));
62       end;
63    end;
64 end;
65 D2=-D2;
66
67 %Convolution product matrix
68 K=zeros(N);
69 for m=1:N;
70    for n=1:N;
71       K(m,n) = sqrt(1/a)*v(1)*v(1)*cos(x(n)*wav(1))*cos(x(m)*wav(1));
72       for k=2:N;
73       K(m,n) = K(m,n) + sqrt((1/wav(k))*tanh((1/a)*wav(k)))*...
             v(k)*v(k)*cos(x(n)*wav(k))*cos(x(m)*wav(k));
74       end;
75    end;
76
```

```
end;

Δ=1/6; %Dispersion coefficient

epsilon=0.05; %Dissipation coefficient

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Below are the system of equations to be solved by Newton's method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Eqns= (-c+1)*eye(N)+Δ*a^(-2)*D2-epsilon*a^(-1)*D1; %System of ...
    equations for the KdV-Burgers equation

Eqns= -c*eye(N)+sqrt(a)*K-epsilon*a^(-1)*D1; %System of equations for ...
    the Whitham-Burgers equation

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Newton loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u=uinit;

tolerance=1e-12; %Tolerance for convergence of Newton's method

change = 1;
while change > tolerance
    DFu = Eqns + diag(u); %Jacobian matrix
    corr = - DFu\( Eqns*u + 1/2*u.^2); %Corrective term
    unew = u + corr; %New guess
    change=norm(corr,inf); %Norm of difference between old and new guess
    u = unew;
end

plot(x0,u,'b')
```

```
% This program solves the time-dependent Whitham-Burgers equation
% numerically by the Fourier Collocation Method in space, and the Implicit
% Trapezoid Method with fixed-point iteration in time.
%
% Time-dependent Whitham-Burgers equation:
% u_t+a^(-1)uu_x+a^(-1/2)(K_1/a * u_x)-epsilon*a^(-2)*u_xx=0 on [-pi,pi]
%
% We solve the equation on [-L,L], but use the scaling u(x) -> u(ax),
% a=L/pi to redefine the equation on [-pi,pi], the interval on which the
% solution is actually computed by the code.
%
% We set the initial data to be an approximate traveling wave for the
% Whitham-Burgers equation, found on [0,pi] by the script "travwave.m", and
% we reconstruct its even extension to [-pi,pi], which ensures periodic
% boundary conditions, at the cost of the left half of the initial data not
% being part of the traveling-wave solution.
```

```
17
18 clear
19
20 travwave; %Find an approximate traveling-wave solution on [0,pi]
21
22 %%
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 %%%Interpolation to [-pi,pi]
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26
27 M=2*N; %Double the number of grid points
28 h=2*pi/M; %New space step
29 x=(-pi:h:pi-h); %Fourier grid from -pi to pi
30
31 dctu=dct(u); %Take the discrete cosine transform of u
32
33 %Use the cosine interpolation formula to reconstruct the traveling-wave
34 %solution on [-pi,pi]
35 for m=1:M
36     u_recon(m) =0;
37     for n=1:N
38         u_recon(m) = u_recon(m) + v(n)*dctu(n)*cos(x(m)*wav(n));
39     end
40 end
41
42 %%
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44 %%%Time-dependent scheme
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46
47 dt=0.0016; %Time step
48 T=100*a*h/c; %Run scheme until this time
49
50 uinit=u_recon; %Set initial data to be the traveling-wave reconstructed ...
    on [-pi,pi]
51
52 k = [1:M/2 -M/2+1:-1]; %Wavenumbers used in Fourier collocation, the ...
    k=0 mode will be filled in below
53
54 %%
55 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
56 %%%Fourier collocation scheme with Implicit Trapezoid in time
57 %%%The resulting system of equations is solved by fixed-point iteration
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59
60 m=0.5*dt*(a^(-1/2)*1i*k.*sqrt(tanh(k/a)./k)+epsilon*a^(-2)*k.^2);
61 k=[0 k]; %Fill in k=0 after defining m to avoid dividing by zero
62 m=[0 m]; %Fill in the limit of m as k goes to zero
63 k(M/2+1)=0; %Avoid asymmetry in the wave numbers
64 m(M/2+1)=0;
65
66 c1=(1-m)./(1+m);
67 c2=a^(-1)*0.25*dt*1i*k./(1+m);
68
```

```
69 t=0;
70 u=uinit;
71
72 for i=1:T/dt
73    fftu=fft(u); %Discrete fourier transform of u
74    fftuu=fft(u.^2); %Discrete fourier transform of u^2 (nonlinear term)
75    w=real(ifft(c1.*fftu-2*c2.*fftuu)); %Initial guess for the ...
          fixed-point iteration
76    tol=1;
77    while tol>1e-12 %Fixed-point iteration
78       w2=real(ifft(c1.*fftu-c2.*(fftuu+fft(w.^2))));
79       tol=norm(w-w2);
80       w=w2;
81    end
82    u=w;
83    t=t+dt;
84 end
85
86 plot(a*x,uinit,a*x,u)
```

Bibliography

[1] Alfatih Ali and Henrik Kalisch. "Energy balance for undular bores". In: *Comptes Rendus Mécanique* 338 (Feb. 2010), pp. 67–70.

[2] T. B. Benjamin and M. J. Lighthill. "On Cnoidal Waves and Bores". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 224.1159 (1954), pp. 448–460.

[3] J. L. Bona and M. E. Schonbek. "Travelling-wave solutions to the Korteweg-de Vries-Burgers equation". In: *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* (1985), pp. 207–226.

[4] William L. Briggs and Van Emden Henson. *The DFT: An Owner's Manual for the Discrete Fourier Transform*. Society for Industrial and Applied Mathematics, 1995.

[5] Mats K Brun and Henrik Kalisch. "Convective wave breaking in the KdV equation". In: *Analysis and mathematical physics* 8 (2017), pp. 57–75. ISSN: 1664-2368.

[6] J. G. B. Byatt-Smith. "The effect of laminar viscosity on the solution of the undular bore". In: *Journal of Fluid Mechanics* 48.1 (1971), pp. 33–40.

[7] C. Canuto et al. *Spectral methods in fluid dynamics*. Springer series in computational physics. New York, 1988.

[8] Mats Ehrnström and Henrik Kalisch. "Global Bifurcation for the Whitham Equation". In: *Mathematical Modelling of Natural Phenomena* 8 (2013).

[9] Mats Ehrnström and Henrik Kalisch. "Traveling waves for the Whitham equation". In: *Differential and Integral Equations* 22 (2009), pp. 1193–1210.

[10] H. Favre. *Étude théorique et expérimentale des ondes de translation dans les canaux découverts*. Dunod, 1935.

[11] Gerald B Folland. *Fourier Analysis and Its Applications*. Providence: American Mathematical Society, 2009.

[12] C Gasquet. *Fourier analysis and applications : filtering, numerical computation, wavelets*. New York, 1999.

[13] R. S. Johnson. *A Modern Introduction to the Mathematical Theory of Water Waves*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1997.

[14] Tsunehiko Kakutani and Kazuo Matsuuchi. "Effect of Viscosity on Long Gravity Waves". In: *Journal of the Physical Society of Japan* 39.1 (1975), pp. 237–246.

[15] Henrik Kalisch, Daulet Moldabayev, and Olivier Verdier. "Numerical Study of Nonlinear Dispersive Wave Models with SpecTraVVave". In: (2016). URL: https://arxiv.org/abs/1606.01465.

[16] Henrik Kalisch et al. *Breather Solutions to the Cubic Whitham Equation*. 2022. URL: https://arxiv.org/abs/2201.12074.

[17] Pijush K. Kundu, Ira M. Cohen, and David R. Dowling. *Fluid Mechanics (Sixth Edition)*. Boston: Academic Press, 2016. ISBN: 978-0-12-405935-1.

[18] R Lemoine. "Sur les ondes positives de translation dans les canaux et sur le ressaut ondulé de faible amplitude". In: *Houille Blanche* 2 (1948), pp. 183–185.

[19] Y. Li and D. H. Sattinger. "Matlab Codes for Nonlinear Dispersive Wave Equations". In: (1998).

[20] *MATLAB dct - Discrete cosine transform - MathWorks*. URL: https://se.mathworks.com/help/signal/ref/dct.html (visited on 05/04/2022).

[21] *MATLAB fft - Fast Fourier transform - MathWorks*. URL: https://se.mathworks.com/help/matlab/ref/fft.html (visited on 05/02/2022).

[22] *MATLAB idct - Inverse discrete cosine transform - MathWorks*. URL: https://se.mathworks.com/help/signal/ref/idct.html (visited on 05/04/2022).

[23] Daulet Moldabayev, Henrik Kalisch, and Denys Dutykh. "The Whitham Equation as a model for surface water waves". In: *Physica D: Nonlinear Phenomena* 309 (2015), pp. 99–107.

[24] Pavel I. Naumkin and Il'ya Andreevich Shishmarev. *Nonlinear Nonlocal Equations in the Theory of Waves*. American Mathematical Society, 1994.

[25] D. Pfirsch and R. N. Sudan. "Conditions for the existence of shock-like solutions of Korteweg-de Vries equation with dissipation". In: *Physics of Fluids* 14.5 (1971), p. 1033.

[26] Filippo Remonato and Henrik Kalisch. "Numerical bifurcation for the capillary Whitham equation". In: *Physica D: Nonlinear Phenomena* 343 (2017), pp. 51–62.

[27] Tim Sauer. *Numerical analysis*. 2nd ed. Pearson, 2014.

[28] Jan Ole Skogestad and Henrik Kalisch. "A boundary value problem for the KdV equation: Comparison of finite-difference and Chebyshev methods". In: *Mathematics and Computers in Simulation* 80.1 (2009), pp. 151–163.

[29] B. Sturtevant. "Implications of Experiments on the Weak Undular Bore". In: *The Physics of Fluids* 8.6 (1965), pp. 1052–1055.

[30] Lloyd N. Trefethen. *Spectral methods in MATLAB*. SIAM, 2008.

[31] G. B Whitham. *Linear and nonlinear waves*. Pure and applied mathematics (Wiley). New York, 1974. ISBN: 0471940909.

[32] G. B. Whitham. "Variational Methods and Applications to Water Waves". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 299.1456 (1967), pp. 6–25. ISSN: 00804630. URL: http://www.jstor.org/stable/2415780.

[33] *Wolfram—Alpha: Computational Intelligence*. URL: https://www.wolframalpha.com/ (visited on 05/20/2022).