Numerical methods for nonlinear nonlocal water wave models

Daulet Moldabayev

Dissertation for the degree of Philosophiae Doctor (PhD)

Department of Mathematics
University of Bergen
Norway

April 2017

Preface

This dissertation is submitted as a partial fulfillment of the requirements for the degree Doctor of Philosophy (PhD) at the Department of Mathematics, University of Bergen. The research was supported by the Research Council of Norway on grant no. 213474/F20.

The advisory committee has consisted of Henrik Kalisch (University of Bergen, Bergen, Norway), Dmitry Pelinovsky (McMaster University, Hamilton, Ontario, Canada) and Angel Durán (Universidad de Valladolid, Valladolid, Spain).

Acknowledgements

First of all I would like to thank Henrik Kalisch and Magnus Svärd who carried out the selection process of candidates for the PhD position. I am sincerely thankful to Professor Henrik Kalisch, my main supervisor, for sharing knowledge and guiding me, for his enthusiasm, support and encouragement. The thesis would not have been possible without you.

In the course of the doctorate studies, I have had fruitful collaboration with Olivier Verdier, Evgueni Dinvay, Denys Dutykh, Ben Segal and others. I would like to thank them for productive discussions, advices on research questions and some inspiring ideas we came up with.

As a PhD Candidate I have been lucky to meet my colleagues and fellow students at the Department of Mathematics. They have always provided a friendly and supportive environment. I appreciate everything we have experienced during the time of our studies. I am also thankful to the administration of the department for organising

I would like to express my gratitude to my family for their continuous support and encouragement over the years of my studies. I am also thankful for all of my teachers and professors from schools and universities I studied at.

I am heartily grateful for my wife Assem. You have always been there for me. Your love, support and understanding are the source of my strength and motivation. You helped me to go this far.

Outline

This thesis is organised in the following way. Part I contains general theoretical background on nonlinear wave models as well as description of methods used to solve the equations involved. Some properties of solutions to the equations and a summary of results are also given in the first part. Part 2 consists of the research papers that present scientific results in detail.

List of reseach papers included in Part II

Paper A:
Moldabayev, D., Kalisch, H., Dutykh, D.: *The Whitham equation as a model for surface water waves*, Phys. D **309**, 99–107 (2015),
http://dx.doi.org/10.1016/j.physd.2015.07.010.

Paper B:
Dinvay, E., Moldabayev, D., Dutykh, D., Kalisch, H.: *The Whitham equation with surface tension*, Nonlinear Dynamics, 1–14 (2017),
http://dx.doi.org/10.1007/s11071-016-3299-7.

Paper C:
Henrik Kalisch, Daulet Moldabayev, Olivier Verdier: *A numerical study of nonlinear dispersive wave models with SpecTraVVave*, specify status of the paper.

Paper D:
Benjamin Segal, Daulet Moldabayev, Henrik Kalisch, Bernard Deconinck: *Explicit solutions for a long-wave model with constant vorticity*, submitted to European Journal of Mechanics - B/Fluids.

Contents

Part I

Background

Chapter 1

Introduction

This is the introduction [**?**]. . .

Chapter 2

Summary of results

This chapter provides an overview of the results achieved in the course of research work. Detailed

2.1 A numerical study of nonlinear dispersive wave models with SpecTraVVave

In nonlinear dispersive evolution equations, the competing effects of nonlinearity and dispersion make a number of interesting phenomena possible. In the current work, the focus is on the numerical approximation of traveling-wave solutions of such equations. We describe our efforts to write a dedicated Python code which is able to compute traveling-wave solutions of nonlinear dispersive equations in a very general form.

The SpecTraVVave code uses a continuation method coupled with a spectral projection to compute approximations of steady symmetric solutions of this equation. The code is used in a number of situations to gain an understanding of traveling-wave solutions. The first case is the Whitham equation, where numerical evidence points to the conclusion that the main bifurcation branch features three distinct points of interest, namely a turning point, a point of stability inversion, and a terminal point which corresponds to a cusped wave.

The second case is the so-called modified Benjamin–Ono equation where the interaction of two solitary waves is investigated. It is found that two solitary waves may interact in such a way that the smaller wave is annihilated. The third case concerns the Benjamin equation which features two competing dispersive operators. In this case, it is found that bifurcation curves of periodic traveling-wave solutions may cross and connect high up on the branch in the nonlinear regime.

2.1.1 Introduction

This paper is concerned with traveling wave solutions for a class of nonlinear dispersive equations of the form

$$u_t + [f(u)]_x + \mathcal{L}u_x = 0, \tag{1.1}$$

where $\mathcal{L}$ is a self-adjoint operator, and f is a real-valued function with $f(0) = 0$ and $f'(0) = 0$, and which satisfies certain growth conditions. Equations of this form arise routinely in the study of wave problems in fluid mechanics and many other contexts.

A prototype of such an equation is the KdV equation that appears if $\mathcal{L} = I + \frac{1}{6}\partial_x^2$ and $f(u) = \frac{3}{4}u^2$. In the current work, the operator $\mathcal{L}$ is considered to be given as a Fourier multiplier operator, such as for instance in the Benjamin–Ono equation, which arises in the study of interfacial waves. In this case, the Fourier multiplier operator is given by $\mathcal{L} = I - \mathcal{H}\partial_x$, where the Hilbert transform $\mathcal{H}$ is defined as

$$\mathcal{H}u(x) = \frac{1}{\pi}\,\text{p.v.}\int_{-\infty}^{\infty} \frac{u(x-y)}{y}\,\mathrm{d}y, \qquad \widehat{\mathcal{H}u}(k) = -\mathrm{i}\,\mathrm{sgn}(k)\widehat{u}(k). \tag{1.2}$$

We also study in detail traveling wave solutions of the Whitham equation, which appears when $\mathcal{L}$ is given by convolution with the integral kernel K_{h_0} in the form

$$\mathcal{L}u(x) = \int_{-\infty}^{\infty} K_{h_0}(y)u(x-y)\,\mathrm{d}y, \qquad \widehat{K_{h_0}}(k) = \sqrt{\tfrac{g\tanh(h_0 k)}{k}}, \tag{1.3}$$

and f is the same function as in the KdV equation.

The particular form of equation (2.1) exhibits the competing effects of dispersion and nonlinearity, which gives rise to a host of interesting phenomena. The most well known special phenomenon is the existence of solitary waves and of periodic traveling waves containing higher Fourier modes. Indeed, note that in the purely dispersive model $u_t + \mathcal{L}u_x = 0$, the only possible permanent progressive waves are simple sinusoidal waves, while in the nonlinear model (2.1) higher Fourier modes must be considered to obtain solutions.

The order of the operator $\mathcal{L}$ appearing in (2.1) has a major effect on the types of solutions that may be found. A higher-order operator, such as in the Korteweg–de Vries equation, acts as a smoothing operator because of its effect of spreading different frequency components out due to a strongly varying phase speed [**?**]. Lower-order operators such as the operator K_{h_0} in (2.3) appearing in the Whitham equation may allow solutions to develop singularities, such as derivative blow-up (see [**?**], [**?**]) and formation of cusps (see [**?**]).

On the other hand, highly nonlinear functions $f(u)$ may lead to L^∞-blow-up. For instance, the generalized KdV equation which is written in normalized form as

$$u_t + u^p u_x + u_x + u_{xxx} = 0, \tag{1.4}$$

features global existence of solutions for $p = 1,2,3$, but the solutions blow-up in the critical case $p = 4$ (the case $p \geq 5$ is open). In the case of the generalized Benjamin–Ono equation

$$u_t + u^p u_x + u_x - \mathcal{H}u_{xx} = 0,$$

where $\mathcal{H}$ is the Hilbert transform, numerical evidence points to singularity formation for $p > 2$ [**?**], but no proofs are available at this time.

In order to study different phenomena related to equations of the form (2.1) and their traveling wave solutions, a Python-based solver package SpecTraVVave was developed by the authors [**?**]. The general idea behind the solver is to use a numerical continuation method [**?**] implemented with a pseudo-spectral algorithm. Similar previous projects include AUTO [**?**] and Wavetrain [**?**]. AUTO is written in C, whereas Wavetrain is written in *Fortran*. Both programs are efficient and very general, as they are able to cover a wide range of problems involving bifurcation analyses. However, from a user's perspective, such a generality coupled with low level programming languages may lead to some difficulties in utilizing these programs efficiently.

SpecTraVVave is designed to provide researchers with a simple yet effective tool for investigating problems on traveling waves. The package is flexible, and its functionality can be easily expanded. The availability of the IPython notebook [**?**] makes the solver very interactive, so that it should be easier for new users to get started.

In order to maximize ease of use, SpecTraVVave was designed to find even solutions of (2.1). Symmetry of steady solutions can be proved for some of the models in the form (2.1), but not for all [**?**]. Some of these equations also admit non-smooth solutions, for instance as termination points of a bifurcation branch. This happens for example for the Whitham equation, which features bifurcation curves which terminate in a solution with a cusp [**?**]. One of the goals of the present paper is to investigate the precise nature of the termination of the bifurcation curve.

The content of the paper is structured as follows. A mathematical description of the numerical method of SpecTraVVave is given in section 2.3. **??** presents results of different experiments carried out with the package. Concluding remarks are given in **??**. A method for finding initial guesses for the solver is described in **??**. **??** contains a schematic of program and a description of its workflow.

2.1.2 Spectral scheme and construction of nonlinear system.

Cosine collocation method.

To compute traveling wave solutions to the equation (2.1) the following ansatz is employed:

$$u(x,t) = \phi(x-ct).$$

Thus, the equation takes the form

$$\phi' + [f(\phi)]' + \mathcal{L}\phi' = 0,$$

which can be integrated to give

$$-c\phi + f(\phi) + \mathcal{L}\phi = B. \tag{1.5}$$

The constant B is a priori undetermined. One may set the B equal to zero as a way of normalizing the solutions. Another option is to impose an additional condition, for example that the integral of ϕ over one wavelength be zero. In this case, B will be found along with the solution ϕ.

We consider $\mathcal{L}$ as a Fourier multiplier operator with symbol $\alpha(k)$. We also assume that f is at least twice differentiable, and we have $f(0) = 0$ and $f'(0) = 0$. When computing traveling-wave solutions we focus on even periodic solutions. While it can be proved in some cases that solutions of (3.5) must be even, this is not known for a general operator $\mathcal{L}$. Nevertheless, we make this assumption here in order to make the numerical procedure as uniform as possible. For even periodic solutions, one may use a cosine collocation instead of a Fourier method. In particular, using the cosine functions as basis elements automatically removes the inherent symmetries due to reflective and translational symmetry. Moreover, the number of unknowns is reduced by a factor of 2, and the problem of the asymmetric arrangement of nodes in the FFT is circumvented.

Of course, all these problems could also be dealt with a collocation method based on the Fourier basis, but the cosine basis does all of the above automatically. In addition, the Python cosine transform is based on an integrated algorithm, which relies on an optimized version of the discrete cosine transform (DCT).

The following description of computation scheme was presented in detail in [?], but we will briefly repeat it here for consistency of the manuscript. For the purpose of clarity, we will refer to full wavelength L of a solution as the (full) wavelength, and half of fundamental wavelength will be called half-wavelength. Such a definition is required because the method computes a half of a solution profile, the other half is automatically constructed due to symmetry.

Traveling wave solutions to the equation (3.5) are to be computed in the form of a linear combination of cosine functions of different wave-numbers, i.e., in the space

$$\mathcal{S}_N = \text{span}_{\mathbb{R}} \cos(lx) 0 \leq l \leq N-1. \tag{1.6}$$

This is a subspace of $L^2(0,2\pi)$, and the collocation points $x_n = \pi\frac{2n-1}{2N}$ for $n = 1, \dots, N$ are used to discretize the domain. If the required full wavelength of solutions is to be $L \neq 2\pi$, one can use a scaling on the x-variable. Defining the new variable

$$x' = \frac{L}{2\pi}x,$$

yields collocation points x'_n and wavenumbers κ_l defined by

$$x'_n = \frac{L}{2}\frac{2n-1}{2N}, \qquad \kappa_l = \frac{2\pi}{L}l.$$

We are seeking a function $\phi_N \in \mathcal{S}_N$ that satisfies the equations

$$-c\phi_N(x'_n) + f(\phi_N)(x'_n) + \mathcal{L}^N\phi_N(x'_n) = 0, \tag{1.7}$$

at the collocation points x'_n. The operator $\mathcal{L}^N$ is the discrete form of the operator $\mathcal{L}$, and ϕ_N is the discrete cosine representation of ϕ which is given by

$$\phi_N(x') = \sum_{l=0}^{N-1} \omega(\kappa_l)\Phi_N(\kappa_l)\cos(\kappa_l x'),$$

$$\omega(\kappa_l) = \begin{cases} \sqrt{1/N}, & \kappa_l = 0, \\ \sqrt{2/N}, & \kappa_l > 0, \end{cases}$$

$$\Phi_N(\kappa_l) = \omega(\kappa_l)\sum_{n=1}^{N} \phi_N(x'_n)\cos(\kappa_l x'_n),$$

where $\kappa_l = 0, \frac{2\pi}{L}, \dots, \frac{2\pi}{L}(N-1)$ are the scaled wavenumbers, and $\Phi_N(\cdot)$ are the discrete cosine coefficients. As the equation (??) is enforced at the collocation points x'_n, one may evaluate the term $\mathcal{L}^N\phi_N$ using the matrix $\mathcal{L}^N(i,j)$ defined by

$$\mathcal{L}^N\phi_N(x'_i) = \sum_{j=1}^{N} \mathcal{L}^N(i,j)\phi_N(x'_j),$$

$$\mathcal{L}^N(i,j) = \sum_{l=0}^{N-1} \omega^2(\kappa_l)\alpha(\kappa_l)\cos(\kappa_l x'_i)\cos(\kappa_l x'_j),$$

where $\alpha(\cdot)$ is the Fourier multiplier function of the operator $\mathcal{L}$.

2.1.3 Construction of nonlinear system.

The equation (**??**) enforced at N collocation points yields a nonlinear system of N equations in N unknowns, which can be written in shorthand as

$$F(\phi_N) = 0.$$

This system can be solved by a standard iterative method, such as Newton's method. In this system, the value of phase speed c has to be fixed for computing one particular solution. Such an approach becomes impractical when a turning point on the bifurcation curve appears.

In SpecTraVVave a different approach is employed: both the amplitude a and the phase speed c of a solution are treated as functions of a parameter θ: $a = a(\theta), c = c(\theta)$. The parameter θ is to be computed from the system (**??**). This construction makes it possible to follow turning points on the bifurcation branch with relative ease. Having computed two solutions, i.e., two points on the bifurcation curve $P_1 = (c_1, a_1)$ and $P_2 = (c_2, a_2)$, one may find a direction vector $\mathbf{d} = (d^c, d^a)$ of the line that contains these points:

$$\mathbf{d}: \quad d^c = c_2 - c_1, \qquad d^a = a_2 - a_1.$$

Then the point $P_3 = (c_3, a_3)$ is fixed at some (small) distance s from the point P_2 in the direction $\mathbf{d}$.

$$P_3: \quad c_3 = c_2 + s \cdot d^c, \qquad a_3 = a_2 + s \cdot d^a.$$

The point P_3 plays the role of the initial guess for velocity and amplitude when computing the next solution $P_* = (c_*, a_*)$. The solution point P_* is required to lay on the line with direction vector $\mathbf{d}_\perp = (d^c_\perp, d^a_\perp)$, which is orthogonal to the vector $\mathbf{d}$.

$$\begin{aligned} \mathbf{d}_\perp: &\quad d^c_\perp = -d^a, &\quad d^a_\perp = d^c, \\ P_*: &\quad c_* = c_3 + \theta d^c_\perp &\quad a_* = a_3 + \theta d^a_\perp. \end{aligned}$$

A schematic sketch of finding a new solution P_* is given in Figure 1.

The variable θ is computed by Newton's method from the extended system

$$F\begin{pmatrix} \phi_N(x_1) \\ \vdots \\ \phi_N(x_N) \\ B \\ \theta \end{pmatrix} = \begin{pmatrix} (-c + \mathcal{L}_N)\phi_N(x_1) + f(\phi_N(x_1)) - B \\ \vdots \\ (-c + \mathcal{L}_N)\phi_N(x_N) + f(\phi_N(x_N)) - B \\ \Omega(\phi_N, c, a, B) \\ \phi_N(x_1) - \phi_N(x_N) - a \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{1.8}$$

Here a nonhomogeneous problem ($B \neq 0$) is considered. The equation

$$\phi_N(x_1) - \phi_N(x_N) - a = 0,$$

makes the waveheight of the computed solution to be that of a. The equation

$$\Omega(\phi_N, c, a, B) = 0,$$

is called the *boundary condition.* It allows to enforce different specifications on the computed traveling wave solution. For example, if one sets

$$\Omega(\phi_N, c, a, B) = \phi_N(x_1) + \cdots + \phi_N(x_N),$$

then *the mean of the computed wave over a period will have to be equal to zero.* One may also experiment with

$$\Omega(\phi_N, c, a, B) = B,$$

to consider the homogeneous problem $(B = 0)$. It can be also interesting to set

$$\Omega(\phi_N, c, a, B) = \phi_N(x_N). \tag{1.9}$$

This enables us to compute traveling wave solutions that mimic solitary wave solutions.

2.1.4 Convergence.

In order to test the numerical implementation of the discretization, the method is applied to a case where the solution is known. One such case is the KdV equation

$$u_t + u_x + \frac{3}{2}uu_x + \frac{1}{6}u_{xxx} = 0,$$

which has a known solution, given in the form

$$u_{\text{exact}}(x,t) = a\,\text{sech}^2\left(\sqrt{\tfrac{3a}{4}}(x-ct)\right),$$

with $c = 1 + a/2$. Using the boundary equation (**??**), SpecTraVVave is capable of computing approximations to solitary wave solutions of nonlinear wave equations. Solitary wave solutions are treated as traveling waves with sufficiently long wavelength that have the wave trough at zero. In case of the KdV equation solitary wave solutions have exponential decay, and therefore, considering the symmetry of solitary solutions, the half-wavelength of 30 is considered for the comparison. Approximation errors are summarized in **??**.

Nb. of grid points	$\log_{10}(\|u_{\text{exact}} - u\|_{L^\infty})$	$\log_{10}(\|u_{\text{exact}} - u\|_{L^2})$	Ratio of L^2-errors
32	−1.389	−2.092	
64	−3.705	−4.549	286.8
128	−8.809	−9.508	90935.0
256	−15.353	−16.144	4329670.9
512	−15.353	−16.087	0.9

Table 2.1: Estimates of error between the exact and computed solitary wave solutions for the KdV equation. Half-wavelength 30, waveheight $a = 1.2651$