Electronic Journal of Differential Equations, Vol. 2017 (2017), No. 62, pp. 1–23.
ISSN: 1072-6691. URL: http://ejde.math.txstate.edu or http://ejde.math.unt.edu

A NUMERICAL STUDY OF NONLINEAR DISPERSIVE WAVE MODELS WITH SPECTRAVVAVE

HENRIK KALISCH, DAULET MOLDABAYEV, OLIVIER VERDIER

Communicated by Jerry Bona

Abstract. 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.

1. Introduction

This article concerns 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

2010 *Mathematics Subject Classification.* 35C07, 35Q53, 45J05, 65M70.
Key words and phrases. Traveling Waves; nonlinear dispersive equations; bifurcation; solitary waves.

Submitted November 7, 2016. Published March 2, 2017.

multiplier operator is given by $\mathcal{L} = I - \mathcal{H}\partial_x$, where the Hilbert transform $\mathcal{H}$ is defined as

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

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

$$\mathcal{L}u(x) = \int_{-\infty}^{\infty} K_{h_0}(y)u(x-y)\,\mathrm{d}y, \quad \widehat{K_{h_0}}(k) = \sqrt{\frac{g\tanh(h_0k)}{k}}, \tag{1.3}$$

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

The particular form of equation (1.1) 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 (1.1) higher Fourier modes must be considered to obtain solutions.

The order of the operator $\mathcal{L}$ appearing in (1.1) 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 [35]. Lower-order operators such as the operator K_{h_0} in (1.3) appearing in the Whitham equation may allow solutions to develop singularities, such as derivative blow-up (see [29, 31]) and formation of cusps (see [25]).

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 $p > 2$ [11], but no proofs are available at this time.

To study different phenomena related to equations of the form (1.1) and their traveling wave solutions, a Python-based solver package SpecTraVVave was developed by the authors [41]. The general idea behind the solver is to use a numerical continuation method [36] implemented with a pseudo-spectral algorithm. Similar previous projects include AUTO [21] and Wavetrain [49]. 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 [45]

makes the solver very interactive, so that it should be easier for new users to get started.

To maximize ease of use, SpecTraVVave was designed to find even solutions of (1.1). Symmetry of steady solutions can be proved for some of the models in the form (1.1), but not for all [16]. 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 [25]. One of the goals of the present paper is to investigate the precise nature of the termination of the bifurcation curve.

The content of this article is structured as follows. A mathematical description of the numerical method of SpecTraVVave is given in Section 2. Section 3 presents results of different experiments carried out with the package. Concluding remarks are given in Section 4. A method for finding initial guesses for the solver is described in Section 5. Section 6 contains a schematic of program and a description of its workflow.

2. Spectral scheme and construction of nonlinear system.

2.1. **Cosine collocation method.** To compute traveling wave solutions to (1.1) 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

$$-c\phi + f(\phi) + \mathcal{L}\phi = B. \tag{2.1}$$

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 (2.1) 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 [24], 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 (2.1) are to be computed in the form of a linear combination of cosine functions of different wave-numbers, i.e., in the space

$$\mathcal{S}_N = \operatorname{span}_{\mathbb{R}}\{\cos(lx) : 0 \le l \le N-1\}. \tag{2.2}$$

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

$$-c\phi_N(x'_n) + f(\phi_N)(x'_n) + \mathcal{L}^N\phi_N(x'_n) = 0, \tag{2.3}$$

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 equation (2.3) 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 $\mathcal{L}$.

2.2. **Construction of nonlinear system.** Equation (2.3) 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 (2.4). 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:

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

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

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

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

$$F\begin{pmatrix} \phi_N(x_1) \\ \vdots \\ \phi_N(x_N) \\ B \\ \theta \end{pmatrix} = \begin{pmatrix} (-c + \mathcal{L}_N)\phi_N(x_1) + f(\phi_N(x_1)) - B \\ \vdots \\ (-c + \mathcal{L}_N)\phi_N(x_N) + f(\phi_N(x_N)) - B \\ \Omega(\phi_N, c, a, B) \\ \phi_N(x_1) - \phi_N(x_N) - a \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.4}$$

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

$$\Omega(\phi_N, c, a, B) = \phi_N(x_N). \tag{2.5}$$

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

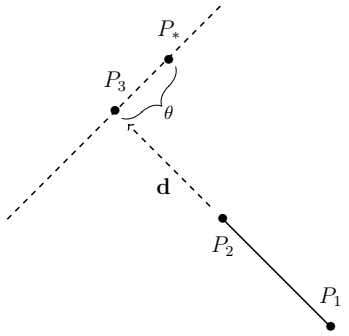


FIGURE 1. Navigation on the bifurcation curve.

2.3. **Convergence.** 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

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

with $c = 1 + a/2$. Using the boundary equation (2.5), 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 Table 1.

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

3. Experiments with SpecTraVVave.

3.1. **Termination of waveheight-velocity bifurcation curve of the Whitham equation.** The waveheight-velocity bifurcation curve of the Whitham equation

$$u_t + \frac{3}{2}uu_x + Ku_x = 0, \quad \widehat{Ku}(k) = \sqrt{\frac{\tanh(k)}{k}}, \tag{3.1}$$

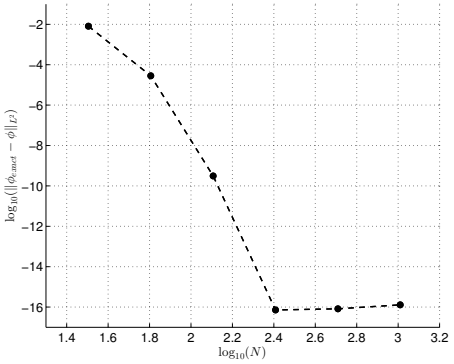


FIGURE 2. Graph of error estimates given in Table 1.

was studied numerically in [24]. An attempt was made to identify the termination point of the Whitham bifurcation curve. The investigation was limited by computational tools and complete results were not obtained. In particular, the authors could not confirm that traveling wave solutions do not exist past the point where the authors, based on pioneering work of Whitham [52] suspected a cusped solution. In this section a number of numerical results on nature of the bifurcation curve for the Whitham equation are presented. Solutions to (3.1) are computed in the form of traveling waves $u(x,t) = \phi(x - ct)$ and the homogeneous ($B = 0$) integrated version the equation is considered:

$$-c\phi + \frac{3}{4}\phi^2 + K\phi = 0. \tag{3.2}$$

Special attention is given to relation between stability of solutions and their wave-height and velocity parameters, i.e., their position on the bifurcation curve. The following questions are under study:

(a) Where does the bifurcation curve terminate?
(b) Where on the bifurcation curve do solutions change their stability?
(c) Is there any role that the turning point on the bifurcation curve plays?

The results presented here focus on 2π-periodic solutions to (3.2), i.e., solutions of system (2.4). Figure 3 presents Whitham bifurcation curves with numbers of grid points $N = 512$, $N = 1024$ and $N = 2048$. The current implementation of the SpecTraVVave package allows fixing the number of grid points N and a so-called doubling parameter $\mathcal{D}$, i.e., the number by which N is doubled as computations are made. This allows us to get sets of solutions with $N, 2N, \ldots, 2^{\mathcal{D}}N$ grid points. If $\mathcal{D} = 1$ then only two sets of solutions are computed and they are regarded as lower grid (lower resolution) and higher grid (higher resolution) solutions. While system (2.4) is processed by Newton solver, lower grid solutions are taken as initial guesses for higher grid solutions. All curves shown in this manuscript have been produced after tests with a number of resolutions were run, and the curves shown did not change significantly under further refinement.

Figure 4(a) presents the Whitham bifurcation curve computed by SpecTraVVave with $N = 1024$ and $\mathcal{D} = 1$. There are three solutions which deserve to be singled out:

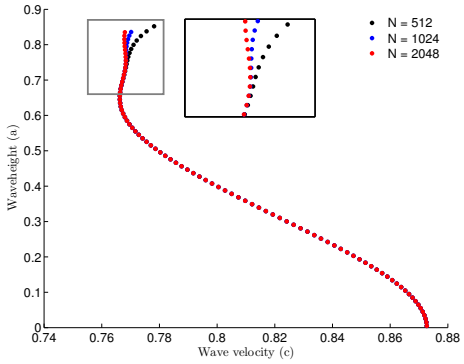


FIGURE 3. Whitham bifurcation curve in different grid resolutions.

(1) Traveling wave solution with minimum velocity (rhombus);
(2) Traveling wave solution with maximum L^2-norm (circle);
(3) Cusped traveling wave solution (square).

Profiles of the above listed solutions are given in Figure 5(a). The solution with minimum velocity corresponds to the turning point of the bifurcation curve. The solution with maximum L^2-norm is very close to the latter one, although it has a higher waveheight and a different velocity. The solution marked by a square is called here the *terminal solution*. As already mentioned, previous studies, such as [23, 24] did not provide any conclusive analysis on the part of the bifurcation curve past the turning point. In particular, it was not clear whether solutions ceased to exist at or after the turning point, or whether solutions were stable or unstable after the turning point.

Let us first focus on the stability of solutions. Note that SpecTraVVave has an evolution integrator routine, which enables one to check the stability of computed solutions. The current version of the package uses the fourth-order method developed in [20]. In addition one may use a more refined analysis, resting on the evaluation of invariant functionals. This analysis is based on the observation that the traveling waves can be thought of as solutions of a constrained minimization problem. This analysis is based on ideas developed by Boussinesq, first exploited in [8], and later used in [12, 14, 42], and many other works.

Let us define two functionals V and E:

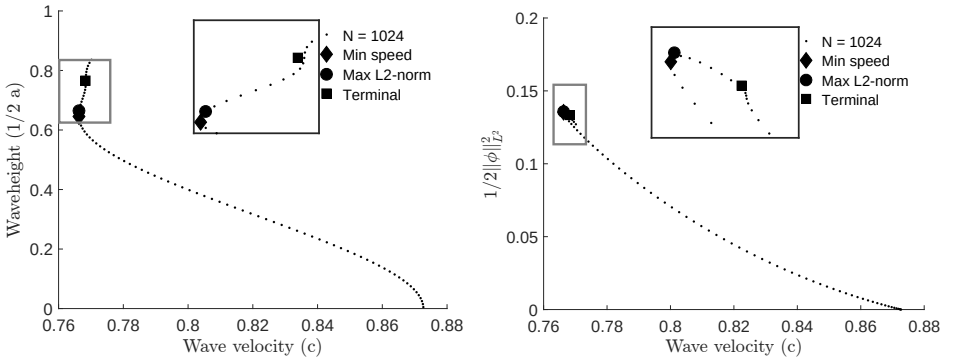
$$V(\phi) = \frac{1}{2}\int_{-\infty}^{+\infty} \phi^2 \, \mathrm{d}\zeta, \quad E(\phi) = \int_{-\infty}^{+\infty} \left\{ \frac{1}{2}\phi^3 - \phi \, K\phi \right\} \mathrm{d}\zeta.$$

Equation (3.2) can be then written in terms of variational derivatives of E and V as

$$E'(\phi) - cV'(\phi) = 0. \tag{3.3}$$

It is known from [12] that under certain conditions, the stability of solitary wave solutions depends on convexity of the function $d(c) = E(\phi) - cV(\phi)$. Solutions with values of c for which $d''(c) > 0$ are stable solutions, and solutions with wave speeds for which $d''(c) < 0$ are unstable solutions.

The current numerical investigation can therefore be interpreted as an indication of the stability properties of traveling wave solutions of the Whitham equation. Note



(a) Bifurcation curve in the wavespeed-wave-height parameter space

(b) Bifurcation curve in a wavespeed-L^2-norm diagram

FIGURE 4. Whitham bifurcation curves for 2π-periodic solutions.

that differentiation of $d(c)$ yields

$$d'(c) = \underbrace{E'(\phi) - cV'(\phi)}_{=0} - V(\phi).$$

Using (3.3) as indicated yields

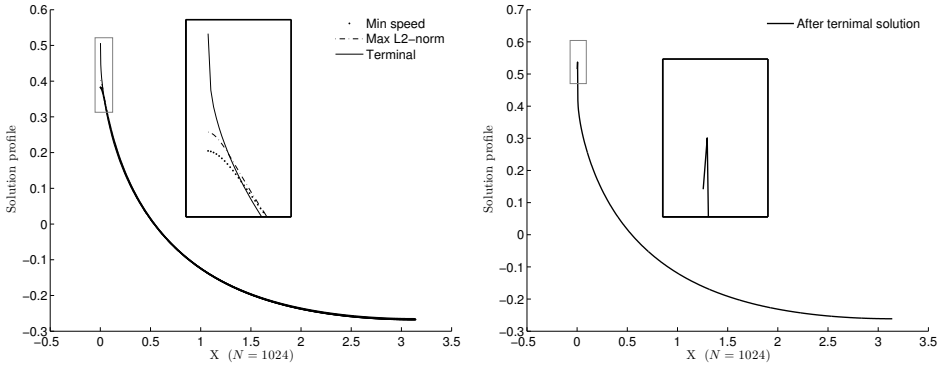
$$d'(c) = -V(\phi) = -\frac{1}{2}\int_{-\infty}^{+\infty} \phi^2 \,\mathrm{d}\zeta = -\frac{1}{2}\|\phi\|_{L^2}^2.$$

Therefore, to understand the convexity of $d(c)$, it is sufficient to find points of maximum L^2-norm on the curve in the right panel of Figure 4. It is straightforward to see that $d''(c)$ changes sign in the neighbourhood of the maximum point of this curve, i.e., around the solution with maximum L^2-norm. In particular, $d''(c) > 0$, i.e., solutions are stable to the left of the maximum point, and $d''(c) < 0$, i.e., solutions are unstable to the right of the maximum point.

In addition, the solutions were tested with the evolution integrator to confirm their stability/instability in time. The solution with maximum L^2-norm and those on the left-hand side were always found to be stable in the time-dependent computations. Solutions on the right-hand side do not preserve their shape and thus are unstable. Examples are given in Figure 6. This analysis confirms that the point corresponding to the minumum wave speed (the turning point), and the point of stability inversion are two distinct points on the bifurcation curve. Moreover, the point of stability inversion is a little further up the branch from the turning point.

Next, we turn our attention to the analysis of the terminal point. There are two main questions. Does the branch terminate, and if so, does the terminal point on the branch correspond to a cusped traveling wave. First of all, note that the solution, which is computed by SpecTraVVave, past the terminal solution has two crests, no matter how small the stepping on the bifurcation branch is taken. (see Figure 5(b)). Secondly, as will be explained presently, the relation

$$\frac{c}{\sup_x \phi(x)} = \frac{3}{2}$$



(a) Profiles of the solutions singled out in Figure 4 (a)

(b) Profile after terminal solution

FIGURE 5. Profiles of specific traveling wave solutions.

holds for the terminal solution with a good degree of approximation. For the most accurate runs, we obtain $c/\sup_{x\in\mathbb{R}}\phi(x) \approx 1.51$. To explain how this relation comes about note that the steady integrated form of the Whitham equation can be written as

$$\Big(\frac{c}{\sqrt{3}} - \frac{\sqrt{3}}{2}\phi\Big)^2 = \frac{1}{3}c^2 - K\phi. \tag{3.4}$$

It is clear that for any $\phi < 3c/2$, the relation (3.4) can be used in a bootstrap argument to show that any continuous solution must be in fact smooth. However for the case $\phi = 3c/2$ this bootstrap argument fails since the left-hand side vanishes. It can be concluded that a solutions containing a cusp will have a maximum value of $3c/2$.

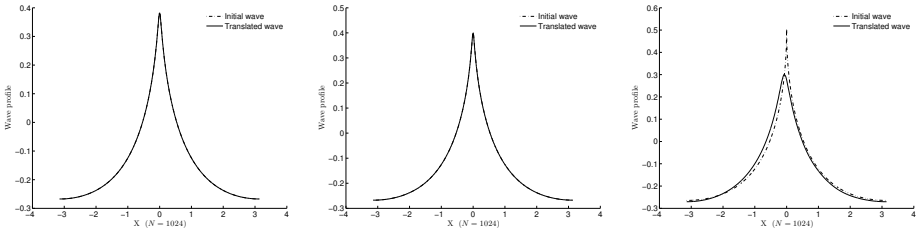
As an additional check, the discrete cosine coefficients of the solutions were examined, and fitted to the following models:

$$\mathcal{E}(k) = \nu_1 e^{-\nu_2 k^n}, \quad \mathcal{P}(k) = \frac{\mu_1}{\mu_2 + \mu_3 k^m},$$

where ν_1, ν_2, μ_1, μ_2, μ_3, n and m are fitting parameters. A smooth function is known to have discrete cosine coefficients with exponential decay in k. On the other hand, if a function is not smooth, the discrete cosine coefficients feature polynomial decay. To identify the best fit, two parameters were used: L^2-norm of the residual and the Akaike information criterion (AIC) measure.

From the data given in the Table 2, one can deduce that for solutions with minimum speed and maximum L^2-norm exponential fit is better than polynomial. That is not the case for the terminal solution. Thus, the first two solutions are smooth and the terminal solution is nonsmooth. In fact, the polynomial fit is better than exponential for solutions that are between the maximum L^2-norm solution and the terminal solution.

The numerical evidence brought forward supports the conclusion that the Whitham bifurcation branch terminates at the terminal point indicated in Figure 3. Of course, as already mentioned, this conclusion has now also been reached using tools of mathematical analysis [25].



(a) Minimum speed solution (b) Maximum L^2-norm solution (c) Terminal solution

FIGURE 6. Evolution of specific solutions in time. Time range for each solution is three periods.

Model	Min speed solution $\mathcal{E}(k)$	Min speed solution $\mathcal{P}(k)$	Max L^2-norm solution $\mathcal{E}(k)$	Max L^2-norm solution $\mathcal{P}(k)$	Terminal solution $\mathcal{E}(k)$	Terminal solution $\mathcal{P}(k)$
Res. L^2-norm	5×10^{-5}	4×10^{-3}	7×10^{-5}	3×10^{-3}	6×10^{-3}	6×10^{-4}
AIC	-543	-321	-529	-333	-298	-416

TABLE 2. Results for measures of fit.

3.2. **Interaction of solitary wave solutions of modified Benjamin–Ono equation.** In this section, we utilize the SpecTraVVave package to obtain high-precision approximations to solitary-wave solutions of the modified Benjamin–Ono

$$u_t + u^2 u_x + u_x - \mathcal{H}u_{xx} = 0,$$

which is a special case of the generalized Benjamin–Ono equation, with $p = 2$. This case corresponds to the critical scaling, i.e., invariance of the energy norm under the natural invariant scaling.

The Benjamin–Ono equation was found by Benjamin [7] as a model for long small-amplitude interfacial waves in deep water. The validity of approximating the more physically correct configuration of a continuous density distribution by the two-layer approximation has recently been justified mathematically [17].

Solitary-wave solutions of the modified Benjamin–Ono equation with $p = 3$, $p = 4$ and $p = 5$ were approximated in [11] with a standard Newton scheme. The solutions in [11] were not very accurate, but since singularity formation of the evolution equations were under study, the accuracy of the solitary-wave approximation was not an important issue. The problem with the method of [11] and some other works was that the fft used there was not purged of possible symmetries (translational and reflective). In the current code, since a cosine formulation is chosen, these symmetries are automatically eliminated, and the resulting computations are able to to render more accurate approximations.

Solitary-wave solutions of these equation could be computed with higher accuracy using a type of Petviashvili method in [44], but here we employ the SpecTraVVave package using the boundary equation (2.5), and treating solitary waves as traveling waves with sufficiently long wavelength that have wave trough at zero. Once these high-accuracy solutions are found, they are aligned in an evolution code using a high-order time integrator, and the interaction of two waves is studied.

Two questions are investigated. First, the interaction is investigated for evidence of integrability. Second, we are looking for possible annihilation of one of the waves, such as may happen in some other evolution equations [19].

A possible approach to studying the question of complete integrability is analyzing the interaction of two solitary wave solutions of the equation, such as carried out in [32, 34] for other nonlocal equations. In Figure 7, snapshots of interaction of two solitary waves at different times are shown. The time difference between two consecutive snapshots is constant. As it may be observed, during the process of interaction, the two initial solitary waves combine into a single wave, and an additional oscillation is produced. This leads us to the conclusion that the interaction of solitary waves is not elastic and the modified Benjamin–Ono equation may not be integrable. In addition, it appears that the smaller wave disappears as most of its mass is acquired by the larger wave. Thus one may argue that the small wave is annihilated by the larger wave. It can also be observed that the larger wave starts growing, and it is likely that this growth will lead to finite-time blow-up. This question was not investigated further since blow-up phenomena have already been studied closely in [34]. Indeed, very recently, the finite-time blow-up has been proven mathematically in [39].

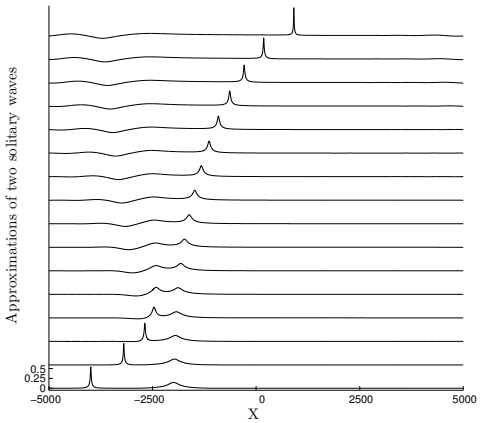


FIGURE 7. Interaction of two solitary waves of the modified Benjamin-Ono equation.

3.3. **Effect of competing dispersion in the Benjamin equation.** The Benjamin equation was found by Benjamin [9] as a model for two-layer flow in the case when the interface is subject to surface tension. The approximation may not be a good model for a stratified situation, but more applicable to the case where two fluids are separated by a sharp interface. The equation is

$$u_t + u_x + uu_x - \mathcal{H}u_{xx} - \tau u_{xxx} = 0, \tag{3.5}$$

where τ is a parameter similar to the inverse of the Bond number in free surface flow [9, 33, 50].

In this section, a study relating to the effects of competing dispersion operators on the shape of periodic traveling waves in the Benjamin equation is presented. An in-depth study of solitary waves was carried out in [22]. As will come to light, the

periodic case features some new phenomena, such as secondary bifurcations, connecting and crossing branches. For the purpose of this study, we fix the parameter $\tau = 0.1$, so that the dispersion relation for the linearized equation is

$$c(k) = 1 - |k| + 0.1k^2. \tag{3.6}$$

Traveling wave solutions with full wavelengths $L_1 = \pi/5$, $L_2 = 4\pi/19$ and $L_3 = 4\pi$ are computed for (3.5). The corresponding wavenumbers are $k_1 = 10$, $k_2 = 19/2$, and $k_3 = 0.5$, respectively. A plot of the dispersion relation (3.6) is given in Figure 8. Bifurcation branches of traveling wave solutions with the selected wavelengths are given in Figure 9.

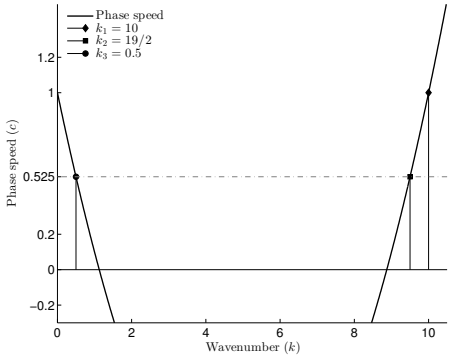


FIGURE 8. Dispersion relation (3.6)

The branch denoted by L_1 originates at the bifurcation point located at $c = 1$ and zero waveheight. The branches denoted by L_2 and L_3 originate from the same bifurcation point, located at $c = 0.525$ and zero waveheight. These two branches continue in different directions, due to differences in wavelength. In particular, the L_3 branch contains waves with shorter wavelengths, and falls into the capillary regime. On the other hand, the L_2 branch falls in the gravity regime. As the waveheight grows, solutions on the L_3 branch first cross the L_1 branch without

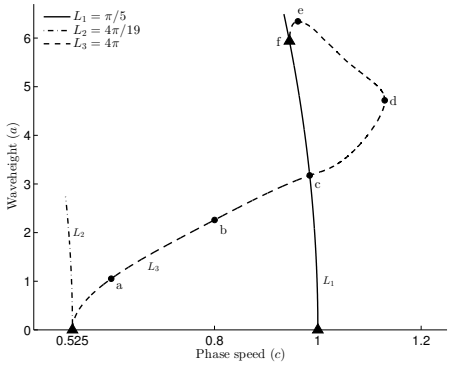


FIGURE 9. Bifurcation curves of (3.5) with different wavelengths, ▲ – points of bifurcation, • – selected solutions (see Figure 10).

connecting. Additional oscillations develop in the solutions until a new fundamental wavelength $\pi/5$ is reached, and the branch terminates as it connects to the L_1 branch. The situation is depicted in Figure 10. The point where the L_1 and L_3 branches meet is approximately $(c^*, a^*) = (0.945, 5.938)$. The corresponding profiles essentially overlap, as shown on Figure 11. This point can also be interpreted as a secondary bifurcation point of the L_1 branch, where solutions with wavelengths that are multiples of $\pi/5$ develop. We should note that similar phenomena concerning crossing and connecting branches were previously observed in [47] for the Whitham equation with surface tension which was introduced in [30].

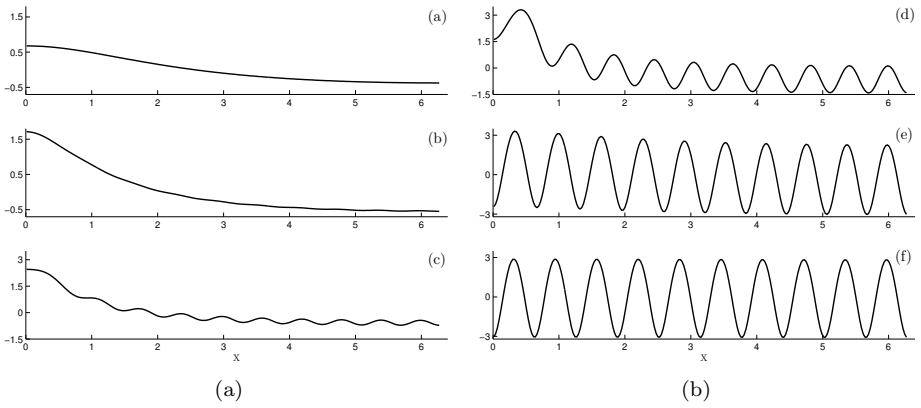


FIGURE 10. Selected solutions of (3.5). Labels preserved as shown in Figure 9.

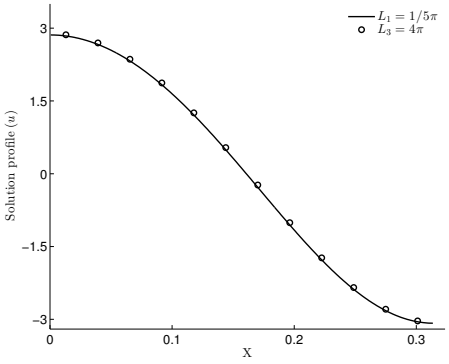


FIGURE 11. Solution profiles at the point (c^*, a^*) where the L_1 and L_3 branches meet (see Figure 9).

4. Conclusions and future work

The numerical algorithm of SpecTraVVave features ample flexibility for researching different aspects of nonlocal dispersive wave equations and their traveling wave

solutions. The solver package is simpler in use when compared with programs such as AUTO and Wavetrain, however it does not have the same level of generality. Moreover, AUTO and Wavetrain are programmed in low-level programming languages and will therefore run more efficiently. SpecTraVVave is implemented in an object-oriented fashion [27], which makes the program easily expandable. IPython provides means for interactive work with the package, and enables users to create convenient notebook-programs. A parametric approach in defining amplitude and phase speed makes it possible to follow turning points on bifurcation curves. Specification of different boundary conditions allows computing solutions with certain features, such as traveling waves with zero mean, or approximations to solitary waves.

In this work, the SpecTraVVave package has been put to use for the study of a number on nonlinear evolution equations: the Whitham equation, the modified Benjamin–Ono equation and the Benjamin equation. For the chosen set of parameters, experiments on the Whitham equation resulted in numerical confirmation of the conjecture on cusped solutions. It was also possible to identify the point of stability inversion of traveling wave solutions of the equation and the termination point of its bifurcation curve.

In case of the modified Benjamin–Ono equation, the study on solitary wave solutions lead us to conclude that interaction process ended with annihilation of one of the two waves. The experiment on the Benjamin equation showed one more example of the effect of competing dispersion. As the amplitude increased, traveling wave solutions of wavelength 4π developed additional oscillations, and later connected up with a branch of solutions with wavelength $\pi/5$.

Future work on the SpecTraVVave package will be focused on development of its functionality and broadening the range of problems that can be studied. Possible extensions may include implementation of algorithms based on the Petviashvili method [4, 5] and generalization to systems of equations.

5. Appendix: Computing initial guesses from Stokes expansion.

$$u_t + [f(u)]_x + \mathcal{L}u_x = 0, \tag{5.1}$$

The goal of this section is to explain how the idea of Stokes's approximation works in providing the initial data (guess) on wave and phase velocity for solving (5.1) numerically.

We will consider $\mathcal{L}$ being linear and self-adjoint Fourier multiplier operator, and a function f that has degree of zeros $p \geq 2$:

$$\widehat{\mathcal{L}u}(k) = \alpha(k) \cdot \widehat{u}(k),$$
$$\langle \mathcal{L}u, v\rangle_{L^2(0,L)} = \langle u, \mathcal{L}v\rangle_{L^2(0,L)},$$
$$p = \min_{k\in\mathbb{N}} f^{(k-1)}(0) = 0 \text{ and } f^{(k)}(0) \neq 0$$

Consider equation (5.1) and its solution in the form $u(x,t) = \phi(x-ct)$, which is a traveling wave solution. Inserting $\phi(x-ct)$ into (5.1) leads to the equation

$$-c\phi' + f'(\phi)\phi' + \mathcal{L}\phi' = 0,$$

which can be integrated to give

$$-c\phi + f(\phi) + \mathcal{L}\phi = B, \quad B = \text{const}. \tag{5.2}$$

Consider $B = 0$ in equation (5.2), and expansions of ϕ and c:

$$\phi = \xi = \varepsilon\xi_1 + \varepsilon^2\xi_2 + \dots, \tag{5.3}$$

$$c = c_0 + \varepsilon c_1 + \varepsilon^2 c_2 + \dots. \tag{5.4}$$

The next step is to insert (5.3) and (5.4) to (5.2) and write out the terms at powers of ε. The function $f(\phi)$ is expanded around zero and, therefore, will appear only in ε^p terms. Thus, the term at the first power of ε reads

$$\varepsilon: \quad -c_0\xi_1 + \mathcal{L}\xi_1 = 0, \tag{5.5}$$

Hence, c_0 is an eigenvalue of the operator $\mathcal{L}$, regarded as defined on L-periodic functions. Taking the Fourier transform of (5.5) gives:

$$-c_0\widehat{\xi_1}(k) + \alpha(k)\widehat{\xi_1}(k) = 0. \tag{5.6}$$

Equation (5.6) has two trivial solutions: either $\xi_1(k) \equiv 0$ or $\alpha(k) \equiv c_0$. If we assume non-trivial ξ_1 and $\alpha(k) \neq \text{const}$, the following solves the problem

$$\widehat{\xi_1}(k) = 2\pi\delta(k - k_0), \quad \text{and} \quad c_0 = \alpha(k_0), \tag{5.7}$$

for some $k_0 \in \mathbb{R}$. Since ξ_1 is the first-order approximation to ϕ, the corresponding wave number should be equal to 1. The L-periodicity condition entails that $k_0 = 2\pi/L \cdot 1$. The spatial variable x has to be scaled to $x' = L/2\pi x$, accordingly. From the solutions in (5.7) we have

$$\xi_1(x') = e^{ik_0x'} = \cos(k_0x') + \sin(k_0x'). \tag{5.8}$$

Considering the projection onto the space $\mathcal{S}_N$, we are led to choose $\xi_1(x') = \cos(k_0x')$. For further analysis, let us define an operator $\mathcal{A}$,

$$\mathcal{A} := -c_0\mathcal{E} + \mathcal{L},$$

where $\mathcal{E}$ is the identity operator. The operator $\mathcal{A}$ inherits the property of being self-adjoint from $\mathcal{L}$. Moreover, it follows from (5.5) that $\mathcal{A}\xi_1 = 0$ and $\xi_1 \in \ker(\mathcal{A})$. If $p > 2$ then $f''(0) = 0$ and the terms at ϵ^2 are:

$$\mathcal{A}\xi_2 - c_1\xi_1 = 0. \tag{5.9}$$

Taking scalar multiplication of the latter with ξ_1, one obtains

$$\langle\xi_1, \mathcal{A}\xi_2\rangle_{L^2(0,L)} = c_1\|\xi_1\|^2_{L^2(0,L)}, \quad \langle\xi_1, \mathcal{A}\xi_2\rangle_{L^2(0,L)} = \langle\mathcal{A}\xi_1, \xi_2\rangle_{L^2(0,L)} = \langle 0, \xi_2\rangle_{L^2(0,L)} = 0.$$

As a result, one has $c_1\|\xi_1\|^2_{L^2(0,L)} = 0$ and, hence, $c_1 = 0$. Repeating the same argument, it becomes clear that $c_k = 0$ for any $k \leq p-1$. Besides that, ξ_2 is in the kernel of $\mathcal{A}$, so it may be assumed to be proportional to ξ_1. The terms at order ε^p are:

$$\mathcal{A}\xi_p - c_{p-1}\xi_1 + \frac{f^{(p)}(0)}{p!}\xi_1^p = 0. \tag{5.10}$$

Let us denote for brevity

$$f_p := \frac{f^{(p)}(0)}{p!}.$$

Pairing (5.10) with ξ_1 (and assuming $\|\xi_1\|_{L^2(0,L)} = 1$) gives

$$c_{p-1} = f_p \cdot \langle\xi_1^p, \xi_1\rangle_{L^2(0,L)}, \tag{5.11}$$

which gives us the value of c_{p-1}. It only remains to solve the following problem numerically in order to obtain ξ_p:

$$\mathcal{A}\xi_p = c_{p-1}\xi_1 - f_p\xi_1^p \tag{5.12}$$

For the last equation to be solved, the operator $\mathcal{A}$ has to be invertible. It is also required that $\langle \xi_1, \xi_p \rangle_{L^2(0,L)} = 0$. Therefore the solution is sought in the space orthogonal to $\ker(\mathcal{A})$. Since $\mathcal{A}$ is still a Fourier multiplier operator, one can take the Fourier transform of (5.12) to find

$$\widehat{\mathcal{A}}(k)\widehat{\xi}_p(k) = c_{p-1}\widehat{\xi}_1(k) - f_p\widehat{\xi_1^p}(k),$$
$$\widehat{\xi}_p(k) = \widehat{\mathcal{A}}(k)^{-1}\left(c_{p-1}\widehat{\xi}_1(k) - f_p\widehat{\xi_1^p}(k)\right).$$

Taking the inverse Fourier transform of $\widehat{\xi}_p(k)$ gives ξ_p. Since only even solutions of the problem are considered the cosine part of the Fourier transforms will be required. It is sufficient to use ξ_1 and c_0 as the initial guesses for the Newton method. However, it should be noted that for different values of p the pair of parameters ξ_p and c_{p-1} are computed in different ways.

(a) If $p = 2$, then ξ_2 is computed from (5.12), but c_{p-1} here becomes zero. Therefore one has to consider the next level of the expansion ε^p.
(b) For odd values of p the parameter c_{p-1} can be computed from (5.11) and ξ_p from (5.12).
(c) For even values of $p \geq 4$ the parameter ξ_p can be computed, but c_{p-1} may not be non-zero in general. In such cases a different strategy of fixing the initial guess should be used.

6. Appendix: Presentation of SpecTraVVave and its workflow

6.1. **Overview.** There are several classes in the SpecTraVVave package. An overview of the program is shown in Figure 12. The workflow begins with defining a flux function f and the Fourier multiplier function α to set up an equation. The traveling wave solution is characterized by the wavelength L and a boundary condition $\Omega(c, a, \phi_N, B)$. These parameters are fixed for a given problem. The defined equation is then discretized. The `Discretization` object contains all required elements such as grid points, wave-numbers and the discrete linear operator.

The initial guess and the equation's residual are passed from the `Discretization` to the `Solver` object. The `Navigation` object is responsible for finding good initial guesses for c and a that are passed to the `Solver` object. The `Solver` object applies Newton's method to find a solution to system of equations (2.4).

The new solution is sent back to the `Discretization` and `Navigation` objects, where variables get updated. All computed solutions are stored for further analysis. This finishes one iteration. For the next iteration the updated variables are used and a new solution is found. The process may be continued as long as the Jacobian of the problem is non-singular.

6.2. **Class Description.** We present an overview of the classes used in SpecTraVVave package. Note that, since the package is under continuous modification and development, we describe here only the basic classes and functions the package. We refer to the package repository [41] for up-do-date tutorials and installation instructions.

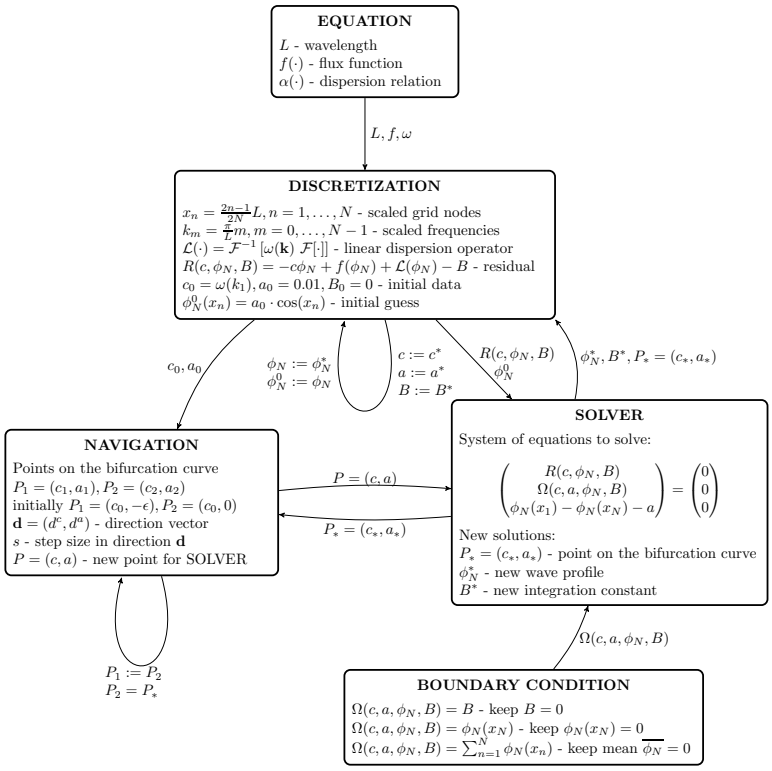


FIGURE 12. Overview of the SpecTraVVave package.

The `Equation` class is the general class for all model equations. Its only role is to store a parameter L, the wavelength:

```
class Equation(object):
    def __init__(self, L):
        self.length = L
```

A subclass of the `Equation` class has to implement two functions, `compute_kernel` and `flux`.

The KdV model equation

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

with $f(u) = \frac{3}{4}u^2$ and $\widehat{\mathcal{L}u}(k) = (1 - \frac{1}{6}k^2)\widehat{u}(k)$ is presented in the program as a subclass of the `Equation` class:

```
class KDV(Equation):
    def compute_kernel(self, k):
        return 1-1/6*k*k

    def flux(self, u):
        return 3/4*u*u
```

One can then create an object of the class `KDV` with the command:

```
kdv = KDV(L=np.pi)
```

The solver will compute only a half of a solutions profile. The full wavelength of the solutions of the defined equation will be equal to 2π.

In order to find solutions with specific features, boundary conditions are introduced as separate classes. For instance, the boundary condition specifying a constant of integration is implemented as follows:

```
class Const(object):
""" The boundary condition under which the constant of integration
(B) is always set to zero. """
    def __init__(self, level=0):
        self.level = level

    def enforce(self, wave, variables, parameters):
    """ Enforces the Const boundary condition. """
        return np.hstack([variables[0] - self.level])

    def variables_num(self):
    """ The number of additional variables that are required to
                                        construct
    the Const boundary condition. """
        return 1
```

A `Const` boundary condition object is created as follows:

```
boundary_condition = Const()
```

The next step is to create an object of `Discretization` class, which is initialized with a model equation such as `kdv_model` and the number of grid points. The main parts of the class are the following:

```
class Discretization(object):
    def __init__(self, model_equation, grid_size):
        self.equation = model_equation
        self.size = grid_size

    def operator(self, u):
        u_ = scipy.fftpack.dct(u, norm='ortho')
        Lv = self.fourier_multiplier()*u_
        result = scipy.fftpack.idct(Lv, norm='ortho')
        return result

    def residual(self, u, wavespeed, const_B):
        residual = - wavespeed*u + self.equation.flux(u)
                                    + self.operator(u) - const_B
        return  residual
```

The call `Discretization.operator(u)` computes $\mathcal{L}u$ as the inverse transform of a transformed convolution

$$\mathcal{L}u = \mathcal{F}^{-1}[\widehat{\mathcal{L}u}(k)] = \mathcal{F}^{-1}\left[\alpha(k)\cdot\widehat{u}(k)\right].$$

The result of the call `Discretization.residual(u, wavespeed, const_B)` is then used in the `Solver` class. An object of the `Solver` class is initialized with an object of the `Discretization` class, and a boundary condition object.

```
class Solver(object):
    def __init__(self, discrete_problem, boundary_condition):
        self.discretization = discrete_problem
        self.boundary = boundary_condition

    def solve(self, guess_wave, parameter_anchor, direction):
    """ Runs a Newton solver on a system of nonlinear equations once.
    Takes the residual(vector) to solve. parameter_anchor
    is the initial guess for (c,a) values and it is taken from the
    Navigation class. """
        size = len(guess_wave)
        self.discretization.size = size

        def residual(vector):
        """ Contructs a system of nonlinear equations. First part,
        main_residual, is from given wave equation; second part,
        boundary_residual, comes from the chosen boundary conditions.
                                                """
        . . .
        return np.hstack([main_residual, boundary_residual,
                                            amplitude_residual])
    . . .
    return new_wave, new_boundary_variables, new_parameter
```

Some omitted parts in the above script are substituted by '. . .' sign. Each iteration on a `Solver` object is run from a `Navigation` object, which takes the `Solver` object for initialization.

```
class Navigation(object):
""" Runs the Solver and stores the results. """

    def __init__(self, solver_object, size=32):
    self.solve = solver_object.solve # function to run Newton method
    self.size = size # size for navigation
    . . .
    def run_solver(self, current_wave, pstar, direction):
        new_wave, variables, p3 = self.solve(current_wave, pstar,
                                            direction)
        return new_wave, variables, p3
```

All the above classes can be modified and developed further, new classes may be defined as well.

6.3. **Detailed Workflow.** The workflow with the package consists of three basic steps:

(1) Once the necessary classes have been imported in the current namespace, generate all necessary objects:

```
equation = KDV(L=np.pi)
boundary_condition = Const()
discretization = Discretization(equation, grid_size=64)
solver = Solver(discretization, boundary_condition)
navigator = Navigation(solver)
```

(2) Choose a number of iterations, i.e., the number of solutions to compute, and run the solver:

```
n_iter = 50
navigator.run(n_iter)
```

(3) All computed solutions are stored in `navigation_object`

```
last_computed = -1
wave_profile = navigator[last_computed]['solution']
wave_speed = navigator[last_computed]['current'][0]
wave_amplitude = navigator[last_computed]['current'][1]
```

For up-to-date instructions on how to run the code, we refer the reader to the code repository `https://github.com/olivierverdier/SpecTraVVave`.

Acknowledgments. The authors would like to thank Mats Ehrnström and Erik Wahlén for fruitful discussions on the subject of the current paper. This research was supported by the Research Council of Norway on grant no. 213474/F20 and by the J C Kempe Memorial Fund on grant no. SMK-1238.

References

[1] Abdelouhab, L.; Bona, J. L.; Felland, M. Saut, J.-C.; *Nonlocal models for nonlinear dispersive waves*. Physica D **40**,360-392 (1989).

[2] Albert, J. P.; Bona, J. L.; Restrepo, J.M.; *Solitary-wave solutions of the Benjamin equation*. SIAM J. Appl. Math. **59**,2139–2161 (1999).

[3] Álvarez, J.; Durán, A.; *A numerical scheme for periodic travelling-wave simulations in some nonlinear dispersive wave models*. J. Comput. Appl. Math. **235**, 1790–1797 (2011).

[4] Álvarez, J.; Durán, A.; *An extended Petviashvili method for the numerical generation of traveling and localized waves*. Commun. Nonlinear Sci. Numer. Simul. **19**, 2272–2283 (2014).

[5] Álvarez, J.; Durán, A.; *, Petviashvili type methods for traveling wave computations: I. Analysis of convergence*. J. Comput. Appl. Math. **266**, 39–51 (2014).

[6] Álvarez, J.; Durán, A.; *Corrigendum to "Petviashvili type methods for traveling wave computations: I. Analysis of convergence*, [J. Comput. Appl. Math. 266 (2014) 39–51]. J. Comput. Appl. Math. **277**, 215–216 (2015).

[7] Benjamin, T.B .; *Internal Waves of permanent form in fluids of great depth*. J. Fluid Mech. **29**, 559–592 (1967).

[8] Benjamin, T. B.; *The stability of solitary waves*. Proc. Roy. Soc. London A 328, 153-183 (1972).

[9] Benjamin, T. B.; *A new kind of solitary wave*. J. Fluid Mech. **245**, 401–411 (1992).

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

[11] Bona, J. L.; Kalisch, H.; *Singularity formation in the generalized Benjamin–Ono equation*. Discrete Contin. Dyn. Syst. **11**, 779–785 (2004).

[12] Bona, J. L.; Souganidis, P. E.; Strauss, W. A.; *Stability and instability of solitary waves of Korteweg–de Vries type*. Proc. R. Soc. Lond. A **411**, 395-412 (1987).

[13] Borluk, H.; Kalisch, H.; Nicholls, D. P.; *A numerical study of the Whitham equation as a model for steady surface water waves*. J. Comput. Appl. Math. **296**, 293–302 (2016).

[14] Bridges, T. J.; *Superharmonic instability, homoclinic torus bifurcation and water-wave breaking*. J. Fluid Mech. **505**, 153-162 (2004).

[15] Chen, H.; *Existence of periodic travelling-wave solutions of nonlinear, dispersive wave equations*. Nonlinearity **17**, 2041–2056 (2004).

[16] Chen, H.; Bona, J. L.; *Existence and asymptotic properties of solitary-wave solutions of Benjamin-type equations*. Adv. Diff. Eq. **3**, 51-84 (1998).

[17] Chen, R. M.; Walsh, S.; *Continuous dependence on the density for stratified steady water waves*. Arch. Ration. Mech. Anal. **219**, 741–792 (2016).

[18] Choi, W.; Camassa, R.; *Fully nonlinear internal waves in a two-fluid system*. J. Fluid Mech. **396**, 1-36 (1999).

[19] Courtenay Lewis, J.; Tjon, J. A.; *Resonant production of solitons in the RLW equation.* Phys. Lett. A **73**, 275–279 (1979).

[20] de Frutos, J.; Sanz-Serna, J. M.; *An easily implementable fourth-order method for the time integration of wave problems.* J. Comp. Phys. **103**, 160–168 (1992).

[21] Doedel, E. J.; Champneys, A. R.; Fairgrieve, T. F.; Kuznetsov, Y.A.; Sandstede, B. and Wang, X.-J.; *AUTO97 : Continuation and bifurcation software for ordinary differential equations*, available at `http://indy.cs.concordia.ca/auto/`, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada, 1997.

[22] Dougalis, V. A.; Duran, A.; Mitsotakis, D.; *Numerical solution of the Benjamin equation.* Wave Motion **52**, 194–215 (2015).

[23] Ehrnström, M.; Kalisch, H.; *Traveling waves for the Whitham equation.* Differential Integral Equations **22**, 1193–1210 (2009).

[24] Ehrnström, M., Kalisch, H.; *Global bifurcation for the Whitham equation.* Math. Mod. Nat. Phenomena **8**, 13–30 (2013).

[25] Ehrnström, M.; Wahlén, E.; *On Whitham's conjecture of a highest cusped wave for a nonlocal dispersive equation.* arXiv:1602.05384 (2016).

[26] Fornberg, B.; Whitham, G. B.; *A numerical and theoretical study of certain nonlinear wave phenomena.* Phil. Trans. Roy. Soc. A **289**, 373–404 (1978).

[27] Führer, C.; Solem, J. E.; Verdier, O.; Computing with Python: An introduction to Python for science and engineering, Pearson (2014).

[28] Grujić, Z.; Kalisch, H.; *Gevrey regularity for a class of water-wave models.* Nonlinear Analysis **71**, 1160–1170 (2009)

[29] Hur, V. M.; *Breaking in the Whitham equation for shallow water waves.* arXiv:1506.04075 (2015).

[30] Hur, V. M.; Johnson, M.; *Modulational instability in the Whitham equation of water waves.* Studies in Applied Mathematics **134**, 120–143 (2015).

[31] Hur, V. M.; Tao, L.; *Wave breaking for the Whitham equation with fractional dispersion.* arXiv:1410.1570, 2014.

[32] Kalisch, H.; *Error analysis of a spectral projection of the regularized Benjamin-Ono equation.* BIT Numerical Mathematics **45**, 69–89 (2005).

[33] Kalisch, H.; *Derivation and comparison of model equations for interfacial capillary-gravity waves in deep water.* Math. Comput. Simulation **74**, 168–178 (2007).

[34] Kalisch, H.; Bona, J. L.; *Models for internal waves in deep water.* Discrete and Continuous Dynamical Systems **6**, 1–20 (2000).

[35] Kato, T.; *On the Cauchy problem for the (generalized) Korteweg–de Vries equation.* Studies in applied mathematics, 93–128, Adv. Math. Suppl. Stud., **8** (Academic Press, New York, 1983).

[36] Keller H. B.; *Numerical solution of bifurcation and nonlinear eigenvalue problems.* Applications of Bifurcation Theory, Rabinowitz, P. H., Academic Press, (1977), pp. 359–384.

[37] Lannes, D.; Saut, J.-C.; *Remarks on the full dispersion Kadomtsev–Petviashvli equation.* Kinet. Relat. Models **6**, 989–1009 (2013).

[38] Martel, Y.; Merle, F.; *Blow up in finite time and dynamics of blow up solutions for the L2-critical generalized KdV equation.* J. Amer. Math. Soc. **15**, 617–664 (2002).

[39] Martel, Y.; Pilod, D.; *Construction of a minimal mass blow up solution of the modified Benjamin-Ono equation.* arXiv:1605.01837 (2016).

[40] Moldabayev, D.; Kalisch, H.; Dutykh, D.; *The Whitham Equation as a model for surface water waves.* Phys. D **309**, 99–107 (2015)

[41] Moldabayev, D.; Verdier, O.; Kalisch, H.; *SpecTraVVave*, available at `https://github.com/olivierverdier/SpecTraVVave`.

[42] Nguyen, N. T.; Kalisch, H.; *Orbital stability of negative solitary waves.* Mathematics and Computers in Simulation **80**, 139-150 (2009).

[43] Ono, H.; *Algebraic Solitary Waves in Stratified Fluids*. J. Phys. Soc. of Japan **39**, 1082-1091 (1975).

[44] Pelinovsky, D. E.; Stepanyants, Y. A.; *Convergence of Petviashvili's iteration method for numerical approximation of stationary solutions of nonlinear wave equations.* SIAM J. Numer. Anal. **42**, 1110–1127 (2004).

[45] Pérez, F.; Granger, B. E.; *IPython: A System for Interactive Scientific Computing*, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007. URL: `http://ipython.org`

[46] Petviashvili, V.I.; *Equation of an extraordinary soliton*, Sov. J. Plasma Phys. **2**, 257–258 (1976).

[47] Remonato, F.; Kalisch, H.; *Numerical bifurcation for the capillary Whitham equation*, Physica D: Nonlinear Phenomena, 343, 51–62 (2017).

[48] Sanford, N.; Kodama, K.; Carter, J. D.; Kalisch, H.; *Stability of traveling wave solutions to the Whitham equation*. Phys. Lett. A **378**, 2100–2107 (2014).

[49] Sherratt, J. A.; *Numerical continuation methods for studying periodic travelling wave (wavetrain) solutions of partial differential equations*. Appl. Math. Comp. **218**, 4684–4694 (2012).

[50] Walsh, S.; *Steady stratified periodic gravity waves with surface tension I: Local bifurcation*. Discrete Contin. Dyn. Syst. **34**, 3241–3285 (2014).

[51] Walsh, S.; *Steady stratified periodic gravity waves with surface tension II: global bifurcation*. Discrete Contin. Dyn. Syst. **34**, 3287–3315 (2014).

[52] Whitham, G. B.; *Variational methods and applications to water waves*. Proc. Roy. Soc. London A, **299**, 6–25 (1967).

[53] Whitham, G. B.; Linear and Nonlinear Waves. Wiley, New York, (1974).

Henrik Kalisch
Department of Mathematics, University of Bergen, P.O. Box 7800, 5020 Bergen, Norway
E-mail address: `henrik.kalisch@math.uib.no`

Daulet Moldabayev
Department of Mathematics, University of Bergen, P.O. Box 7800, 5020 Bergen, Norway
E-mail address: `daulet.moldabayev@math.uib.no`

Olivier Verdier
Department of Mathematics and Statistics, University of Umea, Sweden.
Department of Computing, Mathematics and Physics, Western Norway University of Applied Sciences, Norway
E-mail address: `olivier.verdier@hvl.no`