

Fourier pseudospectral methods for 2D Boussinesq-type equations

D.T. Steinmoeller^{a,*}, M. Stastna^a, K.G. Lamb^a

^a200 University Ave. W., Waterloo, ON, N2L 3G1

Abstract

A global Fourier pseudospectral method is presented and used to solve a dispersive model of shallow water wave motions. The model equations under consideration are from the Boussinesq hierarchy of equations, and allow for appropriate modelling of dispersive short-wave phenomena by including weakly non-hydrostatic corrections to the hydrostatic pressure in the shallow water model. A numerical solution procedure for the Fourier method is discussed and analyzed in some detail, including details on how to efficiently solve the required linear systems. Two time-stepping approaches are discussed. Sample model results are presented, and the Fourier method is compared to the discontinuous Galerkin finite element method (DG-FEM) at various orders of accuracy. The present work suggests that scalable Fourier transform methods can be employed in water-wave problems involving variable bathymetry and can also be an effective tool at solving elliptic problems with variable coefficients if combined properly with iterative linear solvers and pre-conditioning. Additionally, we demonstrate: 1) that the small amounts of artificial dissipation (from filtering) inherent to the Fourier method make it a prime candidate for hypothesis-testing against water wave field data, and 2) the method may also serve as a benchmark for lower order numerical methods (e.g., Finite Volume Method, DG-FEM) that can be employed in more general geometries.

Keywords: Water waves, Wave dispersion, Mathematical models, Fluid dynamics, Boussinesq equations, Shallow water equations

*Corresponding author at: Department of Applied Mathematics, University of Waterloo, Waterloo, ON, Canada. N2L 3G1. Tel.: +1 519 888 4567 ext. 32588; fax: +1 519 746 4319.

Email address: dsteinmo@uwaterloo.ca (D.T. Steinmoeller)
Preprint submitted to Ocean Modelling

May 1, 2012

1. Introduction

Many of the recent advances in the study of dispersive water waves in geophysical fluid dynamics (GFD) and coastal engineering applications have come from numerical solutions to dispersive shallow water systems of equations. These dispersive shallow water models (SWM) all arise from the approach, often referred to as the method of Boussinesq (1872), of seeking an approximate analytical solution to the irrotational flow interior that underlies the wave-dominated free surface, followed by retaining weakly non-hydrostatic pressure corrections in the kinematic and dynamic surface conditions.

In the literature there is an overwhelming number of partial differential equation (PDE) systems referred to as “Boussinesq equations”, e.g., Brandt et al. (1997); Lynett and Liu (2004); Madsen et al. (1991); Nwogu (1993); Peregrine (1967); Madsen et al. (2002), and choosing an appropriate system for a given problem is a difficult task in and of itself since each model offers a different level of applicability and complexity. Past work includes solutions to the “extended Boussinesq equations” of Nwogu (1993) using low-order finite difference methods by Wei and Kirby (1995) and low-order finite element methods by Walkley (1999). Lynett and Liu (2004) derived a dispersive shallow water system using a two-layer depth-integration approach and solved the equations numerically using fourth-order finite differences.

More recently, high-order numerical solutions to the equations of Peregrine (1967) in arbitrary geometries were obtained by Eskilsson and Sherwin (2005) and Karniadakis and Sherwin (2005) using the discontinuous Galerkin finite element method (DG-FEM). Engsig-Karup et al. (2006) also used the high-order DG-FEM method to obtain solutions to the recent “high-order Boussinesq” formulation by Madsen et al. (2002) that represented a vast improvement over existing Boussinesq-type models in terms of more accurate dispersive, shoaling, and nonlinear characteristics.

Recent applications of Boussinesq-type systems in GFD include the studies of Brandt et al. (1997) on internal waves in the Strait of Messina and of de la Fuente et al. (2008) on the effects of dispersion on Kelvin and Poincaré waves in a stratified rotating circular basin. Although these two studies focused on low-order numerical solutions to Boussinesq-type systems, the increasing demand in the GFD community for more accurate solution techniques for these dispersive SWMs is clear.

In this work, we mainly consider high-order solution methods for a simple dispersive shallow water system in the Boussinesq family in two spatial

39 dimensions as stated by de la Fuente et al. (2008). We motivate our choice
 40 of numerical method by considering particular GFD applications where it
 41 is assumed that wave interactions with solid boundaries are not of interest
 42 and that periodic domains are suitable for capturing the desired dynam-
 43 ics. Under these assumptions, the Fourier pseudospectral method is a clear
 44 choice due to the fact that it gives the highest order of accuracy possi-
 45 ble on periodic domains, has excellent resolution characteristics, and has
 46 small amounts of inherent dissipation (see, for example, Boyd (2001)). We
 47 have opted to consider one of the more simple Boussinesq-type systems
 48 with the idea in mind that the methods presented here can be extended to
 49 more complicated sets of equations at the price of further computational
 50 expenses. We have adopted the second-order accurate Leapfrog scheme for
 51 the temporal discretization of the model equations that is commonly used in
 52 atmospheric/oceanic general circulation models (Williams, 2011; Amezcua
 53 et al., 2011). Although it is only second-order accurate, Leapfrog offers
 54 benefits in the form of requiring less memory than the corresponding linear
 55 multi-step methods (i.e., Adams–Bashforth) and fewer computations than
 56 a multi-stage Runge-Kutta method.

57 In the following section, we introduce the choice of governing equations
 58 and discuss their properties. We then introduce a simple time-stepping pro-
 59 cedure followed by a more efficient technique inspired by the approach of
 60 Eskilsson and Sherwin (2005) that reduces the size of the resulting linear
 61 system by a factor of 2 by transforming the dispersive terms to a standard
 62 pressure-type elliptic problem. A Fourier pseudospectral spatial discretiza-
 63 tion method is introduced for numerical solutions in two spatial dimensions
 64 along with strategies for solving the required linear systems. A nodal DG-
 65 FEM spatial discretization method in one dimension is also introduced. The
 66 paper concludes with validation of numerical solutions and a comparison
 67 between Fourier and DG-FEM solutions to the Boussinesq-type system in
 68 one dimension, followed by sample results obtained in two dimensions with
 69 the Fourier method. The present work suggests that scalable Fast Fourier
 70 Transform (FFT) based methods for water wave equations can be extended
 71 to physical cases involving non-constant bathymetry and can also be an ef-
 72 fective tool for solving elliptic problems with non-constant coefficients pro-
 73 vided they are used alongside an appropriate iterative linear solver with
 74 pre-conditioning. Given the highly accurate nature of the Fourier method,
 75 the results presented here may be seen as a benchmark for lower-order spa-
 76 tial discretization techniques such as DG-FEM and FVM, and allow for

77 rational hypotheses to be formulated for subsequent testing against field
78 data of water waves.

79 2. Methods

80 2.1. Governing Equations

The governing equations used by de la Fuente et al. (2008) in their study of internal waves in a circular basin for a single fluid layer are

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0, \quad (1)$$

$$\frac{\partial(uh)}{\partial t} + \nabla \cdot ((uh)\mathbf{u}) = -gh\frac{\partial\eta}{\partial x} + fvh + \frac{H^2}{6}\frac{\partial}{\partial x}\left(\nabla \cdot \frac{\partial(\mathbf{u}h)}{\partial t}\right), \quad (2)$$

$$\frac{\partial(vh)}{\partial t} + \nabla \cdot ((vh)\mathbf{u}) = -gh\frac{\partial\eta}{\partial y} - fuh + \frac{H^2}{6}\frac{\partial}{\partial y}\left(\nabla \cdot \frac{\partial(\mathbf{u}h)}{\partial t}\right), \quad (3)$$

81 where $\mathbf{u}(x, y, t) = (u(x, y, t), v(x, y, t))$ is the velocity field, $h(x, y, t) =$
82 $H(x, y) + \eta(x, y, t)$ is the total depth with H representing the undisturbed
83 depth, and η is the free surface displacement. The constants g and f are
84 the acceleration due to gravity and the Coriolis frequency, respectively. In
85 the test cases considered in this work, we focus on the case where $f = 0$
86 (no rotation) but have included the Coriolis terms in the equations above
87 since it will allow for interesting applications in geophysical fluid dynamics
88 to be studied in future work, e.g., instabilities in geostrophic jets and the
89 evolution of rotating gravity waves. The main difference between the set of
90 equations (1)–(3) and the traditional shallow water model are the dispersive
91 terms $\frac{H^2}{6}\nabla(\nabla \cdot (\mathbf{u}h)_t)$ found in the momentum equations (2) & (3). The
92 above system was first proposed by Brandt et al. (1997) in their study of
93 internal waves in the Strait of Messina.

94 We have neglected bottom and surface stresses in equations (1)–(3)
95 since their inclusion into the numerical scheme is conceptually easy and
96 contributes little to the discussion. We have also chosen to focus on the
97 case of a single fluid layer of constant density. We have made this choice
98 since multiple-layer extensions are numerically straightforward (at least for
99 Fourier methods), aside from the expected increases in computational cost.

100 2.2. Time-Stepping Techniques

101 For the moment, we will assume that we have spatially discretized
102 the system (1)–(3) using a method of lines approach as discussed by Tre-
103 fethen (2000). That is, the flow variables of interest (h, u, v) have been

104 discretized on N grid-points and are now represented by the $N \times 1$ vectors
 105 $(\mathbf{h}, \mathbf{u}, \mathbf{v}) = ([h_1, \dots, h_N]^\top, [u_1, \dots, u_N]^\top, [v_1, \dots, v_N]^\top)$, where we adopt
 106 the notation that bold-faced variables refer to the discretized approximate
 107 solution fields of the system (1)-(3). We further assume that the continu-
 108 ous spatial derivative operators $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial xy}$ have been replaced by
 109 the $N \times N$ matrices $D_x, D_y, D_{xx}, D_{yy}, D_{xy}$ or that the required matrix-
 110 vector products are attainable by other means, such as the pseudospectral
 111 technique Peyret (2002).

112 To keep the discussion as general as possible, we do not specify which
 113 spatial discretization scheme we are using since the following time-stepping
 114 schemes may be applied to a number of spatial discretization methods in-
 115 cluding Finite Difference methods, the Fourier pseudospectral method, the
 116 Chebyshev spectral collocation method, and DG-FEM Trefethen (2000).

Upon applying the method of lines to the Boussinesq system (1)-(3), we recover the semi-discrete system of equations

$$\frac{d\mathbf{h}}{dt} = -D_x(\mathbf{uh}) - D_y(\mathbf{vh}), \quad (4)$$

$$\begin{aligned} \frac{d(\mathbf{uh})}{dt} - \frac{\mathbf{H}^2}{6} \frac{d}{dt} (D_{xx}(\mathbf{uh}) + D_{xy}(\mathbf{vh})) &= -D_x(\mathbf{uuh}) - D_y(\mathbf{uvh}) \\ &\quad - g\mathbf{h}D_x\boldsymbol{\eta} + f\mathbf{vh}, \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{d(\mathbf{vh})}{dt} - \frac{\mathbf{H}^2}{6} \frac{d}{dt} (D_{xy}(\mathbf{uh}) + D_{yy}(\mathbf{vh})) &= -D_x(\mathbf{vuh}) - D_y(\mathbf{vvh}) \\ &\quad - g\mathbf{h}D_y\boldsymbol{\eta} - f\mathbf{uh}, \end{aligned} \quad (6)$$

where we have regrouped terms for later convenience. For notational brevity, we adopt the convention that vector products of the form \mathbf{ab} refer to the Schur product, i.e.,

$$\mathbf{ab} = [a_1b_1, \dots, a_Nb_N]^\top.$$

The question that remains is how to choose the time-discretization to allow for a stable and efficient scheme. The most obvious choice is to apply the same numerical ODE integrator to all instances of $\frac{d}{dt}$ in equations (4)-(5). If we discretize the flow variables $(\mathbf{h}, \mathbf{uh}, \mathbf{vh})$ at the time levels

$$t_n = n\Delta t, \quad k = 0, 1, \dots, \quad (7)$$

where Δt represents the time-step, and we adopt the notation that superscript n denotes the n^{th} time-step. Applying the Leapfrog formula to

equations (4)-(6) results in the scheme

$$\mathbf{h}^{n+1} = \mathbf{h}^{n-1} + 2\Delta t(-D_x(\mathbf{uh})^n - D_y(\mathbf{vh})^n), \quad (8)$$

$$\begin{pmatrix} I - \frac{\mathcal{H}^2}{6}D_{xx} & -\frac{\mathcal{H}^2}{6}D_{xy} \\ -\frac{\mathcal{H}^2}{6}D_{xy} & I - \frac{\mathcal{H}^2}{6}D_{yy} \end{pmatrix} \begin{pmatrix} (\mathbf{uh})^{n+1} \\ (\mathbf{vh})^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{RHS}_1^{n,n-1} \\ \mathbf{RHS}_2^{n,n-1} \end{pmatrix}, \quad (9)$$

117 where

$$\mathbf{RHS}_1^{n,n-1} = (\mathbf{uh})^{n-1} - \frac{\mathbf{H}^2}{6}D_{xx}(\mathbf{uh})^{n-1} - \frac{\mathbf{H}^2}{6}D_{xy}(\mathbf{vh})^{n-1} \quad (10)$$

$$+ 2\Delta t(-D_x(\mathbf{uuh})^n - D_y(\mathbf{uvh})^n - g\mathbf{h}^n D_x \boldsymbol{\eta}^n + f(\mathbf{vh})^n),$$

$$\mathbf{RHS}_2^{n,n-1} = (\mathbf{vh})^{n-1} - \frac{\mathbf{H}^2}{6}D_{xy}(\mathbf{uh})^{n-1} - \frac{\mathbf{H}^2}{6}D_{yy}(\mathbf{vh})^{n-1} \quad (11)$$

$$+ 2\Delta t(-D_x(\mathbf{vuh})^n - D_y(\mathbf{vvh})^n - g\mathbf{h}^n D_y \boldsymbol{\eta}^n - f(\mathbf{uh})^n),$$

118 $\mathcal{H}_{ii} = H_i$ is the $N \times N$ matrix with the entries of $\mathbf{H} = [H_1, \dots, H_N]^T$ along
 119 its diagonal, and I is the $N \times N$ identity matrix. Due to the coupled nature
 120 of the semi-discrete momentum equations (5)-(6), a block matrix of size
 121 $2N \times 2N$ appears in the scheme despite our choice of an explicit numerical
 122 ODE integrator. An approach for reducing the dimension of the required
 123 linear system by a factor of 2 is discussed below.

124 2.2.1. The Scalar Approach

125 Although there is nothing wrong with the scheme represented by (8)-(9),
 126 it is desirable to find an alternative scheme that involves solving a smaller
 127 linear system of equations, if possible. Such a scheme can be obtained
 128 by adding an auxiliary elliptic equation to the Boussinesq system. The
 129 resulting linear system is $N \times N$. This was demonstrated by Eskilsson and
 130 Sherwin (2005) where the DG-FEM method was used to solve the equations
 131 of Peregrine (1967) that are similar to the system (1)-(3).

The approach begins by introducing the scalar variable

$$z = \nabla \cdot (\mathbf{uh})_t, \quad (12)$$

which represents the time rate of change of momentum divergence. If we then take the divergence of the vector form of the momentum equations (2)-(3), we arrive at the elliptic equation

$$\nabla \cdot \left(\frac{H^2}{6} \nabla z \right) - z = -\nabla \cdot \mathbf{a}, \quad (13)$$

that is referred to as a *wave continuity* equation by Eskilsson and Sherwin (2005). The vector $\mathbf{a} = (a_1, a_2)^T$ is given by the flux terms in equation (2)-(3), i.e.,

$$\mathbf{a} = \begin{pmatrix} -\nabla \cdot ((uh)\mathbf{u}) - gh\eta_x + fvh \\ -\nabla \cdot ((vh)\mathbf{u}) - gh\eta_y - fuh \end{pmatrix}. \quad (14)$$

Applying the method of lines to the augmented system represented by equations (1)-(3) and (13) gives the semi-discrete equations

$$\frac{d\mathbf{h}}{dt} = -D_x(\mathbf{uh}) - D_y(\mathbf{vh}), \quad (15)$$

$$\frac{d(\mathbf{uh})}{dt} = -D_x(\mathbf{uuh}) - D_y(\mathbf{uvh}) - ghD_x\boldsymbol{\eta} + f\mathbf{vh} + \frac{\mathbf{H}^2}{6}D_x\mathbf{z}, \quad (16)$$

$$\frac{d(\mathbf{vh})}{dt} = -D_y(\mathbf{vuh}) - D_y(\mathbf{vvh}) - ghD_y\boldsymbol{\eta} - f\mathbf{uh} + \frac{\mathbf{H}^2}{6}D_y\mathbf{z}, \quad (17)$$

$$\begin{aligned} \frac{\mathbf{H}^2}{6}(D_{xx}\mathbf{z} + D_{yy}\mathbf{z}) - \mathbf{z} \\ + \frac{1}{6}(D_x(\mathbf{H}^2)D_x\mathbf{z} + D_y(\mathbf{H}^2)D_y\mathbf{z}) = -(D_x\mathbf{a}_1 + D_y\mathbf{a}_2), \end{aligned} \quad (18)$$

where we have first applied the product rule to equation (13) in arriving at (18). The left-hand side of equation (18) may be factored to resemble a linear system of equations of the form

$$\mathcal{A}\mathbf{z} = \mathbf{b}, \quad (19)$$

with

$$\mathcal{A} = \frac{\mathcal{H}^2}{6}(D_{xx} + D_{yy}) - I + \frac{1}{6}(D_x(\mathcal{H}^2)D_x + D_y(\mathcal{H}^2)D_y), \quad (20)$$

$$\mathbf{b} = -(D_x\mathbf{a}_1 + D_y\mathbf{a}_2). \quad (21)$$

We can then obtain an appropriate numerical scheme by applying the Leapfrog formula to equations (15)-(17) and using time-splitting so that the equation for \mathbf{z} may be inverted using the most recent information avail-

able. The resulting scheme at each time-step is

$$\mathbf{h}^{n+1} = \mathbf{h}^{n-1} + 2\Delta t(-D_x(\mathbf{uh})^n - D_y(\mathbf{vh})^n), \quad (22)$$

$$(\mathbf{uh})^\dagger = (\mathbf{uh})^{n-1} + 2\Delta t \mathbf{a}_1^n, \quad (23)$$

$$(\mathbf{vh})^\dagger = (\mathbf{vh})^{n-1} + 2\Delta t \mathbf{a}_2^n, \quad (24)$$

$$\mathbf{z}^\dagger = \mathcal{A}^{-1} \mathbf{b}^\dagger, \quad (25)$$

$$(\mathbf{uh})^{n+1} = (\mathbf{uh})^\dagger + 2\Delta t \frac{\mathbf{H}^2}{6} D_x \mathbf{z}^\dagger, \quad (26)$$

$$(\mathbf{vh})^{n+1} = (\mathbf{vh})^\dagger + 2\Delta t \frac{\mathbf{H}^2}{6} D_y \mathbf{z}^\dagger, \quad (27)$$

where \mathbf{b}^\dagger is the vector \mathbf{b} evaluated using $(\mathbf{uh})^\dagger$, $(\mathbf{vh})^\dagger$, and \mathbf{h}^{n+1} . An alternative method that requires fewer computations at the cost of slightly worse accuracy is to compute \mathbf{z}^\dagger first using only information from the n^{th} time-step, and then to compute $(\mathbf{h}^{n+1}, (\mathbf{uh})^{n+1}, (\mathbf{vh})^{n+1})$ without time-splitting. Our numerical experiments revealed negligible differences between the two methods.

The most expensive part of the algorithm is in step (25), solving the linear system $\mathcal{A}\mathbf{z} = \mathbf{b}$. For pseudospectral methods, the matrix \mathcal{A} is dense, and due to memory restrictions, direct methods such as LU-factorizations become impractical at high resolutions (Boyd, 2001). To overcome this issue, it is necessary to consider iterative methods such as the generalized minimum residual method (GMRES) and pre-conditioning to reduce the required number of iterations. In Section 2.3.2, we illustrate how to construct a suitable pre-conditioner using a finite differences approximation.

The schemes presented above are not self-starting. Therefore, they must be started by taking either a single time-step with the first order accurate Forward Euler method or a higher order Runge-Kutta method.

2.3. Fourier Spatial Discretization Method

We now present the Fourier spatial discretization method applied to the scheme represented by equations (22)-(27). We begin by discretizing the periodic rectangular domain $\Omega = [0, L_x] \times [0, L_y]$ by constructing a tensor-product grid from the one-dimensional equidistant grids

$$x_i = i\Delta x, \quad i = 0, \dots, N_x - 1, \quad (28)$$

$$y_j = j\Delta y, \quad j = 0, \dots, N_y - 1, \quad (29)$$

where $\Delta x = L_x/N_x$ and $\Delta y = L_y/N_y$ represent the grid spacing in the x and y directions, respectively. The resulting two-dimensional grid then

has $N = N_x N_y$ total grid points. It is also useful to define the discrete wavenumber vectors \mathbf{k} and \mathbf{l} defined as

$$k_i = \frac{2\pi}{L_x} i, \quad i = 0, \dots, N_x - 1, \quad (30)$$

$$l_j = \frac{2\pi}{L_y} j, \quad j = 0, \dots, N_y - 1. \quad (31)$$

150 Rather than using differentiation matrices to compute the approximate
 151 derivatives in the schemes presented above, we employ the “pseudospectral
 152 technique” as described by Peyret (2002). That is, differentiation is per-
 153 formed in the spectral space (the space of the Fourier coefficients) with the
 154 fast discrete Fourier transform (FFT) while products are performed in the
 155 physical space. Doing so allows one to avoid the expense of directly com-
 156 puting convolution sums in the space of the Fourier coefficients, as the non-
 157 linear terms would require. Pseudospectral differentiation is also faster than
 158 explicitly calculating matrix-vector products that require $O(N^2)$ floating-
 159 point operations (FLOPS) since the FFT requires $O(N \log N)$ FLOPS, and
 160 Schur products requires $O(N)$ FLOPS.

For the purposes of pseudospectral differentiation, it is useful to consider the flow fields as $N_y \times N_x$ matrices instead of $N_x N_y \times 1$ vectors. For a given discretized field ϕ which may represent a flow variable or a product of flow variables, we approximate its discrete derivatives as

$$\phi_x = \mathcal{F}_x^{-1} (\mathbf{i} \mathcal{K} \mathcal{F}_x(\phi)) , \quad (32)$$

$$\phi_y = \mathcal{F}_y^{-1} (\mathbf{i} \mathcal{L} \mathcal{F}_y(\phi)) , \quad (33)$$

$$\phi_{xx} = \mathcal{F}_x^{-1} (-\mathcal{K}^2 \mathcal{F}_x(\phi)) , \quad (34)$$

$$\phi_{yy} = \mathcal{F}_y^{-1} (-\mathcal{L}^2 \mathcal{F}_y(\phi)) , \quad (35)$$

$$\phi_{xy} = \mathcal{F}_y^{-1} (\mathbf{i} \mathcal{L} \mathcal{F}_y(\phi_x)) , \quad (36)$$

161 where \mathcal{F}_x and \mathcal{F}_y represent the discrete Fourier transforms with respect to
 162 x and y, respectively, $\mathbf{i} = \sqrt{-1}$, and superscript -1 refers to the inverse
 163 transform. The wavenumber matrices \mathcal{K} and \mathcal{L} are of size $N_y \times N_x$ with
 164 entries $\mathcal{K}_{ij} = k_j$, $\mathcal{L}_{ij} = l_i$. All of the products in (32)-(36) are Schur
 165 products.

166 The underlying assumptions used in this spatial discretization are that
 167 the solution fields are smooth and periodic in space in both directions, and
 168 they are hence well represented by a sinusoidal basis. Given these assump-
 169 tions, the Fourier pseudospectral spatial discretization method guarantees

170 an exponential convergence rate (Boyd, 2001). If one or both of these as-
 171 sumptions are broken, Gibbs oscillations are introduced into the solution
 172 and the convergence rate is reduced to polynomial order.

173 2.3.1. Solving the Linear System

174 In order to solve the linear system (19), one may be tempted to explic-
 175 itly build the large matrix \mathcal{A} using two-dimensional spectral differentiation
 176 matrices. However, this is typically not a good idea due to memory restric-
 177 tions. Two-dimensional spectral differentiation matrices can be built from
 178 kronecker products between the 1D differentiation matrix and the appropri-
 179 ate identity matrix, and require $O(N_x N_y (N_x + N_y))$ memory. If mixed spa-
 180 tial derivatives are required, the situation can be the worst case, $O(N_x^2 N_y^2)$
 181 memory, which is certainly not reasonable. It is clear that indirect meth-
 182 ods for solving the system (19) are required in the case of a pseudospectral
 183 spatial discretization.

In doubly-periodic cases with a flat bottom, the mean depth H is a constant and the linear system (19) may be solved efficiently using the pseudospectral technique by first taking its Fourier transform, yielding

$$\widehat{\mathcal{A}}\mathcal{F}_{xy}(\mathbf{z}) = \mathcal{F}_{xy}(\mathbf{b}) , \quad (37)$$

where

$$\widehat{\mathcal{A}} = \left(-\frac{H^2}{6} (\mathcal{K}^2 + \mathcal{L}^2) - \mathbf{1} \right) , \quad (38)$$

$\mathbf{1}$ is the $N_y \times N_x$ matrix of all ones, and \mathcal{F}_{xy} is the double discrete Fourier transform. To solve the system, we take the Schur product of both sides with $\widehat{\mathcal{A}}^{-1}$, defined as

$$\widehat{\mathcal{A}}_{ij}^{-1} = \frac{1}{\widehat{\mathcal{A}}_{ij}} , \quad (39)$$

the multiplicative inverse of $\widehat{\mathcal{A}}$. Hence,

$$\mathbf{z} = \mathcal{F}_{xy}^{-1} \left(\widehat{\mathcal{A}}^{-1} \mathcal{F}_{xy}(\mathbf{b}) \right) . \quad (40)$$

184 This situation is ideal, since we are able to effectively solve a large, dense lin-
 185 ear system with $O(N_x N_y (N_x + N_y))$ entries in $O(N_x N_y \log(N_x N_y))$ FLOPS.
 186 In cases where the bottom is not flat, the technique represented by (37)-(38)
 187 is not available since point-wise products become convolutions in Fourier
 188 space, so another method must be sought.

189 Iterative linear system solutions appear to be our only course of action
 190 in the case of variable depth. Since the Krylov subspace methods do not
 191 explicitly require the entries of the matrix \mathcal{A} (Golub and Van Loan, 1996),
 192 they are a clear choice. Furthermore, given that the matrices being solved
 193 are not guaranteed to be symmetric nor skew-symmetric (Trefethen, 2000),
 194 a good choice of iterative linear solver is the generalized minimum residual
 195 method (GMRES) (Trefethen and Bau, 1997).

196 The main difficulty with using such iterative solvers, is that the linear
 197 systems to be solved can be quite poorly conditioned, driving the number
 198 of iterations to be on the same order as the problem’s dimension. This issue
 199 typically gets worse at higher resolutions (Boyd, 2001). To overcome this,
 200 it is useful to pre-condition the linear system to obtain convergence at a
 201 relatively small number of iterations as discussed below.

202 2.3.2. Finite Differences Pre-Conditioner

203 Since the linear system to be solved is the result of a high-order PDE
 204 spatial discretization, a popular and effective choice of pre-conditioner is a
 205 low-order spatial discretization of the PDE (Trefethen and Bau, 1997).

206 A finite differences discretization is a natural choice since it allows one
 207 to fix the order of approximation independently of the number of grid
 208 points and the grid-spacing used (Leveque, 2007). The resulting spatial-
 209 discretization operators are typically very sparse and banded, and as a re-
 210 sult can be solved or factored quite easily using sparse matrix manipulation
 211 software libraries.

212 To construct a pre-conditioner for solving the linear system (19), we
 213 employ the second-order centered differences formulas given by Leveque
 214 (2007) to construct the $N \times N$ differentiation matrices $D_x^{(2)}$, $D_y^{(2)}$, $D_{xx}^{(2)}$,
 215 $D_{yy}^{(2)}$, where superscript “(2)” refers to the order of approximation used. A
 216 second order approximation to the matrix \mathcal{A} , denoted $\mathcal{A}^{(2)}$, can then be
 217 constructed using the formula (20). The resulting matrix is pentadiagonal,
 218 requiring $O(5N)$ memory since its construction relies on the 5-point finite
 219 differences stencil for the Laplacian (Iserles, 1996).

Since $\mathcal{A}^{(2)}$ is an approximation of \mathcal{A} , we can imagine left-multiplying
 (19) by $(\mathcal{A}^{(2)})^{-1}$

$$(\mathcal{A}^{(2)})^{-1} \mathcal{A} \mathbf{z} = (\mathcal{A}^{(2)})^{-1} \mathbf{b} , \quad (41)$$

220 to obtain a more well-conditioned linear system since $(\mathcal{A}^{(2)})^{-1} \mathcal{A} \approx I$. Of
 221 course, this is merely illustrative since \mathcal{A} is not explicitly built and comput-
 222 ing the explicit inverse of $\mathcal{A}^{(2)}$ is impractical. Instead, the fact that we are

223 using the GMRES method requires that linear systems of the form $\mathcal{A}^{(2)}\tilde{\mathbf{z}} = \tilde{\mathbf{b}}$
 224 be solved at each iteration. In order to ensure linear systems of this form
 225 can be solved effectively, it is useful to compute the LU-factorization of $\mathcal{A}^{(2)}$
 226 in the pre-processing stage and to simply reuse its factors at each GMRES
 227 iteration.

It has been found that using the factors returned by the sparse-LU factorization routine provided in the UMFPACK library yield the fastest solutions to $\mathcal{A}^{(2)}\tilde{\mathbf{z}} = \tilde{\mathbf{b}}$. In addition to the lower- L and upper-triangular U factors, partial pivoting is also performed with a permutation matrix P and column-reordering matrix Q so that

$$P\mathcal{A}^{(2)}Q = LU . \quad (42)$$

228 The main cost of using this technique is in storing the factors L and
 229 U , which in the worst case, can be the same cost as storing a full $N \times N$
 230 matrix. At high resolutions, storing the factors may become unfeasible, and
 231 incomplete LU-factorizations may be used instead with a drop-tolerance
 232 tuned to give a balance between memory usage and iteration count. At
 233 even higher resolutions, such a balance may not exist, and more memory
 234 efficient techniques such as geometric multigrid (Trottenberg et al., 2000)
 235 or multi-level domain decomposition algorithms (Smith et al., 2004) should
 236 be considered.

237 2.4. Filter Stabilization of Aliasing-driving Instabilities

238 The equations do not possess any viscosity terms and thus lack any phys-
 239 ical energy dissipation mechanism. As a result, the quadratic nonlinearity
 240 terms can cause energy to accumulate at the small scales in an unphysical
 241 manner. Additionally, aliasing errors that occur due to the “pointwise prod-
 242 uct” treatment of the nonlinear terms can drive weak numerical instabilities
 243 that can destroy the numerical solutions (Hesthaven and Warburton, 2008).

In light of these issues, filtering is taken as a procedure to dissipate energy as it accumulates at the small scales and to prevent aliasing errors from driving weak instabilities. This can be achieved by applying a low-pass wavenumber filter of the form

$$\sigma(k) = \begin{cases} 1, & 0 \leq k < k_{crit} \\ \exp\left(-\alpha\left(\frac{k-k_{crit}}{k_{max}-k_{crit}}\right)^s\right), & k_{crit} \leq k \leq k_{max} \end{cases} \quad (43)$$

244 in each direction in spectral space to the solution fields after each time-step.
 245 A similar filter is used by Hesthaven and Warburton (2008) in the nodal

246 DG-FEM framework. Typical parameters used in the simulations presented
 247 in Section 3 are $k_{crit} = 0.65k_{max}$, $s = 4$, $\alpha = 18.4$, where k_{max} is the Nyquist
 248 wavenumber. The parameters α , s , and k_{crit} are tunable and, in general,
 249 their values must be determined through experimentation.

250 3. Results and Discussion

251 3.1. Pre-conditioner performance test

252 To confirm the performance of our pre-conditioned GMRES algorithm
 253 for solving equation (18), we have compared it to the GMRES method
 254 without pre-conditioning. The problem we consider for inversion here cor-
 255 responds to the first time-step of the full simulation presented below in
 256 Section 3.6. Hence, the variable coefficients on the left-hand side as well
 257 as the function on the right-hand side of (18) are both non-trivial. The
 258 convergence criterion was taken to require the magnitude of the relative
 259 residual to be below 10^{-9} , and this value was also used for all simulations
 260 with a non-flat bottom, i.e., whenever GMRES was used. The problem was
 261 solved at a variety of grid resolutions, ranging from 16×16 to 1024×1024 .

262 Iteration counts for both the ‘GMRES with pre-conditioning’ (GM-
 263 RESP) and ‘GMRES without pre-conditioning’ (GMRESNP) methods along
 264 with the ratio of their run-times are listed at all resolutions considered in
 265 Table 1. The corresponding tests were carried out in Matlab, and it was
 266 found that the values for the run-times fluctuated somewhat due to the per-
 267 formance of Matlab’s built-in parallelization’s dependence on processor
 268 load. Hence, the run-time values used in Table 1 were averaged over ten
 269 runs to average out these fluctuations.

270 Table 1 shows the pre-conditioner’s ability to keep the iteration count
 271 relatively low in comparison to the case where pre-conditioning is not used.
 272 At low resolutions (128×128 and below), the reduction in iteration count
 273 does not overcome the computational cost of using a pre-conditioner since
 274 the run-time ratio is less than one. However, at higher resolutions the
 275 savings are considerable, and the high iteration count at high resolutions
 276 makes the ‘GMRESNP’ method unpractical for use in simulations due to
 277 the unreasonable amount of computational time required. For example, the
 278 GMRESNP method at 1024×1024 resolution typically took about 280 s to
 279 converge.

Resolution	GMRESP	GMRESNP	Run-Time Ratio
	Iteration Count	Iteration Count	
16×16	5	5	0.71
32×32	6	7	0.77
64×64	9	10	0.78
128×128	9	15	0.94
256×256	12	32	1.75
512×512	16	70	4.27
1024×1024	19	147	16.2

Table 1: Iteration count vs. grid resolution for the ‘GMRES with pre-conditioning’ (GMRESP) and ‘GMRES without pre-conditioning’ (GMRESNP) methods. At each resolution, the run-time ratio is given by the time taken for the GMRESNP method to converge divided by time taken for the GMRESP method to converge.

3.2. Fourier method versus DG-FEM in 1D

In addition to the Fourier method, we have also obtained solutions to the one-dimensional form of the system (1)-(3) with a flat bottom using the nodal discontinuous Galerkin finite element method (DG-FEM). High order nodal and modal DG-FEM solutions to Boussinesq-type systems have been previously obtained by Engsig-Karup et al. (2006) and Eskilsson and Sherwin (2005), respectively. The interested reader may find the details of the DG-FEM solution procedure to the 1D system in Appendix A. The main reason we seek DG-FEM solutions here is to illustrate how a global spatial discretization method (Fourier) compares to a local spatial discretization method (DG-FEM) at various orders of approximation.

To perform this comparison, we have decided to run a simulation where a packet of short waves of two distinct wavelengths is released from rest. The domain was taken to be periodic and $L_x = 4000$ m in length, the depth was fixed at $H = 5$ m, and the acceleration due to gravity was taken to be $g = 9.81 \text{ m s}^{-2}$. The initial condition is

$$\eta(x, 0) = \eta_0 \cos(0.15x) \cos(0.05x) e^{-5\left(\frac{x-0.5L_x}{400}\right)^2}, \quad (44)$$

$$u(x, 0) = 0, \quad (45)$$

where $\eta_0 = 0.1$ m. The amplitude of the wave packet was chosen to be small enough so that linear wave theory would be a good predictor of the group speeds. This was confirmed by solving the linearized equations exactly in Fourier space and comparing with the numerical solution (not shown). Due

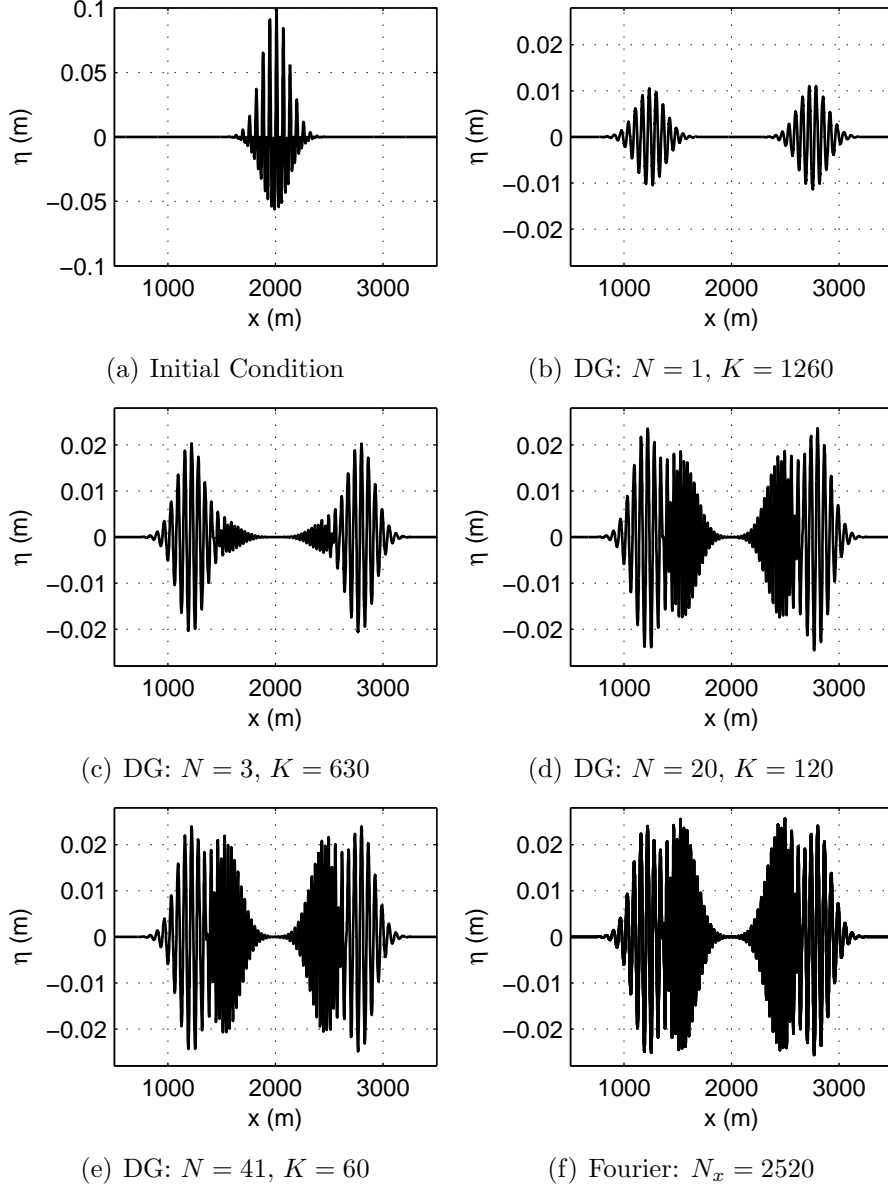


Figure 1: Fixed time snapshots of the free surface displacement at various orders of approximation for the *1D dispersive short-waves* run. Panels (b)-(f) are all at time $t = 100$ s. (a) η at $t = 0$. (b) DG-FEM $N = 1$ result. (c) DG-FEM $N = 3$ result. (d) DG-FEM $N = 20$ result. (e) DG-FEM $N = 41$ result. (f) Fourier Method with $N_x = 2520$ grid points result.

295 to dispersion, we expect the longer waves to overtake and lead the shorter
 296 waves, after sufficient time has passed, since the linear group speed of the
 297 longer waves is $c_g \approx 9.31 \text{ m s}^{-1}$ while the group speed of the shorter waves is
 298 $c_g \approx 6.14 \text{ m s}^{-1}$. This run was also used to validate the numerical methods
 299 in the regime where nonlinear effects are negligible and the bottom is flat.

300 In Figure 1, the results of the runs are displayed at various orders of
 301 accuracy. The values of K (total number of elements) and N (order of the
 302 basis functions) were chosen such that the total number of points used in
 303 the DG-FEM method would be fixed at $N_{dof} = K(N + 1) = 2520$. Modal
 304 filtering was not used in any of the runs, since the choices of small-amplitude
 305 waves and a flat bottom remove most, if not all, of the sources of nonlinearity
 306 and aliasing errors. A striking observation is that for the low-order runs,
 307 the shorter waves are dissipated to a very large degree, and in the $N = 1$
 308 case, in effect entirely.

In Figure 2, the time series of the domain-integrated total energy, defined
 by

$$E(t) = \int_0^{L_x} \frac{1}{2} h u^2 + \frac{1}{2} g \eta^2 dx , \quad (46)$$

309 is plotted for each of the cases shown in Figure 1. It can be shown that
 310 the Boussinesq system (1)-(3) does not conserve energy (as is true of most
 311 Boussinesq-type systems), and even exact solutions would not satisfy the
 312 physical property $E(t) = \text{constant}$. Indeed, the plots in Figure 2 reveal that
 313 in the absence of numerical dissipation, $E(t)$ is oscillatory. Once again,
 314 this fact has been confirmed by comparing to the exact solution of the
 315 linearized equations in Fourier space (and invoking Parseval's theorem).
 316 This oscillatory behaviour is a well-known consequence of using Boussinesq-
 317 type systems, and Boussinesq-type systems that conserve energy exactly
 318 have been proposed (Christov, 2000). However, these energy-conserving
 319 systems tend to be undesirable for numerical integration due to the presence
 320 of third-order spatial derivatives.

321 The plots reveal the difference in numerical dissipation between the DG-
 322 FEM method at different orders when compared to the Fourier method.
 323 Even at a very high order of $N = 41$, the DG-FEM method cannot match
 324 the energy-conserving qualities of the Fourier method. This fact is likely
 325 owed to the numerical dissipation introduced by using the numerical flux
 326 function (A.8) that is only an *approximate* Riemann solver, and for stability,
 327 is chosen to contribute a non-positive value to the global energy balance at
 328 each time-step (Hesthaven and Warburton, 2008). Regardless of this fact,

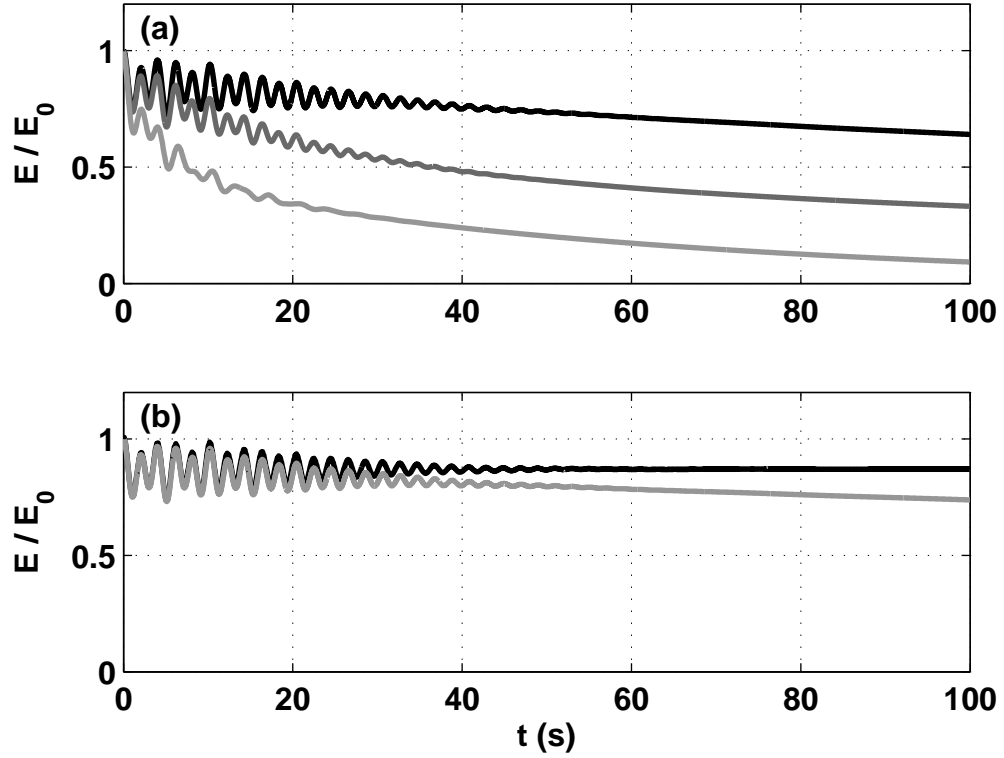


Figure 2: Domain-integrated total energy time series for the *1D dispersive short-waves* run with (a) the DG-FEM method at orders $N = 1$ (light grey), $N = 3$ (dark grey), and $N = 20$ (black), and (b) the DG-FEM method at order $N = 41$ (grey) and the Fourier method with $N_x = 2520$ points (black). The domain-integrated total energy E has been scaled by E_0 , its value at $t = 0$. The number of grid points (degrees of freedom) is fixed at $N_{dof} = 2520$ in all cases.

for a fixed number of degrees of freedom ($N_{dof} = 2520$), one still expects the DG-FEM method's result to converge to the Fourier method's result in the high-order limit $(N, K) = (2519, 1)$ where the number of interior elemental interfaces is zero.

It was thought that a more accurate choice of approximate Riemann solver for the advective (flux gradient) terms, such as the Harten-Lax-Van Leer solver modified for contact waves (HLLC) used in Eskilsson and Sherwin (2005), would improve the energy-conserving qualities of the DG-FEM solutions shown here. Upon implementing the HLLC numerical flux, however, significant improvements to the solutions were only found in low-order simulations ($N = 1$) (not shown). This apparent insensitivity to the choice of approximate Riemann solver is undoubtedly owed to the dispersive terms in our model equations that result in solutions that are more regular than those obtained from the traditional (hydrostatic) shallow water model.

3.3. Comparison of numerical code to approximate analytical solutions

In the next step towards validating our numerical methodology for the Fourier spatial discretization method, we compared numerical solutions obtained from our numerical code to approximate analytical solutions obtained using the WKB (Wentzel-Kramers-Brillouin) approximation for situations involving variable depth in 1D. The approximation is valid in situations where the depth H varies more slowly in space than the free surface η . Hence, we assume that H depends only on a slow coordinate. Here, we merely state the WKB solution, and relegate the full details of its derivation to Appendix B. The WKB solution is given by

$$\eta(x, y, t) \sim A(\epsilon x) e^{i(\frac{S_0}{\epsilon}(\epsilon x) - \sigma t)}, \text{ as } \epsilon \rightarrow 0, \quad (47)$$

$$u(x, y, t) = \sqrt{\frac{g}{H}} \eta(x, y, t). \quad (48)$$

where $A(\epsilon x) = a_0 H^{-\frac{1}{4}}$, a_0 is an arbitrary constant, ϵ is a small parameter, and $S_0(\epsilon x)$ is given by equation (B.7).

To compare our numerical code with the WKB solution (47), we initialized the numerical solver with the real part of the WKB solution (with $S_0 > 0$) taken at $t = 0$, stepped the solution forwards in time for five wave periods, and compared the numerical solution to the approximate analytical solution at the final time. We chose the slowly varying depth profile

$$H(\epsilon x) = H_0 - \Delta H \sin(\epsilon x), \quad (49)$$

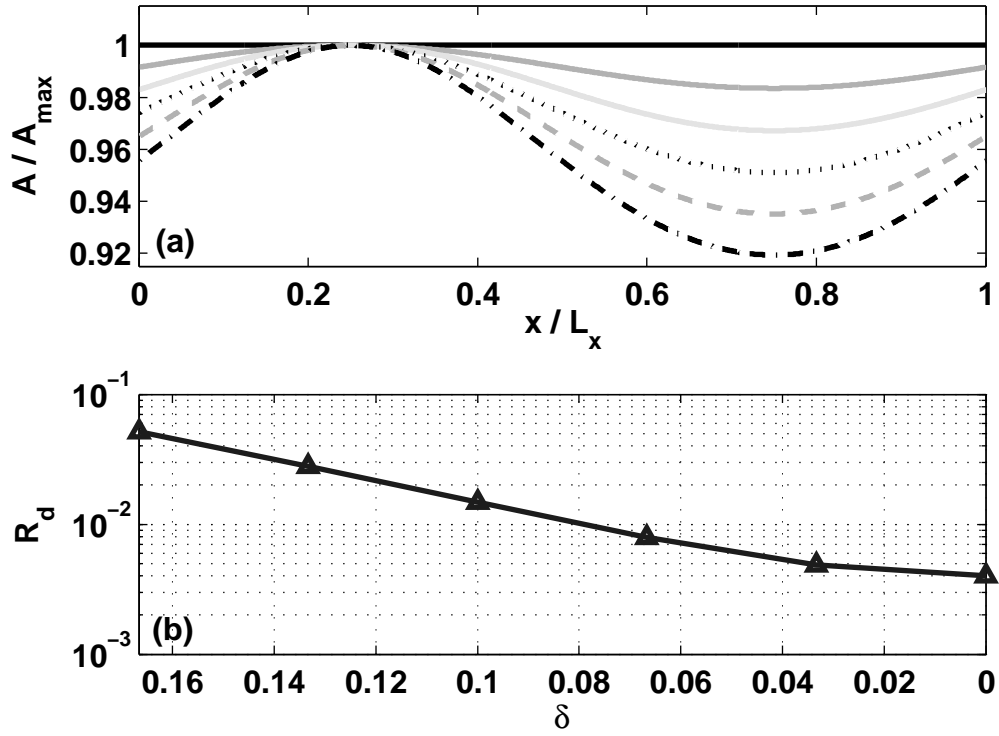


Figure 3: Panel (a): Envelopes of the WKB solution, scaled by their maximum value, for the values of $\delta = \Delta H / H_0 = 0$ (solid, black), $1/30$ (solid, dark grey), $1/15$ (solid, light grey), $1/10$ (dotted), $2/15$ (dashed), $1/6$ (dash-dotted). Panel (b): Relative difference (R_d) between the numerical solution and the WKB solution after five wave periods vs. δ .

where $\epsilon = 2\pi/L_x$ is the wavenumber of the longest sinusoidal wave that fits in the domain. Here, $L_x = 3000$ m and $H_0 = 15$ m. We have varied the parameter ΔH from 0-2.5 m, expecting the two solutions to agree best in the limit that $\Delta H \rightarrow 0$ (a flat bottom). We set $a_0 = 10^{-4}H_{min}$ in all cases to ensure that nonlinear effects in the numerical solution were negligible. The numerical grid was taken to have 1024 points (grid halving experiments suggest that the simulations are numerically converged upon reaching 256 points), and the time-step was taken to be

$$\Delta t = \frac{1}{20} \frac{\Delta x}{\sqrt{gH_0}}, \quad (50)$$

where Δx is the uniform grid spacing. The time-step was taken to be smaller than what is typically required for numerical stability. This was done in order to minimize the amount of error introduced during the numerical time-integration process.

The function $S_0(\epsilon x)$ was calculated numerically using quadrature rules for integration, since a closed-form analytical expression is not available for our choice of $H(\epsilon x)$. We chose the value

$$\sigma = \sqrt{gH_0} \left(\frac{10\pi}{L_x} \right), \quad (51)$$

for the frequency of the waves. If the bottom is flat, this choice represents the frequency of a sinusoidal wave whose wavelength is a factor of five shorter than the longest wavelength that fits in the domain.

After time-stepping was completed, the relative L^2 difference

$$R_d = \frac{\int_0^{L_x} (\eta_{Num} - \eta_{WKB})^2 dx}{\int_0^{L_x} (\eta_{WKB})^2 dx}, \quad (52)$$

was calculated where η_{Num} and η_{WKB} represent the numerical and WKB η fields, respectively. The integrals were evaluated using the Fourier expansion coefficients of each integrand (obtained with FFT).

Close agreement between the two solutions in the limit that $\Delta H \rightarrow 0$ is illustrated in Figure 3 where we have introduced the non-dimensional parameter $\delta = \Delta H/H_0$. Panel **(a)** shows the shape of the spatially dependent wave amplitude function, $H^{-\frac{1}{4}}$, for several choices of δ , and panel **(b)** shows the decline in the relative difference between the analytical and numerical solution as $\delta \rightarrow 0$. The agreement was found to improve somewhat by increasing the domain length while keeping the depth fixed, but the difference was less than an order of magnitude.

364 *3.4. Grid-convergence study using a simulation of 1D wave-topography in-*
 365 *teraction*

366 We next focus our attention on a 1D simulation of nonlinear and dis-
 367 persive waves repeatedly propagating over a ridge with the Fourier method.
 368 Since analytical solutions are not available to confirm the validity of the
 369 results, we rely on grid-doubling experiments to illustrate the method's
 370 convergence in the well-resolved limit.

We begin by considering a periodic domain of length $L_x = 2$ km. The depth profile is given by

$$H(x, y) = H_1 - \Delta H e^{-5(\frac{x-0.5L_x}{100})^4}, \quad (53)$$

with $H_1 = 10$ m and $\Delta H = 2$ m, reflecting a pre-dominantly flat bottom with a 2 m tall ridge in the center of the domain. The simulation was initialized using the initial conditions

$$\eta(x, 0) = \eta_0 e^{-(\frac{x-0.25L_x}{100})^2}, \quad (54)$$

$$u(x, 0) = \sqrt{\frac{g}{H_1}} \eta(x, 0), \quad (55)$$

with $\eta_0 = 1$ m, representing a single wave of elevation, initialized to propagate in the positive x -direction with the off-ridge long wave speed $\sqrt{gH_1}$. A schematic diagram of the initialization is shown in Figure 4(a). The governing equations were stepped forward until a final time of $t = T_{final} = 605$ s was reached. The final time was chosen such that a linear wave would traverse the length of the domain three times. The time-step was taken to be

$$\Delta t = \frac{1}{2} \frac{\Delta x}{\sqrt{gH_1}}, \quad (56)$$

371 and numerical instabilities were prevented by employing the spatial filtering
 372 methodology discussed in Section 2.4.

373 In this simulation, nonlinearity plays a key role in the evolution of the
 374 flow. The initial wave immediately begins to steepen, and the steepening is
 375 further enhanced due to shoaling as the wave propagates over the ridge (not
 376 shown). Dispersion then acts to balance the nonlinearity and prevent the
 377 formation of shocks. The final result is a collection of three solitary waves
 378 propagating in the positive x -direction followed by a dispersive wavetrain.
 379 These solitary waves are similar in shape to the $\text{sech}^2(\cdot)$ solitons predicted
 380 by Korteweg-de Vries (KdV) theory (Whitham, 1999). It can be shown that

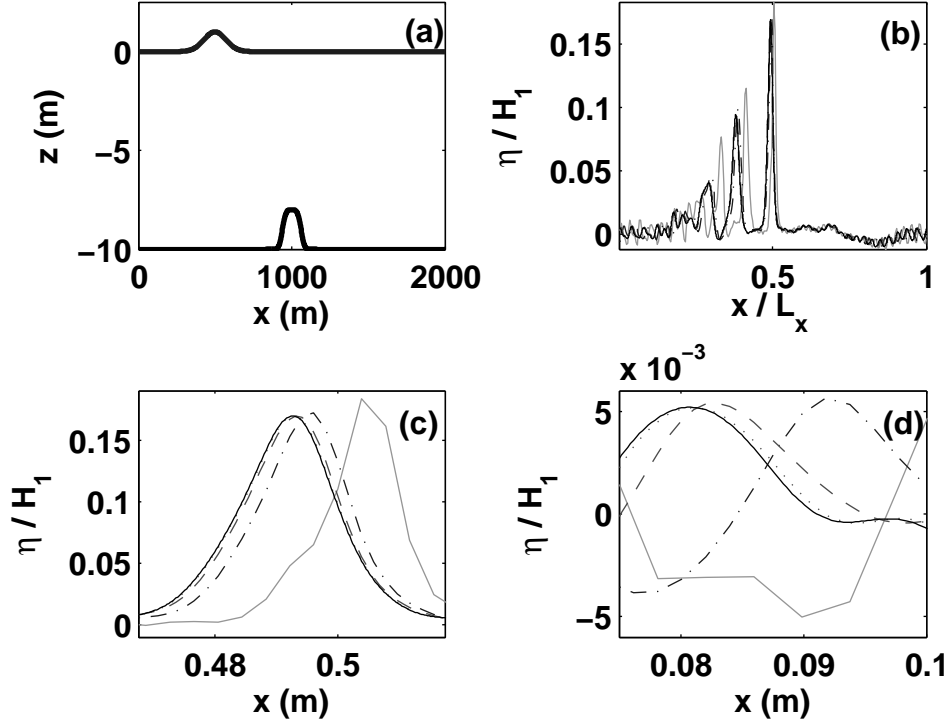


Figure 4: Results for the *1D wave-topography interaction* run. Panel (a): Plot of the initialization, showing the topography $z = -H(x)$ and the initial free surface displacement $z = \eta(x, 0)$. Panels (b)–(d): η at $t = T_{final}$ at resolutions $N_x = 256$ (solid, light grey), $N_x = 512$ (dash-dotted), $N_x = 1024$ (dashed), $N_x = 2048$ (dotted), $N_x = 4096$ (solid, black). Panel (c) is zoomed-in on the leading solitary wave, and panel (d) is zoomed-in on a section of the dispersive tail. In panels (b)–(d), the variable η has been made dimensionless by dividing by the off-ridge water depth, $H_1 = 10$ m.

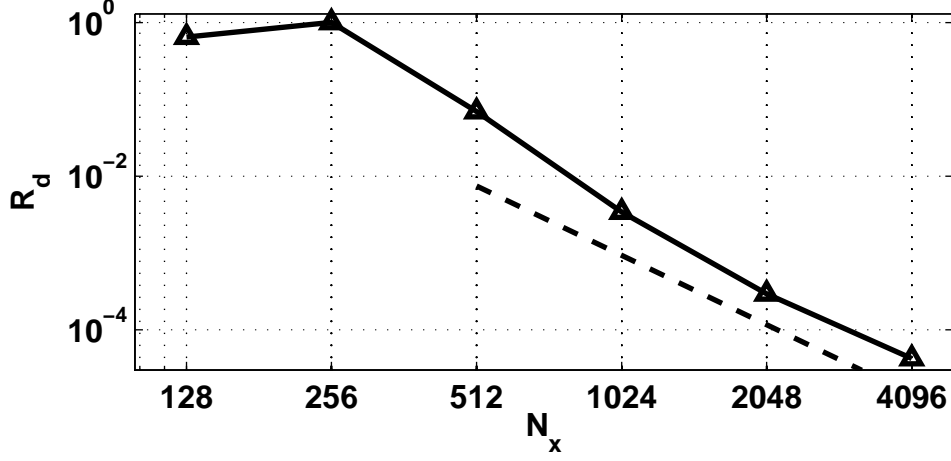


Figure 5: Relative difference (triangles) between the solution for each of the resolutions plotted in Figure 4 and the solution computed with $N_x = 8192$ points at $t = T_{final}$. The dashed line represents a curve of $O(N_x^{-3})$.

such solitons are approximate solutions to the governing equations (1)–(3) under the assumption of a flat-bottom, as was done for a similar system by Wei and Kirby (1995).

Details of the η field at $t = T_{final}$ at several resolutions is depicted in Figure 4(b)–(d). Inspecting the various plots suggests that grid convergence has been reached when $N_x = 2048$ grid points are used, since doubling the resolution once more to $N_x = 4096$ only yields minute differences in the fine-scale features of the η field (see Figure 4(d)). To verify that the numerical method is converging rapidly, in Figure 5 we have plotted the relative L^2 difference, R_d , between each of the solutions shown in Figure 4 and the solution computed with $N_x = 8192$ grid points taken at $t = T_{final}$, revealing fast convergence towards the solution with $N_x = 8192$ grid points. A slope analysis reveals a convergence rate of about 3, as is depicted in the figure.

3.5. A 2D simulation of wave generation by flow over topography

In our next test-case, we present a two-dimensional simulation of forced surface waves interacting with bottom topography to illustrate the numerical model’s applicability to real-world problems in water wave dynamics. It is quite well known that when the inflow speed approaches the long wave speed, upstream propagating nonlinear waves are generated. This process

401 is referred to as resonant generation (Grimshaw and Smyth, 1986). Non-
 402 dispersive shallow water dynamics for flow over axisymmetric obstacles has
 403 been discussed by Esler et al. (2007) using finite volume methods.

The physical parameters were set to: $g = 9.81 \text{ m s}^{-2}$, $f = 0$ (no rotation), and $L_x = L_y = 2 \text{ km}$, reflecting a (periodic) square domain. The grid was taken to have 2048 points in the x -direction and 256 points in the y -direction. Modal filtering in each direction was carried out using the parameters discussed in Section 2.4. The depth profile was taken to be

$$H(x, y) = H_1 - \Delta H e^{-5(\frac{x-0.5L_x}{100})^4 - 5(\frac{y-0.5L_y}{200})^4}, \quad (57)$$

with $H_1 = 20 \text{ m}$ and $\Delta H = 2 \text{ m}$. This is essentially a two-dimensional version of the depth-profile used in Section 3.4, i.e., a predominantly flat profile with a square-shaped ridge in the center of the domain. The simulation was initialized from quiescent conditions and forced by adding the body forcing term hF_x to the right-hand side of equation (2), where

$$F_x = \begin{cases} \frac{\sqrt{gH_1}}{\beta}, & 0 \leq t < 10 \text{ s} \\ 0, & t \geq 10 \text{ s} \end{cases} \quad (58)$$

404 and $\beta = 50/3 \text{ s}$ is a time-scale. The forcing is constant in space and piece-
 405 wise constant in time. Its effect is to induce a flow over the topography in the
 406 positive x -direction, that is constant upstream of the topography. The value
 407 of β was chosen so that the final upstream velocity is equal to three-fifths
 408 of the off-ridge long wave speed, and hence the flow is formally sub-critical.
 409 Since the addition of body forcing simply represents a source term in the
 410 governing equations, it was added to the time-stepping procedure using a
 411 straight-forward explicit evaluation.

412 Snapshots of the developing η -field are shown in Figure 6. In addition
 413 to a trapped wave of depression generated over the ridge, an upstream-
 414 propagating wavefront of elevation (with a slightly depressed tail, or possi-
 415 bly a second wave) can be seen emanating from the ridge, and travelling
 416 westward. This wavefront can be seen losing amplitude as time progresses.
 417 This is due to radial spreading, or in other words geometric decay. The
 418 extent of this decrease in amplitude (energy density) due to geometric decay
 419 can be illustrated qualitatively by comparing this 2D simulation to an
 420 analogous 1D simulation where variations in the y -direction are neglected.
 421 This comparison is carried out in Figure 7, where it can be seen that in the
 422 1D case (panel **(b)**), the upstream-propagating wave front better retains its

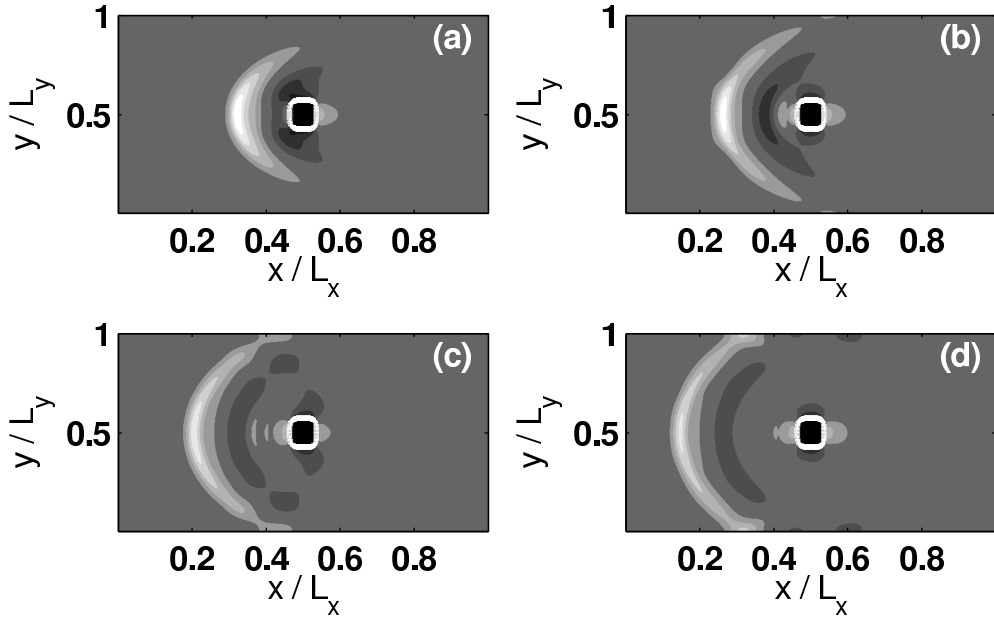


Figure 6: Fixed time snapshots of the free surface displacement at (a) $t = 60$ s, (b) $t = 80$ s, (c) $t = 100$ s, and (d) $t = 120$ s in the *2D wave generation by flow over topography* run. The contours shown have values of $\eta = \pm 0.1$ m, ± 0.2 m, ± 0.3 m, ± 0.4 m, ± 0.5 m, ± 0.6 m, ± 1 m, where $\eta = -1$ m is shown in black and $\eta = +1$ m is shown in white. The solid-white line is the depth contour $H = 19.5$ m, indicating the location of the ridge.

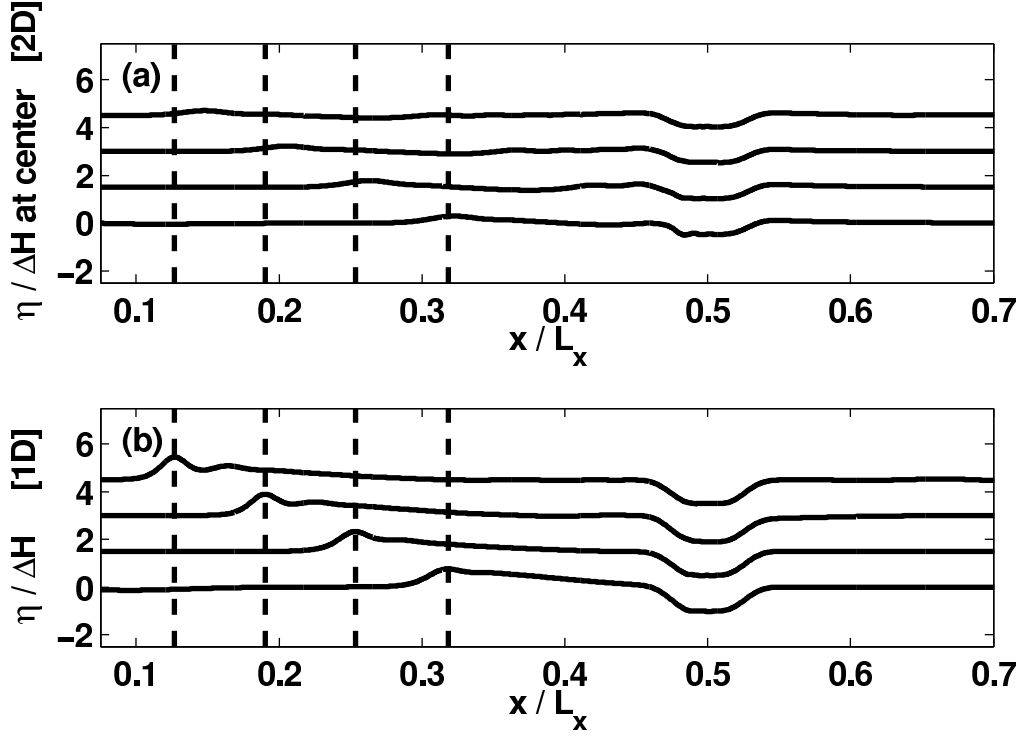


Figure 7: Panel (a): 1D slices of the snapshots presented in Figure 6 through the line $y = 1$ km. Panel (b): Snapshots of the η field for an analogous 1D simulation, where variations in y have been neglected. In each panel, a single curve corresponds to a time in Figure 6, with the lowest curve giving a slice through the snapshot taken at $t = 60$ s and the uppermost curve giving a slice through the snapshot taken at $t = 120$ s. Each curve has been shifted upwards by $3/40 (t - 60)$ units. Dashed vertical lines represent the location of the maximum height of the upstream-propagating wavefront at each snapshot from the 1D simulation. The variable η has been made dimensionless by dividing by the ridge height, $\Delta H = 2$ m.

423 amplitude than in the 2D case (panel **(a)**) since there is no radiation in the
 424 y -direction.

425 3.6. A 2D simulation of wave propagation over a shoal

In our final test case, we follow the evolution of a wave front with an initially one-dimensional shape as it propagates over a shoal that is partitioned by a deep region in the center of the domain. The physical parameters (g, f, L_x, L_y) were chosen to be the same as in the previous test case (see Section 3.5), as were the filtering parameters. The grid was taken to be have 1024 points in each direction. The depth-profile was taken to be

$$H = H_1 - \Delta H \left[\operatorname{sech} \left(\frac{y - 0.5L_y}{500} \right) - e^{-\left(\frac{r_0}{200}\right)^2} \right]. \quad (59)$$

Here, $r_0 = \sqrt{(x - 0.5L_x)^2 + 0.25(y - 0.5L_y)^2}$, $H_1 = 20$ m, and $\Delta H = 10$ m. The initial conditions were set to

$$\eta(x, y, 0) = \eta_0 e^{-\frac{(y - 0.5L_y)^2}{3200}}, \quad (60)$$

$$v(x, y, 0) = \sqrt{\frac{g}{H_1}} \eta(x, y, 0), \quad (61)$$

$$u(x, y, 0) = 0. \quad (62)$$

426 where $\eta_0 = 0.25 (H_1 - \Delta H)$, reflecting a one-dimensional (symmetric in x)
 427 wave propagating in the positive y -direction at the linear long wave speed.

428 Snapshots of the evolving η field are shown in Figure 8. By $t = 70$ s
 429 (panel **(b)**), the symmetry in the x -direction has been broken due to refraction
 430 as the portion of the wave front propagating over the deep region in
 431 the center of the domain (near $x = 0.5L_x$) has a faster effective wave speed
 432 than the portion of the wave front that is propagating over the shoal. As
 433 was observed in the 1D simulation shown in Section 3.4, wave propagation
 434 over the shoal has lead to the emergence of solitary waves, most evident
 435 in panels **(b)**, **(d)**, and **(f)** when the main wave front is situated on top of
 436 the shoal. At the later times (panels **(c)**-**(f)**), an interesting interference
 437 pattern of relatively weak waves follows the main wave front due to wave
 438 scattering and solitary wave fissioning induced by the topography.

439 While the decrease in relative amplitude of the portion of the main
 440 wave front over the deep region to that over the shallow region is intuitive,
 441 it should be noted that the absolute wave amplitude decreases over the
 442 deep region as well. This effect occurs due to the bending of wavefronts

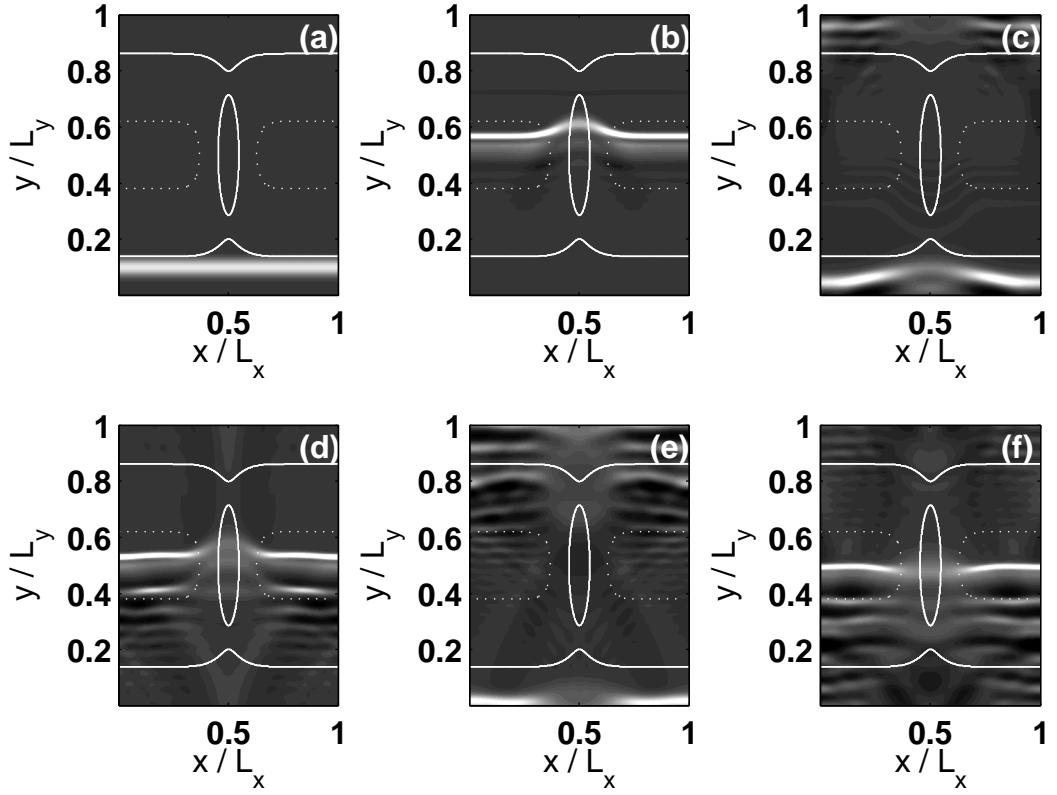


Figure 8: Fixed time snapshots of the free surface displacement at (a) $t = 0$, (b) $t = 70$ s, (c) $t = 140$ s, (d) $t = 210$ s, (e) $t = 280$ s, (f) $t = 350$ s in the *wave propagation over topography* run. The contours are given by 33 equally spaced values between -1.25 m (black) and 2.75 m (white). The dotted and solid white lines correspond to the $H = 12$ m and $H = 18$ m depth contours, respectively, illustrating the shape of the topography.

443 towards the lines of constant depth, which leads to a divergence of energy
 444 over the deep region. Thus, there is a corresponding focusing of energy near
 445 $0.3L_x < x < 0.4L_x$ and $0.6L_x < x < 0.7L_x$ that is more clearly seen in plots
 446 of u , the x -component of velocity, shown at times $t = 70$ s, $t = 210$ s, and
 447 $t = 350$ s in Figure 9. In field situations, this energy focusing could have
 448 implications for wave-boundary-layer interactions. It is also interesting to
 449 note that both the solitary wave widening and fissioning is qualitatively
 450 consistent with KdV theory (Whitham, 1999).

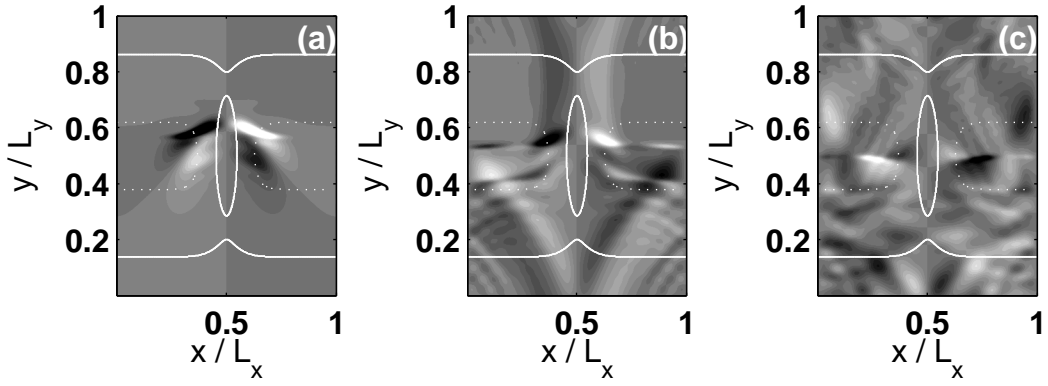


Figure 9: Fixed time snapshots of the u field (x -component of velocity) at (a) $t = 70$ s, (b) $t = 210$ s (c) $t = 350$ s in the *wave propagation over topography* run, corresponding to Figures 8(b),(d),(f), respectively. The contours are given by 21 equally spaced values between -0.25 m s^{-1} (black) and 0.25 m s^{-1} (white). The dotted and solid white lines correspond to the $H = 12$ m and $H = 18$ m depth contours, respectively, illustrating the shape of the topography.

451 4. Conclusions

452 In this manuscript, we have introduced a Fourier pseudospectral method
 453 for solving a dispersive shallow water model of the Boussinesq type in pe-
 454 riodic domains with variable water depth. In line with previous studies
 455 (Eskilsson and Sherwin, 2005; Karniadakis and Sherwin, 2005), we dis-
 456 cussed two approaches for the time-discretization method, the so-called
 457 “coupled” and “scalar” approaches. Although both methods are stable, the
 458 scalar approach reduces the dimension of the resulting linear systems to be
 459 solved by a factor of 2, and transforms the problem of time-stepping mixed

space-time derivatives to a familiar pressure-type elliptic problem. Practical details of implementation were discussed, including details of obtaining efficient solutions to the aforementioned linear systems with numerical linear algebra techniques and pre-conditioning, or discrete Fourier transforms where appropriate. Other practical considerations, such as filter stabilization of aliasing/nonlinearity-driven numerical instabilities were outlined as well. In light of these methods presented, it is clear that FFT-based methods can be extended to problems involving variable bathymetry and can also be a highly-accurate means of solving elliptic problems with variable coefficients if used in conjunction with iterative linear system solvers and pre-conditioning.

Our numerical methodology was validated in one dimension against approximate analytical solutions for the cases of dispersive short waves over a flat-bottom and long waves over a slowly varying bottom. The accuracy of our global Fourier method was also compared to the local nodal DG-FEM method at various orders of accuracy. For a fixed number of degrees of freedom, the Fourier method was shown to have superior resolution and energy-conserving characteristics than the DG-FEM method in all cases considered. Of particular note was that in the low-order DG-FEM simulations ($N < 4$), the short waves are rapidly dissipated by numerical diffusion, yielding a highly inaccurate numerical solution for the physical scenario. These results indicate that the Fourier method is an excellent choice of benchmark for lower-methods (DG-FEM, FVM) that can be used in much more general geometries than the Fourier method. Furthermore, the high accuracy of the Fourier method allows classical water-wave solutions to be explored without the uncertainty associated with the numerical dissipation inherent in low-order methods, thus allowing for a rational set of hypotheses to be constructed for testing against field data.

Grid convergence of the Fourier method was illustrated for the test-case of a long wave steepening and propagating over topography leading to the emergence of solitary waves. This test case was important because it showed that in the well-resolved limit the numerical model is accurate in situations where both dispersion and nonlinearity are prevalent in the dynamics.

A two-dimensional wave dynamical simulation of waves driven by flow over topography was carried out to illustrate how the proposed numerical model may be used in practical GFD problems. A set of rich wave dynamics, including topographically-trapped waves, upstream propagating waves, and waves radiating in the cross-stream direction, was observed. Our results

498 agreed qualitatively with past analytical and numerical results of resonant
 499 wave generation by flow over topography (Grimshaw and Smyth, 1986; Esler
 500 et al., 2007).

501 A second two-dimensional simulation corresponding to a long wave prop-
 502 agating over a shoal was carried out. Interesting linear and nonlinear phe-
 503 nomena such as wave scattering, steepening, and the emergence of fissioning
 504 solitary waves (only in sufficiently shallow regions) rapidly broke the sym-
 505 metry of the initial conditions resulting in a rather complicated final wave
 506 field with a variety of fine scale features.

507 There are many possible improvements and extensions one could make to
 508 the methodology presented here. Improvements include using a higher-order
 509 time discretization method for improved accuracy and using an adaptive fil-
 510 tering procedure to minimize the amount of filtering required for numerical
 511 stability. Another significant improvement would come from using a scal-
 512 able multigrid/domain decomposition approach for the elliptic problem so
 513 that higher resolution simulations can be carried out on parallel computing
 514 clusters. Extensions include replacing the Fourier discretization in one or
 515 both of the spatial directions with a Chebyshev pseudospectral discretiza-
 516 tion Boyd (2001) so that simulations in periodic channels and specialized
 517 closed basins may be carried out. The particular case of an annular circular
 518 basin with application to mid-sized lakes has been explored in Steinmoeller
 519 et al. (2012, in press). Another possible extension would solve a multi-layer
 520 extension of the system (1)-(3) (e.g. de la Fuente et al. (2008); Cotter et al.
 521 (2010)) as a suitable model of internal waves in a density-stratified fluid.
 522 Finally, one may extend the DG-FEM methodology presented in Section 3.2
 523 to the case of two-dimensional arbitrary closed basins using triangulated un-
 524 structured grids to model wave dynamics in real-world lakes with a realistic
 525 representation of the coast-line.

526 **Appendix A. Discontinuous Galerkin Spatial Discretization Method**

In 1D, the augmented system (1)-(3) & (13) reduces to

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} = 0, \quad (\text{A.1})$$

$$\frac{\partial(hu)}{\partial t} + \frac{\partial f(h, u)}{\partial x} = \gamma \frac{\partial z}{\partial x}, \quad (\text{A.2})$$

$$\gamma \frac{\partial^2 z}{\partial x^2} - z = -\frac{\partial a}{\partial x}, \quad (\text{A.3})$$

527 where $f(h, u) = hu^2 + \frac{1}{2}gh^2$, $\gamma = \frac{H^2}{6}$ is a constant, and $a = -\frac{\partial f}{\partial x}$.

Following the developments on nodal discontinuous Galerkin methods in Hesthaven and Warburton (2008), we partition the domain $\Omega = [0, L]$ into K elements $\mathbf{D}^k = [x_l^k, x_r^k]$, $k = 1, \dots, K$. Each element is then discretized on $N+1$ points, using the Legendre-Gauss-Lobatto polynomial interpolation nodes. We proceed by representing the numerical solutions locally on each element in terms of the Lagrange interpolating polynomials, i.e.,

$$h^k(x) = \sum_{i=1}^{N+1} h(x_i^k) \ell_i^k(x), \quad (hu)^k(x) = \sum_{i=1}^{N+1} h(x_i^k) u(x_i^k) \ell_i^k(x), \quad (\text{A.4})$$

with

$$\ell_i^k(x) := \prod_{\substack{0 \leq m \leq k \\ m \neq i}} \frac{x^k - x_m^k}{x_i^k - x_m^k}, \quad (\text{A.5})$$

528 and N is the order of the polynomial interpolants.

To apply the DG-FEM method in strong form, we multiply equations (A.1)-(A.2) on each element k by a member of the space of local test functions $\ell_j^k \in V_h^k = \{\ell_i^k\}_{i=1}^{N+1}$ and integrate the flux terms by parts twice, yielding the semi-discrete equations

$$(\ell_i^k, \ell_j^k)_{\mathbf{D}^k} \frac{dh_j^k}{dt} + (\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} hu_j^k = [\ell_j^k ((hu)^k - (hu)^*)]_{x_l^k}^{x_r^k}, \quad (\text{A.6})$$

$$\begin{aligned} (\ell_i^k, \ell_j^k)_{\mathbf{D}^k} \frac{dhu_j^k}{dt} + (\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} f_j^k &= [\ell_j^k (f^k - f^*)]_{x_l^k}^{x_r^k} \\ &+ \gamma (\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} z_j^k - \gamma [\ell_j^k (z^k - z^*)]_{x_l^k}^{x_r^k}, \end{aligned} \quad (\text{A.7})$$

where we have introduced the local inner product $(u, v)_{\mathbf{D}^k} = \int_{x_l^k}^{x_r^k} uv \, dx$, and it is understood that repeated indices are summed over. To recover an explicit semi-discrete scheme, (A.6)-(A.7) are multiplied by the inverse of the local mass matrix $(\ell_i^k, \ell_j^k)_{\mathbf{D}^k}$ which is typically small $((N+1) \times (N+1))$ and inexpensive to invert. The numerical flux functions f^* and $(hu)^*$ are chosen to be given by the local Lax-Friedrichs flux, e.g.,

$$f^* = \{f\} + \frac{\lambda}{2} [u], \quad (\text{A.8})$$

where

$$\lambda = \max_{u \in [u^-, u^+]} |u| + \sqrt{gH}, \quad (\text{A.9})$$

529 approximates the maximum linearized wave speed. The quantity $\{f\} =$
530 $(f^- + f^+)/2$ represents the average of f 's interior value f^- , on the edge of the
531 element, and its exterior value f^+ , on the edge of the neighboring element,
532 and $\llbracket u \rrbracket = (u^- \hat{n}^- - u^+ \hat{n}^-)$ is the jump in u across the element interface
533 with unit outward-pointing normal \hat{n}^- . In accordance with Eskilsson and
534 Sherwin (2005), z^* was chosen to be given by the central flux, i.e., $z^* = \{z\}$.
535 For a thorough discussion of nodal discontinuous Galerkin methods with a
536 more detailed introduction to the notation used here, we refer the reader to
537 Hesthaven and Warburton (2008).

As explained in Hesthaven and Warburton (2008), in order to solve the
Helmholtz problem (A.3) with DG-FEM, it is necessary to introduce the
auxiliary variable $q = \sqrt{\gamma} \frac{\partial z}{\partial x}$ and rewrite equation (A.3) as the first-order
system

$$\begin{aligned} \sqrt{\gamma}(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} q_j^k - (\ell_i^k, \ell_j^k) z_j^k &= [\ell_j^k (\sqrt{\gamma} q^k - \sqrt{\gamma} q^*)]_{x_l^k}^{x_r^k}, \quad (\text{A.10}) \\ &= -(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} a_j^k + [\ell_j^k (a^k - a^*)]_{x_l^k}^{x_r^k} \end{aligned}$$

$$(\ell_i^k, \ell_j^k)_{\mathbf{D}^k} q_j^k = \sqrt{\gamma}(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} z_j^k - \sqrt{\gamma} [\ell_j^k (z^k - z^*)]_{x_l^k}^{x_r^k}, \quad (\text{A.11})$$

538 where we choose $a^* = \{a\}$, $z^* = \{z\}$, together with the stabilized (or
539 penalized) central flux $q^* = \{q\} - \tau \llbracket z \rrbracket$, $\tau > 0$ for the auxiliary variable, q .
540 The penalty term's purpose is to remove the null eigenmode that would be
541 present if $\tau = 0$, (Hesthaven and Warburton, 2008). Our choice of numerical
542 fluxes for the elliptic problem is essentially a stabilized version of the fluxes
543 used by Bassi and Rebay (1997) for a DG-FEM discretization of the viscous
544 terms in the compressible Navier-Stokes equations. In the results presented
545 in Section 3.2, we use the value $\tau = 1$ for the stabilization parameter. It is
546 known that the convergence rate of the solutions to the discretized elliptic
547 problem is sensitive to the choice of τ , and ideal scalings for τ , dependent
548 on grid-spacing and polynomial order, have been suggested in the literature
549 (Eskilsson and Sherwin, 2005; Hesthaven and Warburton, 2008). However,
550 since the DG-FEM simulations presented in Section 3.2 are well resolved,
551 we do not expect our choice of τ to affect the quality of the solutions.

552 A sparse-matrix representation of the DG-FEM spatial discretization
553 operator represented by (A.10)-(A.11) is then constructed using the tech-
554 niques explained in Hesthaven and Warburton (2008). As in Section 2.3.1,
555 the LU factors of the matrix are computed and stored in the pre-processing
556 stage of the numerical code and re-used at each time-step.

557 The semi-discrete equations are time-stepped using an algorithm that is
 558 analogous to (22)-(27) with the exception that the fourth-order low-storage
 559 explicit Runge-Kutta (LSERK) method (see Hesthaven and Warburton
 560 (2008)) is used in place of the second-order Leapfrog method.

561 Appendix B. Derivation of the WKB solution

To begin, we introduce the slowly-varying spatial coordinate

$$\chi = \epsilon x , \quad (\text{B.1})$$

where ϵ is a small parameter. If we substitute this change of variables into the model equations, retain only terms of order ϵ^2 and lower, one can then find the variable-speed 1D wave equation in terms of η

$$\eta_{tt} - \epsilon^2 (gH\eta_\chi)_\chi = 0 . \quad (\text{B.2})$$

562 It is worth noting that this approximate equation does not contain any
 563 dispersive terms such as those included in the full system (1)-(3), so the
 564 approximation is only expected to be accurate for waves that are sufficiently
 565 long with respect to the water depth.

The solution, $\eta(\chi, t)$, may then be separated into the product of a sinusoidal time-dependent component and an unknown spatial structure, $\psi(\chi)$, as

$$\eta = \psi(\chi)e^{-i\sigma t} , \quad (\text{B.3})$$

where we are considering waves of a single frequency, σ . The spatial structure of the free surface is then assumed to have the form of the WKB ansatz

$$\psi(\chi) = e^{i(\frac{S_0}{\epsilon}(\chi) + S_1(\chi) + \epsilon S_2(\chi) + \dots)} , \quad (\text{B.4})$$

such that

$$\frac{S_0}{\epsilon} \gg S_1 \gg \epsilon S_2 \gg \dots , \quad (\text{B.5})$$

$$\epsilon S_2 \ll 1 , \text{ as } \epsilon \rightarrow 0 . \quad (\text{B.6})$$

Substituting the ansatz (B.4) into the wave equation (B.2) and solving the resulting problems at orders 1 and ϵ yields the WKB solution

$$S_0(\chi) = \pm \int_0^\chi \frac{\sigma}{\sqrt{gH(\zeta)}} d\zeta , \quad (\text{B.7})$$

$$S_1(\chi) = \frac{i}{2} \ln |HS'_0| = \frac{i}{2} \ln \left| \sigma \sqrt{\frac{H}{g}} \right| , \quad (\text{B.8})$$

where prime (') denotes differentiation with respect to χ . Thus, we have

$$\eta(x, y, t) \sim A(\chi) e^{i\left(\frac{S_0}{\epsilon}(\epsilon x) - \sigma t\right)}, \text{ as } \epsilon \rightarrow 0, \quad (\text{B.9})$$

where $A(\chi) = a_0 H^{-\frac{1}{4}}$ and a_0 is an arbitrary constant. Since the problem is linear, it is a straight-forward task to show that

$$u(x, y, t) = \sqrt{\frac{g}{H}} \eta(x, y, t). \quad (\text{B.10})$$

566 Acknowledgement

567 The research is supported by the Natural Sciences and Engineering Research
568 Council of Canada through Discovery Grants for basic research to M.S. and
569 K.G.L.

570 References

- 571 Amezcua, J., Kalnay, E., Williams, P., 2011. The Effects of the RAW Filter on the
572 Climatology and Forecast Skill of the SPEEDY Model. *Mon. Weather Rev.* **139**, 608–
573 619.
- 574 Bassi, F., Rebay, S., 1997. A high-order accurate discontinuous finite element method for
575 the numerical solution of the compressible Navier-Stokes equations. *J. Comp. Phys.*
576 **131**, 267–279.
- 577 Boussinesq, J., 1872. Théorie des ondes et des remous qui se propagent le long d'un
578 canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des
579 vitesses sensiblement pareilles de la surface au fond. *Journal de Mathématique Pures*
580 *et Appliquée, Deuxième Série* **17**, 55–108.
- 581 Boyd, J., 2001. Chebyshev and Fourier Spectral Methods, 2nd Edition. Dover Publica-
582 tions.
- 583 Brandt, P., Rubino, A., Alpers, W., Backhaus, J., 1997. Internal waves in the Strait
584 of Messina Studied by a numerical model and synthetic aperture radar images from
585 *ERS 1/2* Satellites. *J. Phys. Oceanogr.* **27**, 648–663.
- 586 Christov, C., 2000. An energy-consistent dispersive shallow-water model. *Wave Motion*
587 **1018**, 1–14.
- 588 Cotter, C. J., Holm, D. D., Percival, J. R., 2010. The square root depth wave equations.
589 *Proc. R. Soc. A* **466**, 3621–3633.
- 590 de la Fuente, A., Shimizu, K., Imberger, J., Niño, Y., 2008. The evolution of internal
591 waves in a rotating, stratified, circular basin and the influence of weakly nonlinear and
592 nonhydrostatic accelerations. *Limnol. Oceanogr.* **53**(6), 2738–2748.
- 593 Engsig-Karup, A., Hesthaven, J., Bingham, H., Madsen, P., 2006. Nodal DG-FEM solu-
594 tion of high-order Boussinesq-type equations. *J. Eng. Math.* **56**, 351–370.

- 595 Eskilsson, C., Sherwin, S., 2005. Spectral/*hp* discontinuous Galerkin methods for mod-
596 elling 2d Boussinesq equations. *J. Sci. Comp.* **22**, 269–288.
- 597 Esler, J., Rump, O., Johnson, E., 2007. Non-dispersive and weakly dispersive single-layer
598 flow over an axisymmetric obstacle: the equivalent aerofoil formulation. *J. Fluid Mech.*
599 574, 209–237.
- 600 Golub, G., Van Loan, C., 1996. *Matrix Computations*, 3rd Edition. The Johns Hopkins
601 University Press.
- 602 Grimshaw, R., Smyth, N., 1986. Resonant flow of a stratified fluid over topography. *J.*
603 *Fluid Mech.* 169, 429–464.
- 604 Hesthaven, J., Warburton, T., 2008. *Nodal Discontinuous Galerkin Methods*. Springer.
- 605 Iserles, A., 1996. *A First Course in the Numerical Analysis of Differential Equations*.
606 Cambridge University Press.
- 607 Karniadakis, G., Sherwin, S., 2005. *Spectral/*hp* Element Methods for Computational*
608 *Fluid Dynamics*, 2nd Edition. Oxford University Press, USA.
- 609 Leveque, R., 2007. *Finite Difference Methods for Ordinary and Partial Differential Equa-*
610 *tions: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied
611 Mathematics.
- 612 Lynett, P., Liu, P. L.-F., 2004. A two-layer approach to wave modelling. *Proc. R. Soc.*
613 *Lond. A* **460**, 2637–2669.
- 614 Madsen, P., Bingham, H., Liu, H., 2002. A new Boussinesq method for fully nonlinear
615 waves from shallow to deep water. *J. Fluid Mech.* **462**, 1–30.
- 616 Madsen, P., Murray, R., Sørensen, O., 1991. A new form of the Boussinesq equations
617 with improved linear dispersion characteristics. *Coastal Engineering* **15**, 371–388.
- 618 Nwogu, O., 1993. Alternative form of Boussinesq equations for nearshore wave propaga-
619 tion. *J. Waterw. Port Coast. Ocean Eng.* **119**(6), 618–638.
- 620 Peregrine, D., 1967. Long waves on a beach. *J. Fluid. Mech.* **27**(4), 815–827.
- 621 Peyret, R., 2002. *Spectral Methods for Incompressible Viscous Flow*. Springer-Verlag
622 New York, Inc.
- 623 Smith, B., Bjorstad, P., Gropp, W., 2004. *Domain Decomposition: Parallel multilevel*
624 *methods for elliptic partial differential equations*. Cambridge University Press.
- 625 Steinmoeller, D., Stastna, M., Lamb, K., 2012, in press. Pseudospectral methods for
626 Boussinesq-type equations in an annular domain with applications to mid-sized lakes.
627 *J. Comp. Sci.*
- 628 Trefethen, L., 2000. *Spectral Methods in MATLAB*. Society for Industrial and Applied
629 Mathematics.
- 630 Trefethen, L., Bau, D., 1997. *Numerical Linear Algebra*. Society for Industrial and Ap-
631 plied Mathematics.
- 632 Trottenberg, U., Oosterlee, C., Schuller, A., 2000. *Multigrid*, 1st Edition. Academic Press.
- 633 Walkley, M., 1999. A numerical method for extended boussinesq shallow-water wave
634 equations. Ph.D. thesis, University of Leeds, UK.
- 635 Wei, G., Kirby, J., 1995. Time-dependent numerical code for extended Boussinesq equa-
636 tions. *J. Waterw. Port Coast. Ocean Eng.* **121**, 251–261.
- 637 Whitham, G., 1999. *Linear and Nonlinear Waves*. Wiley-Interscience.
- 638 Williams, P., 2011. The RAW Filter: An Improvement of the Robert–Asselin Filter in
639 Semi-Implicit Integrations. *Mon. Weather Rev.* 139, 1996–2007.