# Fourier pseudospectral methods for 2D Boussinesq-type equations

D.T. Steinmoeller[a,*], M. Stastna[a], K.G. Lamb[a]

[a]*200 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)

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

In the following section, we introduce the choice of governing equations and discuss their properties. We then introduce a simple time-stepping procedure followed by a more efficient technique inspired by the approach of Eskilsson and Sherwin (2005) that reduces the size of the resulting linear system by a factor of 2 by transforming the dispersive terms to a standard pressure-type elliptic problem. A Fourier pseudospectral spatial discretization method is introduced for numerical solutions in two spatial dimensions along with strategies for solving the required linear systems. A nodal DG-FEM spatial discretization method in one dimension is also introduced. The paper concludes with validation of numerical solutions and a comparison between Fourier and DG-FEM solutions to the Boussinesq-type system in one dimension, followed by sample results obtained in two dimensions with the Fourier method. The present work suggests that scalable Fast Fourier Transform (FFT) based methods for water wave equations can be extended to physical cases involving non-constant bathymetry and can also be an effective tool for solving elliptic problems with non-constant coefficients provided they are used alongside an appropriate iterative linear solver with pre-conditioning. Given the highly accurate nature of the Fourier method, the results presented here may be seen as a benchmark for lower-order spatial discretization techniques such as DG-FEM and FVM, and allow for rational hypotheses to be formulated for subsequent testing against field data of water waves.

## 2. Methods

### *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

$$\begin{aligned}
\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) &= 0\,, &(1)\\
\frac{\partial (uh)}{\partial t} + \nabla \cdot ((uh)\mathbf{u}) &= -gh\frac{\partial \eta}{\partial x} + fvh + \frac{H^2}{6}\frac{\partial}{\partial x}\left(\nabla \cdot \frac{\partial (\mathbf{u}h)}{\partial t}\right)\,, &(2)\\
\frac{\partial (vh)}{\partial t} + \nabla \cdot ((vh)\mathbf{u}) &= -gh\frac{\partial \eta}{\partial y} - fuh + \frac{H^2}{6}\frac{\partial}{\partial y}\left(\nabla \cdot \frac{\partial (\mathbf{u}h)}{\partial t}\right)\,, &(3)
\end{aligned}$$

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

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

### *2.2. Time-Stepping Techniques*

For the moment, we will assume that we have spatially discretized the system (1)–(3) using a method of lines approach as discussed by Trefethen (2000). That is, the flow variables of interest $(h, u, v)$ have been discretized on $N$ grid-points and are now represented by the $N \times 1$ vectors $(\mathbf{h}, \mathbf{u}, \mathbf{v}) = ([h_1, \cdots, h_N]^\mathsf{T}, [u_1, \cdots, u_N]^\mathsf{T}, [v_1, \cdots, v_N]^\mathsf{T})$, where we adopt the notation that bold-faced variables refer to the discretized approximate solution fields of the system (1)-(3). We further assume that the continuous 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 the $N \times N$ matrices $D_x$, $D_y$, $D_{xx}$, $D_{yy}$, $D_{xy}$ or that the required matrix-vector products are attainable by other means, such as the pseudospectral technique Peyret (2002).

To keep the discussion as general as possible, we do not specify which spatial discretization scheme we are using since the following time-stepping schemes may be applied to a number of spatial discretization methods including Finite Difference methods, the Fourier pseudospectral method, the 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

$$\begin{aligned} \frac{d\mathbf{h}}{dt} &= -D_x(\mathbf{uh}) - D_y(\mathbf{vh}) \,, && (4) \\ \frac{d(\mathbf{uh})}{dt} - \frac{\mathbf{H}^2}{6}\frac{d}{dt}\left(D_{xx}(\mathbf{uh}) + D_{xy}(\mathbf{vh})\right) &= -D_x(\mathbf{uuh}) - D_y(\mathbf{uvh}) \\ &- g\mathbf{h}D_x\boldsymbol{\eta} + f\mathbf{vh} \,, && (5) \\ \frac{d(\mathbf{vh})}{dt} - \frac{\mathbf{H}^2}{6}\frac{d}{dt}\left(D_{xy}(\mathbf{uh}) + D_{yy}(\mathbf{vh})\right) &= -D_x(\mathbf{vuh}) - D_y(\mathbf{vvh}) \\ &- g\mathbf{h}D_y\boldsymbol{\eta} - f\mathbf{uh} \,, && (6) \end{aligned}$$

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.,

$$\mathbf{ab} = [a_1 b_1, \cdots, a_N b_N]^\mathsf{T} \,.$$

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). We discretize the flow variables $(\mathbf{h}, \mathbf{uh}, \mathbf{vh})$ at the time levels

$$t_n = n\Delta t, \quad n = 0, 1, \cdots \,, \tag{7}$$

where $\Delta t$ represents the time-step and adopt the notation that superscript $n$ denotes the $n^{th}$ time-step. Applying the Leapfrog formula to equations

(4)-(6) results in the scheme

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

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

where

$$\begin{aligned} \mathbf{RHS}_1^{n,n-1} &= (\mathbf{uh})^{n-1} - \frac{\mathbf{H}^2}{6} D_{xx}(\mathbf{uh})^{n-1} - \frac{\mathbf{H}^2}{6} D_{xy}(\mathbf{vh})^{n-1} && (10) \\ &+ 2\Delta t(-D_x(\mathbf{uuh})^n - D_y(\mathbf{uvh})^n - g\mathbf{h}^n D_x \boldsymbol{\eta}^n + f(\mathbf{vh})^n) \ , \\ \mathbf{RHS}_2^{n,n-1} &= (\mathbf{vh})^{n-1} - \frac{\mathbf{H}^2}{6} D_{xy}(\mathbf{uh})^{n-1} - \frac{\mathbf{H}^2}{6} D_{yy}(\mathbf{vh})^{n-1} && (11) \\ &+ 2\Delta t(-D_x(\mathbf{vuh})^n - D_y(\mathbf{vvh})^n - g\mathbf{h}^n D_y \boldsymbol{\eta}^n - f(\mathbf{uh})^n) \ , \end{aligned}$$

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

### *2.2.1. The Scalar Approach*

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

The approach begins by introducing the scalar variable

$$z = \nabla \cdot (\mathbf{u}h)_t \ , \tag{12}$$

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

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

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.,

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

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

$$\begin{aligned} \frac{d\mathbf{h}}{dt} &= -D_x(\mathbf{uh}) - D_y(\mathbf{vh}) , & (15)\\ \frac{d(\mathbf{uh})}{dt} &= -D_x(\mathbf{uuh}) - D_y(\mathbf{uvh}) - g\mathbf{h}D_x\boldsymbol{\eta} + f\mathbf{vh} + \frac{\mathbf{H}^2}{6}D_x\mathbf{z} , & (16)\\ \frac{d(\mathbf{vh})}{dt} &= -D_y(\mathbf{vuh}) - D_y(\mathbf{vvh}) - g\mathbf{h}D_y\boldsymbol{\eta} - f\mathbf{uh} + \frac{\mathbf{H}^2}{6}D_y\mathbf{z} , & (17)\\ \frac{\mathbf{H}^2}{6}(D_{xx}\mathbf{z} &+ D_{yy}\mathbf{z}) - \mathbf{z} & \\ &+ \frac{1}{6}\left(D_x(\mathbf{H}^2)D_x\mathbf{z} + D_y(\mathbf{H}^2)D_y\mathbf{z}\right) = -(D_x\mathbf{a_1} + D_y\mathbf{a_2}), & (18) \end{aligned}$$

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

$$\mathcal{A}\mathbf{z} = \mathbf{b} , \tag{19}$$

with

$$\begin{aligned} \mathcal{A} &= \frac{\mathcal{H}^2}{6}(D_{xx} + D_{yy}) - I + \frac{1}{6}\left(D_x(\mathcal{H}^2)D_x + D_y(\mathcal{H}^2)D_y\right) , & (20)\\ \mathbf{b} &= -(D_x\mathbf{a_1} + D_y\mathbf{a_2}) . & (21) \end{aligned}$$

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 available. The resulting scheme at each time-step is

$$
\begin{aligned}
\mathbf{h}^{n+1} &= \mathbf{h}^{n-1} + 2\Delta t(-D_x(\mathbf{uh})^n - D_y(\mathbf{vh})^n) \ , && (22)\\
(\mathbf{uh})^{\dagger} &= (\mathbf{uh})^{n-1} + 2\Delta t \mathbf{a_1}^n \ , && (23)\\
(\mathbf{vh})^{\dagger} &= (\mathbf{vh})^{n-1} + 2\Delta t \mathbf{a_2}^n \ , && (24)\\
\mathbf{z}^{\dagger} &= \mathcal{A}^{-1}\mathbf{b}^{\dagger} \ , && (25)\\
(\mathbf{uh})^{n+1} &= (\mathbf{uh})^{\dagger} + 2\Delta t \frac{\mathbf{H}^2}{6} D_x \mathbf{z}^{\dagger} \ , && (26)\\
(\mathbf{vh})^{n+1} &= (\mathbf{vh})^{\dagger} + 2\Delta t \frac{\mathbf{H}^2}{6} D_y \mathbf{z}^{\dagger} \ , && (27)
\end{aligned}
$$

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 in accuracy 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

$$
\begin{aligned}
x_i &= i\Delta x, \quad i = 0, \cdots, N_x - 1 \ , && (28)\\
y_j &= j\Delta y, \quad j = 0, \cdots, N_y - 1 \ , && (29)
\end{aligned}
$$

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

$$k_i = \frac{2\pi}{L_x} i, \quad i = 0, \cdots, N_x - 1 , \tag{30}$$

$$l_j = \frac{2\pi}{L_y} j, \quad j = 0, \cdots, N_y - 1 . \tag{31}$$

Rather than using differentiation matrices to compute the approximate derivatives in the schemes presented above, we employ the "pseudospectral technique" as described by Peyret (2002). That is, differentiation is performed in spectral space (the space of the Fourier coefficients) with the fast discrete Fourier transform (FFT) while products are performed in physical space. Doing so allows one to avoid the expense of directly computing convolution sums in the space of the Fourier coefficients, as the nonlinear terms would require. Pseudospectral differentiation is also faster than explicitly calculating matrix-vector products that require $O(N^2)$ floating-point operations (FLOPS) since the FFT requires $O(N \log N)$ FLOPS, and 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 $\boldsymbol{\phi}$ which may represent a flow variable or a product of flow variables, we approximate its discrete derivatives as

$$\begin{aligned}
\boldsymbol{\phi}_x &= \mathcal{F}_x^{-1}\left(\mathrm{i}\mathcal{K}\mathcal{F}_x(\boldsymbol{\phi})\right) , &(32)\\
\boldsymbol{\phi}_y &= \mathcal{F}_y^{-1}\left(\mathrm{i}\mathcal{L}\mathcal{F}_y(\boldsymbol{\phi})\right) , &(33)\\
\boldsymbol{\phi}_{xx} &= \mathcal{F}_x^{-1}\left(-\mathcal{K}^2\mathcal{F}_x(\boldsymbol{\phi})\right) , &(34)\\
\boldsymbol{\phi}_{yy} &= \mathcal{F}_y^{-1}\left(-\mathcal{L}^2\mathcal{F}_y(\boldsymbol{\phi})\right) , &(35)\\
\boldsymbol{\phi}_{xy} &= \mathcal{F}_y^{-1}\left(\mathrm{i}\mathcal{L}\mathcal{F}_y(\boldsymbol{\phi}_x)\right) , &(36)
\end{aligned}$$

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

The underlying assumptions used in this spatial discretization are that the solution fields are smooth and periodic in space in both directions, and they are hence well represented by a sinusoidal basis. Given these assumptions, the Fourier pseudospectral spatial discretization method guarantees an exponential convergence rate (Boyd, 2001). If one or both of these assumptions are broken, Gibbs oscillations are introduced into the solution and the convergence rate is reduced to polynomial order.

### *2.3.1. Solving the Linear System*

In order to solve the linear system (19), one may be tempted to explicitly build the large matrix $\mathcal{A}$ using two-dimensional spectral differentiation matrices. However, this is typically not a good idea due to memory restrictions. Two-dimensional spectral differentiation matrices can be built from kronecker products between the 1D differentiation matrix and the appropriate identity matrix, and require $O(N_xN_y(N_x+N_y))$ memory. If mixed spatial derivatives are required, the situation can be the worst case, $O(N_x^2N_y^2)$ memory, which is certainly not reasonable. It is clear that indirect methods for solving the system (19) are required in the case of a pseudospectral 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

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

where

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

$\mathbf{1}$ 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

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

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

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

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

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

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

### *2.3.2. Finite Differences Pre-Conditioner*

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

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

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

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

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

to obtain a more well-conditioned linear system since $(\mathcal{A}^{(2)})^{-1}\mathcal{A} \approx I$. Of course, this is merely illustrative since $\mathcal{A}$ is not explicitly built and computing the explicit inverse of $\mathcal{A}^{(2)}$ is impractical. Instead, the fact that we are using the GMRES method requires that linear systems of the form $\mathcal{A}^{(2)}\tilde{\mathbf{z}} = \tilde{\mathbf{b}}$ be solved at each iteration. In order to ensure linear systems of this form can be solved effectively, it is useful to compute the LU-factorization of $\mathcal{A}^{(2)}$ in the pre-processing stage and to simply reuse its factors at each GMRES iteration.

It has been found that using the factors returned by the sparse-LU factorization routine provided in the UMFPACK library yield very fast 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

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

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

### *2.4. Filter Stabilization of Aliasing-driving Instabilities*

The equations do not possess any viscosity terms and thus lack any physical energy dissipation mechanism. As a result, the quadratic nonlinearity terms can cause energy to accumulate at the small scales in an unphysical manner. Additionally, aliasing errors that occur due to the “pointwise product” treatment of the nonlinear terms can drive weak numerical instabilities 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

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

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

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

## 3. Results and Discussion

### *3.1. Pre-conditioner performance test*

To confirm the performance of our pre-conditioned GMRES algorithm for solving equation (18), we have compared it to the GMRES method without pre-conditioning. The problem we consider for inversion here corresponds to the first time-step of the full simulation presented below in Section 3.7. Hence, the variable coefficients on the left-hand side as well as the function on the right-hand side of (18) are both non-trivial. The convergence criterion was taken to require the magnitude of the relative residual to be below $10^{-9}$, and this value was also used for all simulations with a non-flat bottom, i.e., whenever GMRES was used. The problem was solved at a variety of grid resolutions, ranging from $16 \times 16$ to $1024 \times 1024$.

Iteration counts for both the 'GMRES with pre-conditioning' (GMRESP) and 'GMRES without pre-conditioning' (GMRESNP) methods along with the ratio of their run-times are listed at all resolutions considered in Table 1. The corresponding tests were carried out in Matlab, and it was found that the values for the run-times fluctuated somewhat due to the performance of Matlab's built-in parallelization's dependendence on processor load. Hence, the run-time values used in Table 1 were averaged over ten runs to average out these fluctuations.

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

|  | GMRESP | GMRESNP |  |
|---|---|---|---|
| Resolution | Iteration Count | Iteration Count | Run-Time Ratio |
| $16 \times 16$ | 5 | 5 | 0.71 |
| $32 \times 32$ | 6 | 7 | 0.77 |
| $64 \times 64$ | 9 | 10 | 0.78 |
| $128 \times 128$ | 9 | 15 | 0.94 |
| $256 \times 256$ | 12 | 32 | 1.75 |
| $512 \times 512$ | 16 | 70 | 4.27 |
| $1024 \times 1024$ | 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.

### 280 *3.2. Convergence test of the Fourier method*

To verify that the Fourier spatial discretization method is giving the desired exponential convergence rate, we have performed a convergence study for the Helmholtz problem (13) in two dimensions, where we have chosen the form of the exact solution and variable coefficient *a priori*, and the right-hand side function was calculated analytically from the known functions. The problem we consider here is

$$\nabla \cdot (\alpha \nabla z) - z = f \ , \tag{44}$$

on the periodic square $\Omega = [0, 2\pi] \times [0, 2\pi]$, where

$$\begin{aligned} \alpha(x,y) &= 2 + \sin(x)\cos(y) \ , && (45) \\ f(x,y) &= -\mathrm{sech}(10(y-\pi))^3[9\cos(y)\cos(x)\sin(9x)\cosh(10(y-\pi))^2 \\ &- 10\sin(y)\sin(x)\cos(9x)\sinh(10(y-\pi))\cosh(10(y-\pi)) \\ &- 19\cos(9x)\cos(y)\sin(x)\cosh(10(y-\pi))^2 \\ &- 37\cos(9x)\cosh(10(y-\pi))^2 \\ &+ 200\cos(9x)\cos(y)\sin(x) + 400\cos(9x)] \ , && (46) \end{aligned}$$

so that the exact solution is given by

$$z(x,y) = \cos(9x)\,\mathrm{sech}(10(y-\pi)) \ . \tag{47}$$

The problem was first solved on a coarse $16 \times 16$ grid. The resolution was then doubled in each direction and the problem was solved repeatedly until

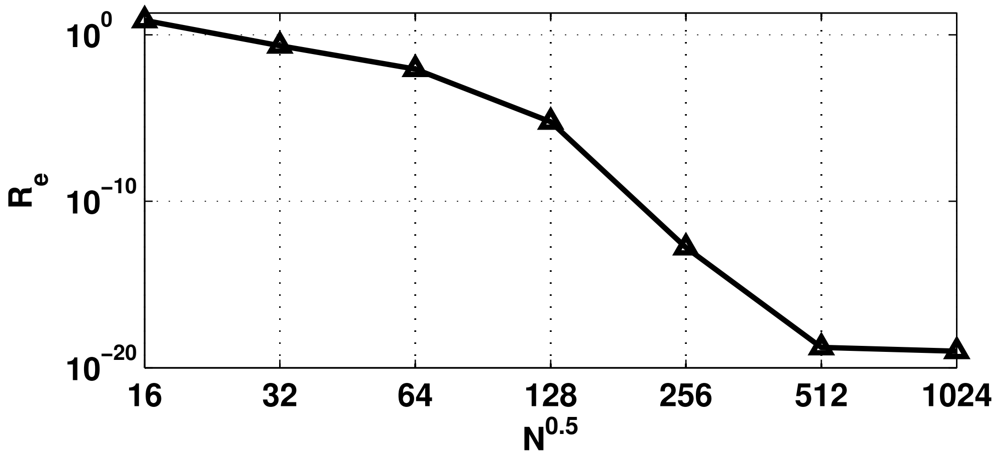


Figure 1: Relative error ($R_e$) between exact and numerical solutions to the 2D Helmholtz problem (44) vs. $N^{0.5}$, the square-root of the total number of grid points.

convergence to numerical precision was reached. In each case, the number of grid points was kept the same in each direction, i.e., $N_x = N_y = N^{0.5}$, where $N^{0.5}$ is the square root of the total number of grid points. In Figure 1, we plot the relative $L^2$ error, $R_e$, vs. $N^{0.5}$, where

$$R_e = \frac{\iint\limits_{\Omega} \left(z_{Num} - z\right)^2 dS}{\iint\limits_{\Omega} z^2 dS} \ , \tag{48}$$

and $z_{Num}$ is the solution computed numerically via the pre-conditioned GMRES method. The integral in the numerator of equation (48) was computed numerically using the Fourier expansion coefficients (obtained with FFT), and the integral in the denominator was calculated exactly.

Figure 1 reveals that the Fourier method converges exponentially fast to the exact solution, as expected, and that convergence to within numerical precision has been reached at a grid resolution of 512 × 512.

### *3.3. 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 $g = 9.81$ m s$^{-2}$. The initial condition is

$$\begin{aligned} \eta(x,0) &= \eta_0 \cos(0.15x)\cos(0.05x)\mathrm{e}^{-5\left(\frac{x-0.5L_x}{400}\right)^2} , && (49) \\ u(x,0) &= 0 , && (50) \end{aligned}$$

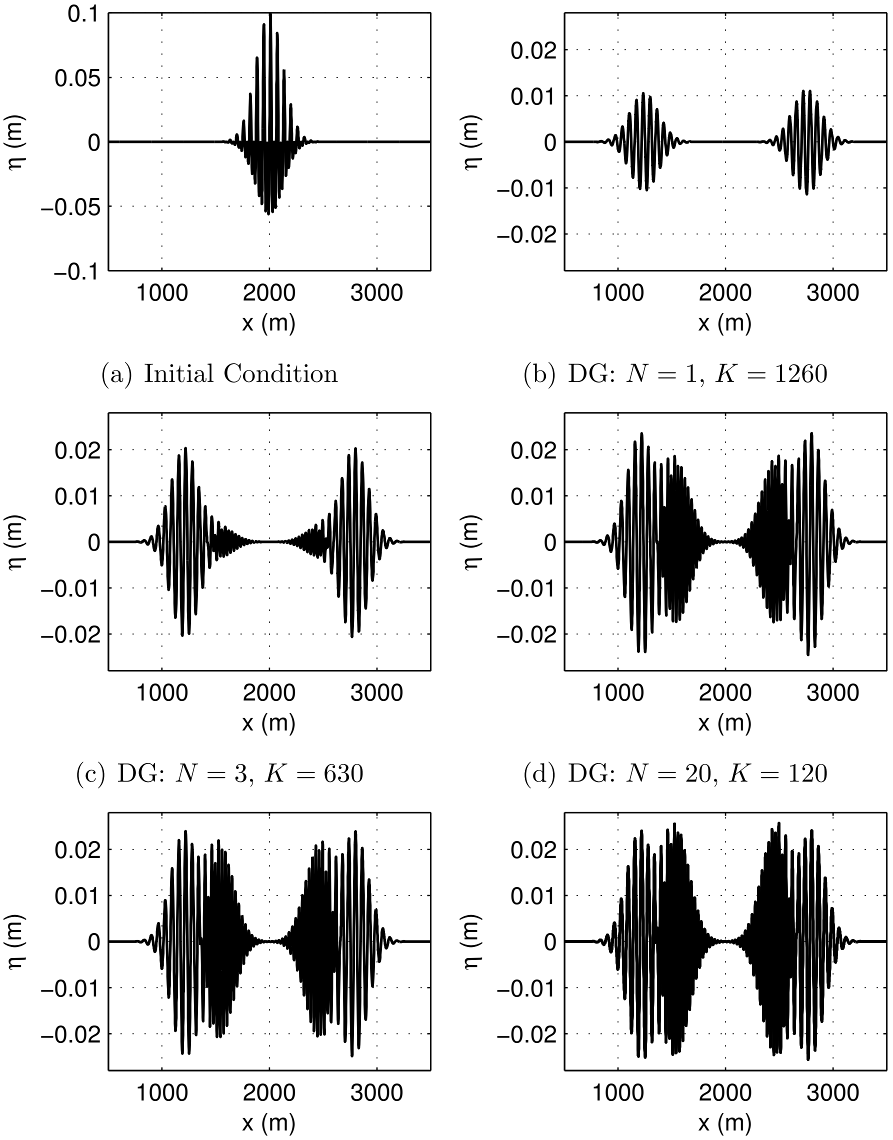
where $\eta_0 = 0.1$ m. The amplitude of the wave packet was chosen to be small enough so that linear wave theory would be a good predictor of the group velocities. This was confirmed by solving the linearized equations exactly in Fourier space and comparing with the numerical solution (not shown). Due to dispersion, we expect the longer waves to overtake and lead the shorter waves, after sufficient time has passed, since the linear group velocity of the longer waves is $c_g \approx 9.31$ m s$^{-1}$ while the group velocity of the shorter waves is $c_g \approx 6.14$ m s$^{-1}$. This run was also used to validate the numerical methods in the regime where nonlinear effects are negligible and the bottom is flat.

In Figure 2, the results of the runs are displayed at various orders of accuracy. The values of $K$ (total number of elements) and $N$ (order of the basis functions) were chosen such that the total number of points used in the DG-FEM method would be fixed at $N_{dof} = K(N+1) = 2520$. Modal filtering was not used in any of the runs, since the choices of small-amplitude waves and a flat bottom remove most, if not all, of the sources of nonlinearity and aliasing errors. A striking observation is that for the low-order runs, the shorter waves are dissipated to a very large degree, and in the $N = 1$ case, in effect entirely.

In Figure 3, the time series of the domain-integrated total energy, defined by

$$E(t) = \int_0^{L_x} \frac{1}{2}hu^2 + \frac{1}{2}g\eta^2 \, dx \, , \qquad (51)$$

is plotted for each of the cases shown in Figure 2. It can be shown that the Boussinesq system (1)-(3) does not conserve energy (as is true of most



(a) Initial Condition

(b) DG: $N = 1$, $K = 1260$

(c) DG: $N = 3$, $K = 630$

(d) DG: $N = 20$, $K = 120$

(e) DG: $N = 41$, $K = 60$

(f) Fourier: $N_x = 2520$

Figure 2: 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) $\eta$ 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.

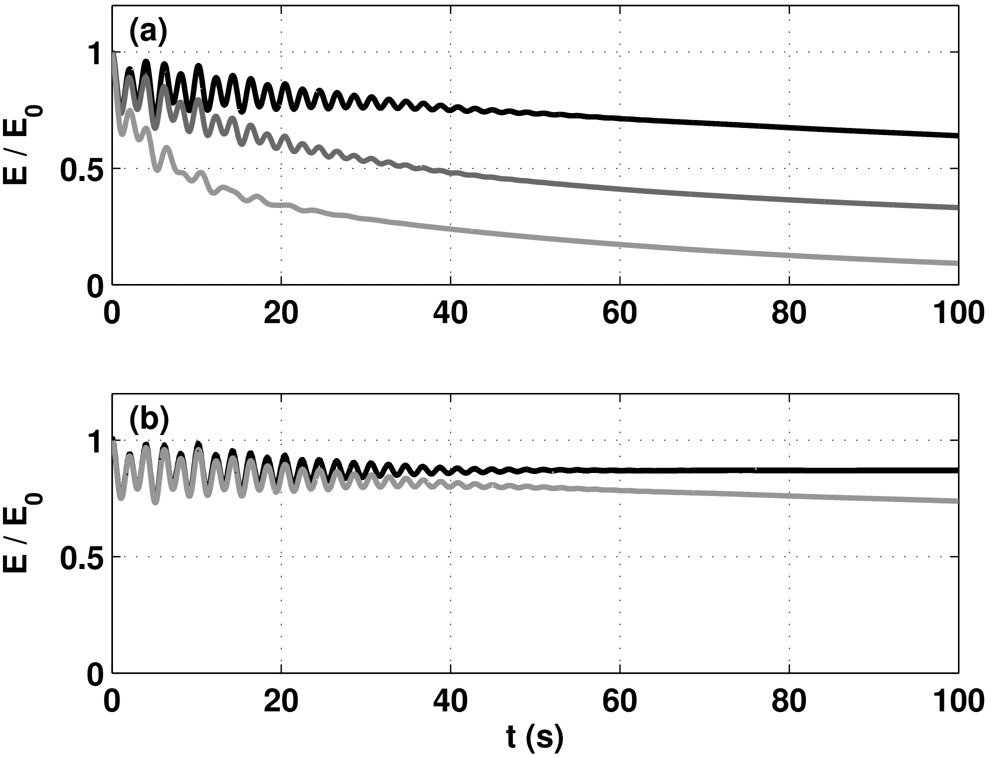


Figure 3: Domain-integrated total energy time series for the *1D dispersive short-waves* run with (a) the DG-FEM method at orders $N = 1$ (light grey), $N = 3$ (dark grey), and $N = 20$ (black), and (b) the DG-FEM method at order $N = 41$ (grey) and the Fourier method with $N_x = 2520$ points (black). 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.

Boussinesq-type systems), and even exact solutions would not satisfy the physical property $E(t)$ =constant. Indeed, the plots in Figure 3 reveal that in the absence of numerical dissipation, $E(t)$ is oscillatory. Once again, this fact has been confirmed by comparing to the exact solution of the linearized equations in Fourier space (and invoking Parseval's theorem). This oscillatory behaviour is a well-known consequence of using Boussinesq-type systems, and Boussinesq-type systems that conserve energy exactly have been proposed (Christov, 2000). However, these energy-conserving systems tend to be undesirable for numerical integration due to the presence of third-order spatial derivatives.

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

### *3.4. 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 $\eta$. 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

$$\begin{aligned} \eta(x,y,t) &\sim A(\epsilon x)\mathrm{e}^{i\left(\frac{S_0}{\epsilon}(\epsilon x)-\sigma t\right)}\ , \text{ as } \epsilon \to 0\ , && (52) \\ u(x,y,t) &= \sqrt{\frac{g}{H}}\eta(x,y,t)\ . && (53) \end{aligned}$$

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

To compare our numerical code with the WKB solution (52), 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

$$H(\epsilon x) = H_0 - \Delta H \sin(\epsilon x)\ , \tag{54}$$

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 $\Delta H$ from 0–2.5 m, expecting the two solutions to agree best in the limit that $\Delta H \to 0$ (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

$$\Delta t = \frac{1}{20}\frac{\Delta x}{\sqrt{gH_0}}\ , \tag{55}$$

where $\Delta x$ 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

$$\sigma = \sqrt{gH_0}\left(\frac{10\pi}{L_x}\right)\ , \tag{56}$$

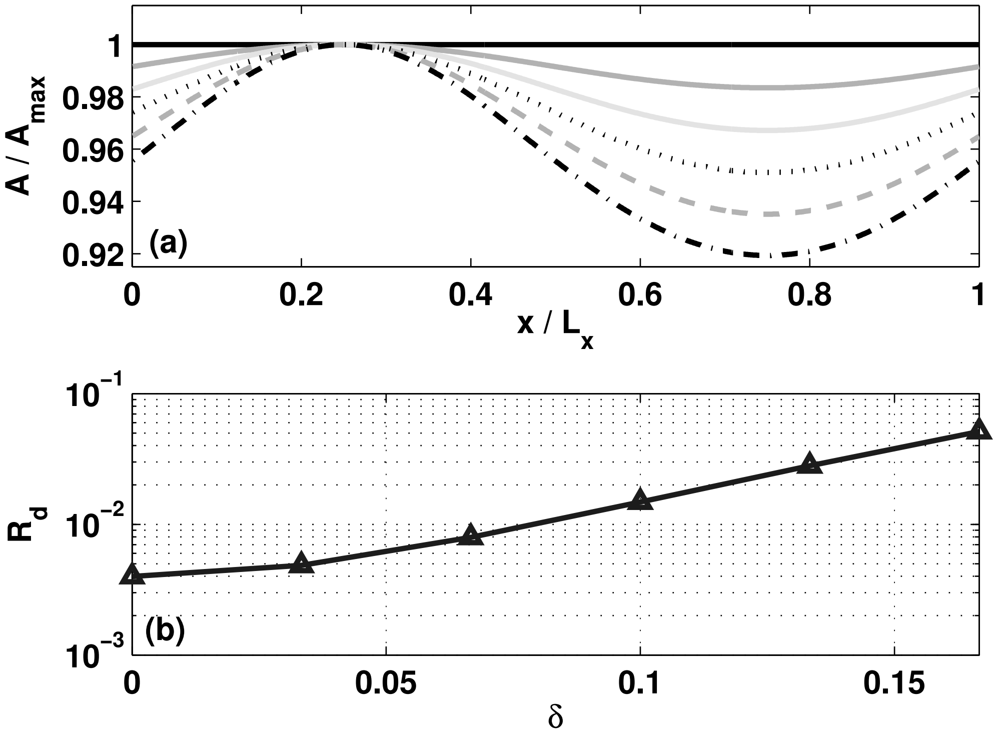


Figure 4: Panel **(a)**: Envelopes of the WKB solution, scaled by their maximum value, for the values of $\delta = \Delta H/H_0 = 0$ (solid, black), 1/30 (solid, dark grey), 1/15 (solid, light grey), 1/10 (dotted), 2/15 (dashed), 1/6 (dash-dotted). Panel **(b)**: Relative difference ($R_d$) between the numerical solution and the WKB solution after five wave periods vs. $\delta$.

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

$$R_d = \frac{\int_0^{L_x} \left(\eta_{Num} - \eta_{WKB}\right)^2 dx}{\int_0^{L_x} (\eta_{WKB})^2 dx} \; , \tag{57}$$

was calculated where $\eta_{Num}$ and $\eta_{WKB}$ represent the numerical and WKB $\eta$ 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 $\Delta H \to 0$ is illustrated in Figure 4 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 $\delta$, and panel **(b)** shows the decline in the relative difference between the analytical and numerical solution as $\delta \to 0$. 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.

### *3.5. Grid-convergence study using a simulation of 1D wave-topography interaction*

We next focus our attention on a 1D simulation of nonlinear and dispersive waves repeatedly propagating over a ridge with the Fourier method. Since analytical solutions are not available to confirm the validity of the results, we rely on grid-doubling experiments to illustrate the method's 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

$$H(x,y) = H_1 - \Delta H \mathrm{e}^{-5\left(\frac{x-0.5L_x}{100}\right)^4} \; , \tag{58}$$

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

$$\begin{aligned} \eta(x,0) &= \eta_0 \mathrm{e}^{-\left(\frac{x-0.25L_x}{100}\right)^2} \; , & (59) \\ u(x,0) &= \sqrt{\frac{g}{H_1}}\eta(x,0) \; , & (60) \end{aligned}$$

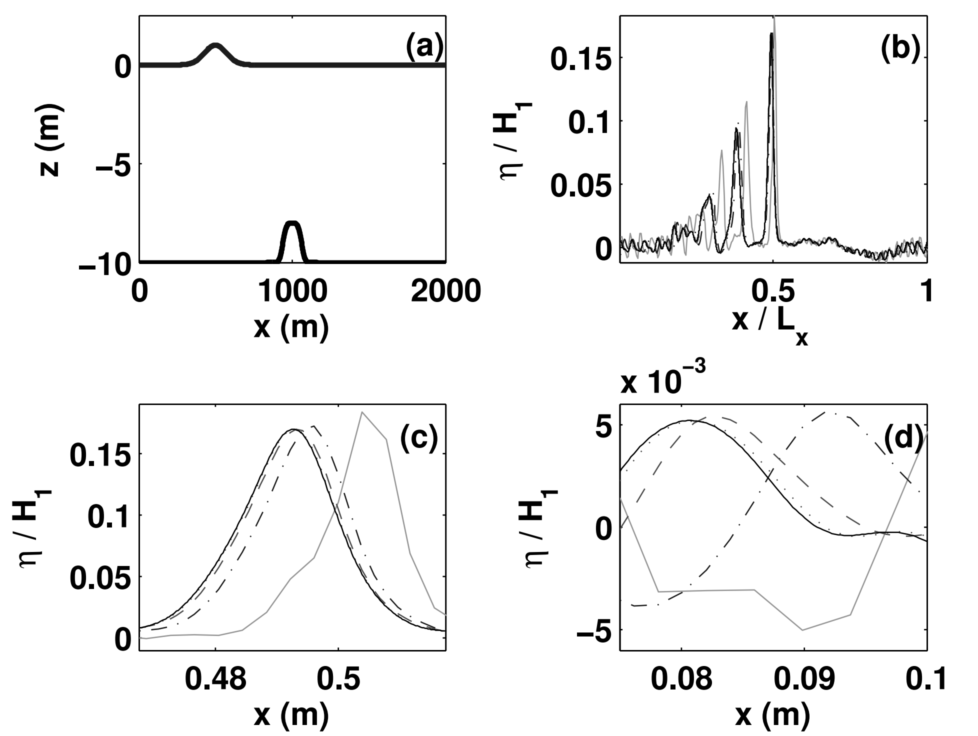


Figure 5: Results for the *1D wave-topography interaction* run. Panel **(a)**: Plot of the initialization, showing the topography $z = -H(x)$ and the initial free surface displacement $z = \eta(x, 0)$. Panels **(b)**–**(d)**: $\eta$ at $t = T_{final}$ at resolutions $N_x = 256$ (solid, light grey), $N_x = 512$ (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 $\eta$ has been made dimensionless by dividing by the off-ridge water depth, $H_1 = 10$ m.

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 5**(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

$$\Delta t = \frac{1}{2}\frac{\Delta x}{\sqrt{gH_1}} , \tag{61}$$

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

In this simulation, nonlinearity plays a key role in the evolution of the flow. The initial wave immediately begins to steepen, and the steepening is further enhanced due to shoaling as the wave propagates over the ridge (not shown). Dispersion then acts to balance the nonlinearity and prevent the formation of shocks. The final result is a collection of three solitary waves propagating in the positive $x$-direction followed by a dispersive wavetrain. These solitary waves are similar in shape to the $\text{sech}^2(\cdot)$ solitons predicted by Korteweg-de Vries (KdV) theory (Whitham, 1999). It can be shown that 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 $\eta$ field at $t = T_{final}$ at several resolutions are depicted in Figure 5**(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 $\eta$ field (see Figure 5**(d)**).

In light of the strong results on exponential convergence reported in Section 3.2, the question arises as to whether the same properties can be expected for time-dependent problems, such as the one presented in this section. Unfortunately, only an algebraic rate of convergence can be expected due to the fact that the Leapfrog time-stepper is only second-order accurate. This fact has been verified by computing the relative $L^2$ difference ($R_d$) between each of the solutions shown in Figure 5 and the solution computed with $N_x = 8192$ grid points (not shown). It was found that $R_d$ satisfied the algebraic relation $R_d = O(N_x^{-3})$, and thus the convergence rate is not exponential.

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

The physical parameters were set to: $g = 9.81$ m s$^{-2}$, $f = 0$ (no rotation), and $L_x = L_y = 2$ 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

$$H(x, y) = H_1 - \Delta H \mathrm{e}^{-5(\frac{x-0.5L_x}{100})^4 - 5(\frac{y-0.5L_y}{200})^4}\ , \tag{62}$$

with $H_1 = 20$ m and $\Delta H = 2$ m. This is essentially a two-dimensional version of the depth-profile used in Section 3.5, 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

$$F_x = \begin{cases} \frac{\sqrt{gH_1}}{\beta}, & 0 \leq t < 10 \text{ s} \\ 0, & t \geq 10 \text{ s} \end{cases} \tag{63}$$

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

Snapshots of the developing $\eta$-field are shown in Figure 6. In addition to a trapped wave of depression generated over the ridge, an upstream-propagating wavefront of elevation (with a slightly depressed tail, or possibly a second wave) can be seen emanating from the ridge, and travelling westward. This wavefront can be seen losing amplitude as time progresses.

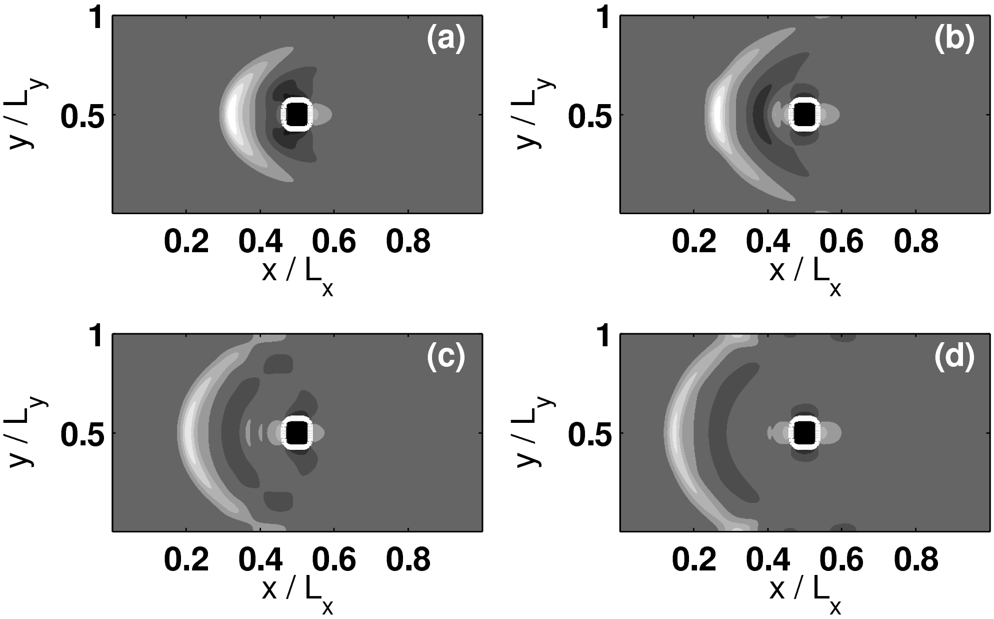


Figure 6: Fixed time snapshots of the free surface displacement at **(a)** $t$ = 60 s, **(b)** $t$ = 80 s, **(c)** $t$ = 100 s, and **(d)** $t$ = 120 s in the *2D wave generation by flow over topography* run. The contours shown have values of $\eta = \pm 0.1$ m, $\pm 0.2$ m, $\pm 0.3$ m, $\pm 0.4$ m, $\pm 0.5$ m, $\pm 0.6$ m, $\pm 1$ 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.

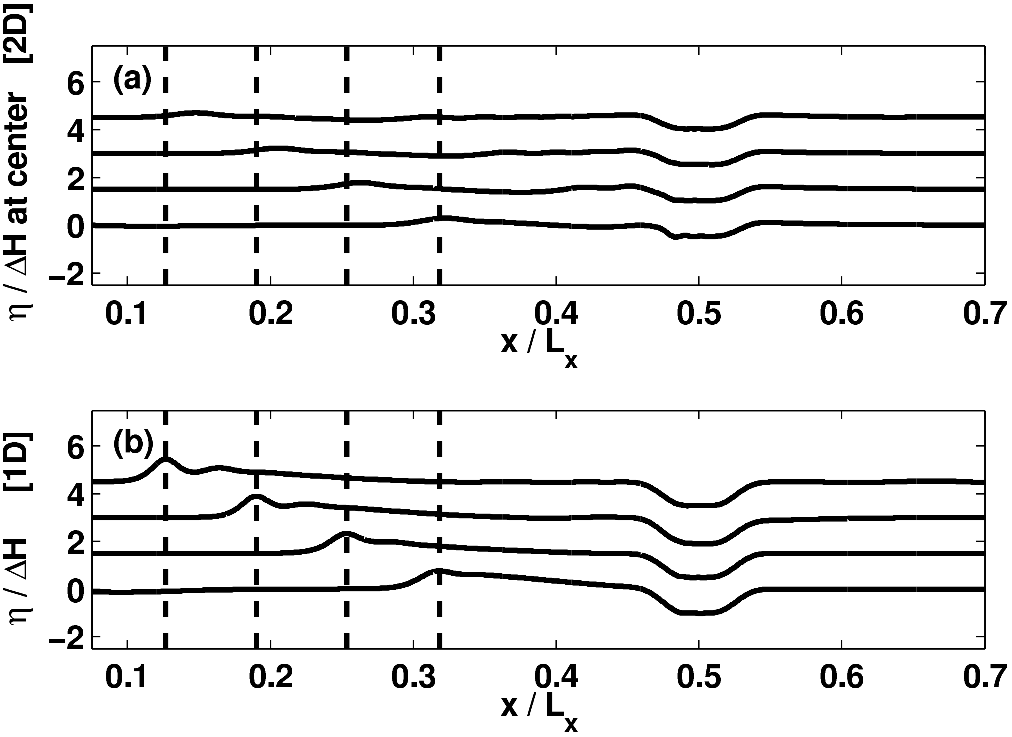


Figure 7: Panel (**a**): 1D slices of the snapshots presented in Figure 6 through the line $y = 1$ km. Panel (**b**): Snapshots of the $\eta$ field for an analogous 1D simulation, where variations in $y$ have been neglected. In each panel, a single curve corresponds to a time in Figure 6, with the lowest 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 $\eta$ has been made dimensionless by dividing by the ridge height, $\Delta H = 2$ m.

This is due to to radial spreading, or in other words geometric decay. The extent of this decrease in amplitude (energy density) due to geometric decay can be illustrated qualitatively by comparing this 2D simulation to an analogous 1D simulation where variations in the $y$-direction are neglected. This comparison is carried out in Figure 7, where it can be seen that in the 1D case (panel **(b)**), the upstream-propagating wave front better retains its amplitude than in the 2D case (panel **(a)**) since there is no radiation in the $y$-direction.

### *3.7. 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 3.6), as were the filtering parameters. The grid was taken to have 1024 points in each direction. The depth-profile is

$$H = H_1 - \Delta H \left[ \text{sech}\left( \frac{y - 0.5L_y}{500} \right) - \mathrm{e}^{-\left(\frac{r_0}{200}\right)^2} \right] . \qquad (64)$$

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 depth-profile is shown in Figure 8. The initial conditions were set to

$$\begin{aligned} \eta(x, y, 0) &= \eta_0 \mathrm{e}^{-\frac{(y-0.1L_y)^2}{3200}} , & (65) \\ v(x, y, 0) &= \sqrt{\frac{g}{H_1}} \eta(x, y, 0) , & (66) \\ u(x, y, 0) &= 0 . & (67) \end{aligned}$$

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

Snapshots of the evolving $\eta$ field are shown in Figure 9. By $t = 70$ s (panel **(b)**), the symmetry in the $x$-direction has been broken due to refraction as the portion of the wave front propagating over the deep region in the center of the domain (near $x = 0.5L_x$) has a faster effective wave speed than the portion of the wave front that is propagating over the shoal. As the initial wave shoals, solitary waves emerge. The solitary waves are most evident in panels **(b)**,**(d)**, and **(f)** when the main wave front is situated on top of the shoal. At the later times (panels **(c)**-**(f)**), an interesting interference pattern of relatively weak waves follows the main wave front due to wave scattering and solitary wave fissioning induced by the topography.

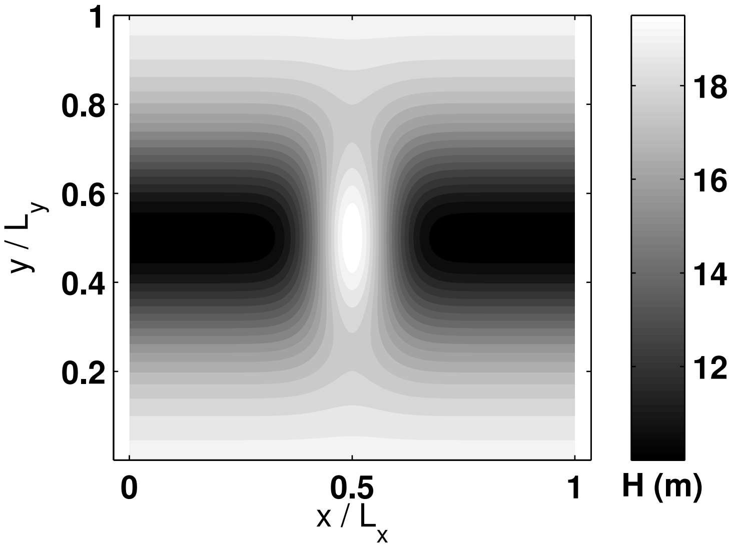


Figure 8: The depth-profile used in the *wave propagation over topography* run, corresponding to equation (64).

While the decrease in relative amplitude of the portion of the main wave front over the deep region to that over the shallow region is intuitive, it should be noted that the absolute wave amplitude decreases over the deep region as well. This effect occurs due to the bending of wavefronts towards the lines of constant depth, which leads to a divergence of energy over the deep region. Thus, there is a corresponding focusing of energy near $0.3L_x < x < 0.4L_x$ and $0.6L_x < x < 0.7L_x$ that is more clearly seen in plots of $u$, the $x$-component of velocity, shown at times $t = 70$ s, $t = 210$ s, and $t = 350$ s in Figure 10. In field situations, this energy focusing could have implications for wave-boundary-layer interactions. It is also interesting to note that both the solitary wave widening and fissioning is qualitatively consistent with KdV theory (Whitham, 1999).

## 4. Conclusions

In this manuscript, we have introduced a Fourier pseudospectral method for solving a dispersive shallow water model of the Boussinesq type in periodic domains with variable water depth. In line with previous studies (Eskilsson and Sherwin, 2005; Karniadakis and Sherwin, 2005), we dis-

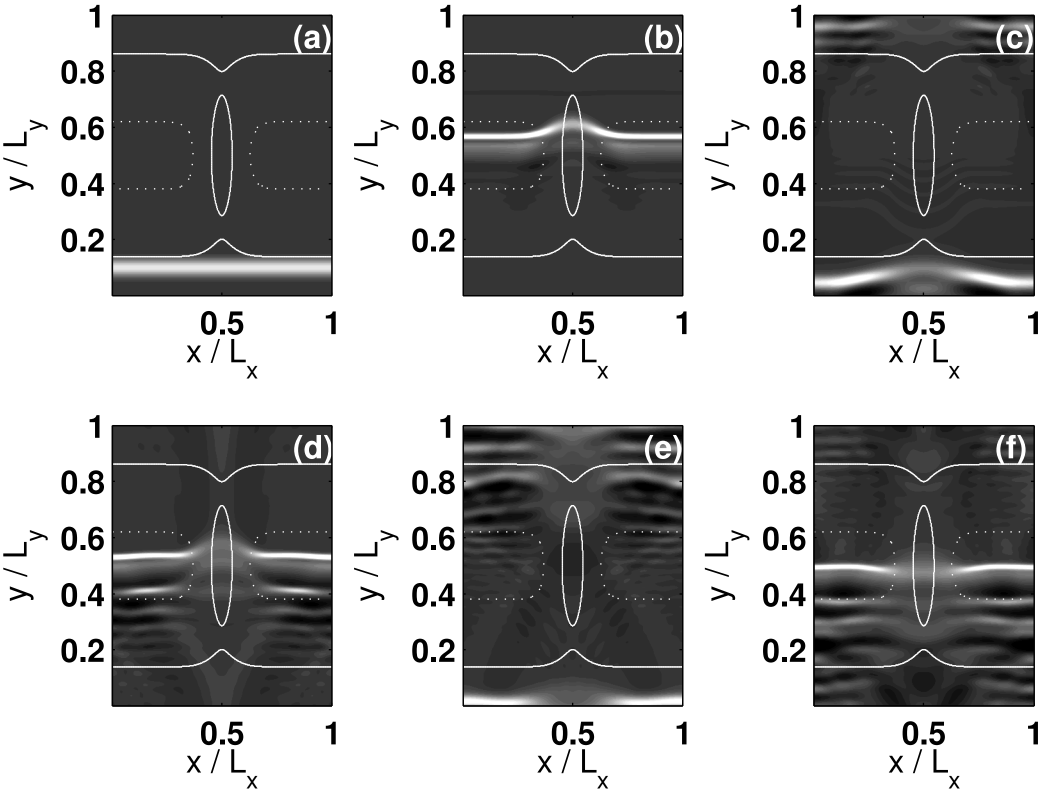


Figure 9: Fixed time snapshots of the free surface displacement at **(a)** $t = 0$, **(b)** $t = 70$ s, **(c)** $t = 140$ s, **(d)** $t = 210$ s, **(e)** $t = 280$ s, **(f)** $t = 350$ s in the *wave propagation over topography* run. The contours are given by 33 equally spaced values between $-1.25$ 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.

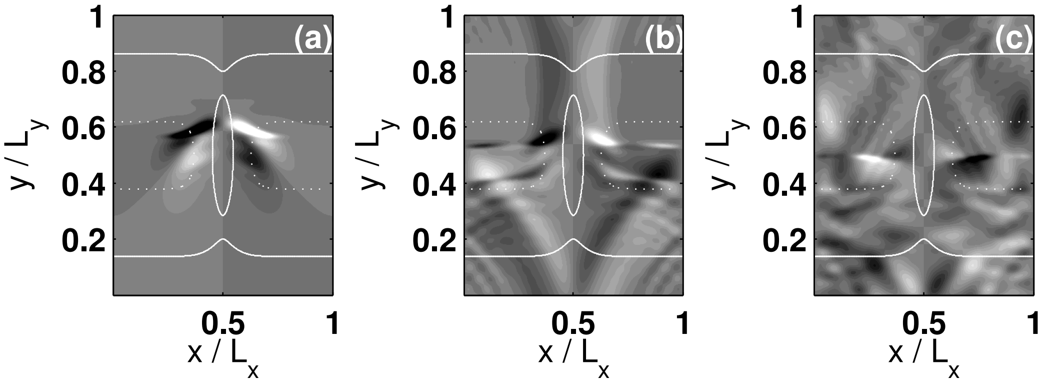


Figure 10: Fixed time snapshots of the $u$ field ($x$-component of velocity) at **(a)** $t = 70$ s, **(b)** $t = 210$ s **(c)** $t = 350$ s in the *wave propagation over topography* run, corresponding to Figures 9**(b)**,**(d)**,**(f)**, respectively. The contours are given by 21 equally spaced values 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.

cussed two approaches for the time-discretization method, the so-called "coupled" and "scalar" approaches. Although both methods are stable, the scalar approach reduces the dimension of the resulting linear systems to be 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 exponential convergence rate of the Fourier spatial discretization method was verified in two-dimensions by comparing numerical solutions to the exact solution of an elliptic problem. 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-order 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 agreed qualitatively with past analytical and numerical results of resonant wave generation by flow over topography (Grimshaw and Smyth, 1986; Esler et al., 2007).

A second two-dimensional simulation corresponding to a long wave propagating over a shoal was carried out. Interesting linear and nonlinear phenomena such as wave scattering, steepening, and the emergence of fissioning solitary waves (only in sufficiently shallow regions) rapidly broke the symmetry of the initial conditions resulting in a rather complicated final wave field with a variety of fine scale features.

There are many possible improvements and extensions one could make to the methodology presented here. Improvements include using a higher-order time discretization method for improved accuracy and using an adaptive filtering procedure to minimize the amount of filtering required for numerical stability. Another significant improvement would come from using a scalable multigrid/domain decomposition approach for the elliptic problem so that higher resolution simulations can be carried out on parallel computing clusters. Extensions include replacing the Fourier discretization in one or both of the spatial directions with a Chebyshev pseudospectral discretization (Boyd (2001)) so that simulations in periodic channels and specialized closed basins may be carried out. The particular case of an annular circular basin with application to mid-sized lakes has been explored in Steinmoeller et al. (2012, in press). Another possible extension would solve a multi-layer extension of the system (1)-(3) (e.g. de la Fuente et al. (2008); Cotter et al. (2010)) as a suitable model of internal waves in a density-stratified fluid. Finally, one may extend the DG-FEM methodology presented in Section 3.3 to the case of two-dimensional arbitrary closed basins using triangulated unstructured grids to model wave dynamics in real-world lakes with a realistic representation of the coast-line.

## Appendix A. Discontinuous Galerkin Spatial Discretization Method

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

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

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

$$\gamma \frac{\partial^2 z}{\partial x^2} - z = -\frac{\partial a}{\partial x} \, , \tag{A.3}$$

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 $\mathbf{D}^k = [x_l^k, x_r^k]$, $k = 1, \cdots, K$. Each element is then discretized on $N+1$ points, using the Legendre-Gauss-Lobotto polynomial interpolation nodes. We proceed by representing the numerical solutions locally on each element in terms of the Lagrange interpolating polynomials, i.e.,

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

with

$$\ell_i^k(x) := \prod_{\substack{m=1\\ m\neq i}}^{N+1} \frac{x^k - x_m^k}{x_i^k - x_m^k} \,, \tag{A.5}$$

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

$$\begin{aligned}
(\ell_i^k, \ell_j^k)_{\mathbf{D}^k} \frac{dh_j^k}{dt} + (\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} hu_j^k &= \left[\ell_j^k \left((hu)^k - (hu)^*\right)\right]_{x_l^k}^{x_r^k} \,, && \text{(A.6)}\\
(\ell_i^k, \ell_j^k)_{\mathbf{D}^k} \frac{dhu_j^k}{dt} + (\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} f_j^k &= \left[\ell_j^k \left(f^k - f^*\right)\right]_{x_l^k}^{x_r^k} && \text{(A.7)}\\
&+ \gamma(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} z_j^k - \gamma\left[\ell_j^k \left(z^k - z^*\right)\right]_{x_l^k}^{x_r^k} \,,
\end{aligned}$$

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.,

$$f^* = \{\!\{f\}\!\} + \frac{\lambda}{2}\llbracket u \rrbracket \,, \tag{A.8}$$

where

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

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

$$\begin{aligned} \sqrt{\gamma}(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} q_j^k - (\ell_i^k, \ell_j^k) z_j^k &- \left[\ell_j^k\left(\sqrt{\gamma}q^k - \sqrt{\gamma}q^*\right)\right]_{x_l^k}^{x_r^k} , \qquad \text{(A.10)} \\ &= -(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} a_j^k + \left[\ell_j^k\left(a^k - a^*\right)\right]_{x_l^k}^{x_r^k} \\ (\ell_i^k, \ell_j^k)_{\mathbf{D}^k} q_j^k = \sqrt{\gamma}(\ell_i^k, \frac{d\ell_j^k}{dx})_{\mathbf{D}^k} z_j^k &- \sqrt{\gamma}\left[\ell_j^k\left(z^k - z^*\right)\right]_{x_l^k}^{x_r^k} , \qquad \text{(A.11)} \end{aligned}$$

where we choose $a^* = \{\!\{a\}\!\}$, $z^* = \{\!\{z\}\!\}$, together with the stabilized (or penalized) central flux $q^* = \{\!\{q\}\!\} - \tau[\![z]\!]$, $\tau > 0$ for the auxiliary variable, $q$. The penalty term's purpose is to remove the null eigenmode that would be present if $\tau = 0$, (Hesthaven and Warburton, 2008). Our choice of numerical fluxes for the elliptic problem is essentially a stabilized version of the fluxes used by Bassi and Rebay (1997) for a DG-FEM discretization of the viscous terms in the compressible Navier-Stokes equations. In the results presented in Section 3.3, we use the value $\tau = 1$ for the stabilization parameter. It is known that the convergence rate of the solutions to the discretized elliptic problem is sensitive to the choice of $\tau$, and ideal scalings for $\tau$, dependent on grid-spacing and polynomial order, have been suggested in the literature (Eskilsson and Sherwin, 2005; Hesthaven and Warburton, 2008). However, since the DG-FEM simulations presented in Section 3.3 are well resolved, we do not expect our choice of $\tau$ to affect the quality of the solutions.

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

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

## Appendix B. Derivation of the WKB solution

To begin, we introduce the slowly-varying spatial coordinate

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

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

$$\eta_{tt} - \epsilon^2 \left(gH\eta_\chi\right)_\chi = 0 \ , \tag{B.2}$$

where $H = H(\chi)$ only. It is worth noting that this approximate equation does not contain any dispersive terms such as those included in the full system (1)–(3), so the approximation is only expected to be accurate for waves that are sufficiently 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

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

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

$$\psi(\chi) = \mathrm{e}^{i\left(\frac{S_0}{\epsilon}(\chi)+S_1(\chi)+\epsilon S_2(\chi)+\cdots\right)} \ , \tag{B.4}$$

such that

$$\begin{aligned} \frac{S_0}{\epsilon} &\gg S_1 \gg \epsilon S_2 \gg \cdots \ , & \text{(B.5)} \\ \epsilon S_2 &\ll 1 \ , \ \text{as } \epsilon \to 0 \ . & \text{(B.6)} \end{aligned}$$

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

$$\begin{aligned} S_0(\chi) &= \pm \int_0^\chi \frac{\sigma}{\sqrt{gH(\zeta)}} d\zeta \ , & \text{(B.7)} \\ S_1(\chi) &= \frac{i}{2} \ln |HS_0'| = \frac{i}{2} \ln \left| \sigma \sqrt{\frac{H}{g}} \right| , & \text{(B.8)} \end{aligned}$$

where prime ($'$) denotes differentiation with respect to $\chi$. Thus, we have

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

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

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

## Acknowledgement

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

## References

Amezcua, J., Kalnay, E., Williams, P., 2011. The Effects of the RAW Filter on the Climatology and Forecast Skill of the SPEEDY Model. Mon. Weather Rev. 139, 608–619.

Bassi, F., Rebay, S., 1997. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J. Comp. Phys. 131, 267–279.

Boussinesq, J., 1872. Théorie des ondes et des remous qui se propagent le long d'un canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement pareilles de la surface au fond. Journal de Mathématique Pures et Apliquée, Deuxième Série **17**, 55–108.

Boyd, J., 2001. Chebyshev and Fourier Spectral Methods, 2nd Edition. Dover Publications.

Brandt, P., Rubino, A., Alpers, W., Backhaus, J., 1997. Internal waves in the Strait of Messina Studied by a numerical model and synthetic aperture radar umages from *ERS 1/2* Satellites. J. Phys. Oceangr. **27**, 648–663.

Christov, C., 2000. An energy-consistent dispersive shallow-water model. Wave Motion **1018**, 1–14.

Cotter, C. J., Holm, D. D., Percival, J. R., 2010. The square root depth wave equations. Proc. R. Soc. A 466, 3621–3633.

de la Fuente, A., Shimizu, K., Imberger, J., Niño, Y., 2008. The evolution of internal waves in a rotating, stratified, circular basin and the influence of weakly nonlinear and nonhydrostatic accelerations. Limnol. Oceanogr. **53**(6), 2738–2748.

Engsig-Karup, A., Hesthaven, J., Bingham, H., Madsen, P., 2006. Nodal DG-FEM solution of high-order Boussinesq-type equations. J. Eng. Math. **56**, 351–370.

Eskilsson, C., Sherwin, S., 2005. Spectral/*hp* discontinuous Galerkin methods for modelling 2d Boussinesq equations. J. Sci. Comp. **22**, 269–288.

Esler, J., Rump, O., Johnson, E., 2007. Non-dispersive and weakly dispersive single-layer flow over an axisymmetric obstacle: the equivalent aerofoil formulation. J. Fluid Mech. 574, 209–237.

Golub, G., Van Loan, C., 1996. Matrix Computations, 3rd Edition. The Johns Hopkins University Press.

Grimshaw, R., Smyth, N., 1986. Resonant flow of a stratified fluid over topography. J. Fluid Mech. 169, 429–464.

Hesthaven, J., Warburton, T., 2008. Nodal Discontinuous Galerkin Methods. Springer.

Iserles, A., 1996. A First Course in the Numerical Analysis of Differential Equations. Cambridge University Press.

Karniadakis, G., Sherwin, S., 2005. Spectral/hp Element Methods for Computational Fluid Dynamics, 2nd Edition. Oxford University Press, USA.

Leveque, R., 2007. Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. Society for Industrial and Applied Mathematics.

Lynett, P., Liu, P. L.-F., 2004. A two-layer approach to wave modelling. Proc. R. Soc. Lond. A **460**, 2637–2669.

Madsen, P., Bingham, H., Liu, H., 2002. A new Boussinesq method for fully nonlinear waves from shallow to deep water. J. Fluid Mech. **462**, 1–30.

Madsen, P., Murray, R., Sørensen, O., 1991. A new form of the Boussinesq equations with improved linear dispersion characteristics. Coastal Engineering **15**, 371–388.

Nwogu, O., 1993. Alternative form of Boussinesq equations for nearshore wave propagation. J. Waterw. Port Coast. Ocean Eng. **119**(6), 618–638.

Peregrine, D., 1967. Long waves on a beach. J. Fluid. Mech. **27**(4), 815–827.

Peyret, R., 2002. Spectral Methods for Incompressible Viscous Flow. Springer-Verlag New York, Inc.

Smith, B., Bjorstad, P., Gropp, W., 2004. Domain Decomposition: Parallel multilevel methods for elliptic partial differential equations. Cambridge University Press.

Steinmoeller, D., Stastna, M., Lamb, K., 2012, in press. Pseudospectral methods for Boussinesq-type equations in an annular domain with applications to mid-sized lakes. J. Comp. Sci.

Trefethen, L., 2000. Spectral Methods in MATLAB. Society for Industrial and Applied Mathematics.

Trefethen, L., Bau, D., 1997. Numerical Linear Algebra. Society for Industrial and Applied Mathematics.

Trottenberg, U., Oosterlee, C., Schuller, A., 2000. Multigrid, 1st Edition. Academic Press.

Walkley, M., 1999. A numerical method for extended boussinesq shallow-water wave equations. Ph.D. thesis, University of Leeds, UK.

Wei, G., Kirby, J., 1995. Time-dependent numerical code for extended Boussinesq equations. J. Waterw. Port Coast. Ocean Eng. **121**, 251–261.

Whitham, G., 1999. Linear and Nonlinear Waves. Wiley-Interscience.

Williams, P., 2011. The RAW Filter: An Improvement ot the Robert–Asselin Filter in Semi-Implicit Integrations. Mon. Weather Rev. 139, 1996–2007.