JOURNAL OF COMPUTATIONAL PHYSICS **143,** 224–240 (1998)
ARTICLE NO. CP985957

Traveling Water Waves: Spectral Continuation Methods with Parallel Implementation

David P. Nicholls

Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912
E-mail: nicholls@cfm.brown.edu

Received September 17, 1997; revised February 24, 1998

We present a numerical continuation method for traveling wave solutions to the full water wave problem using a spectral collocation discretization. The water wave problem is reformulated in terms of surface variables giving rise to the Zakharov–Craig–Sulem formulation, and traveling waves are studied by introducing a phase velocity vector as a parameter. We follow non-trivial solution branches bifurcating from the trivial solution branch via numerical continuation methods. Techniques such as projections and filtering allow the computation to proceed for greater distances up the branch, and parallelism allows the computation of larger problems. We conclude with results including the formation of hexagonal patterns for the three dimensional problem. © 1998 Academic Press

1. INTRODUCTION

The study of inviscid, irrotational, free-surface flows, known as the water wave problem, has interested researchers for many years. The observations by Russell of solitary waves inspired the nineteenth century work of Boussinesq, Rayleigh, and others on long waves (see Section 10.9 of Stoker [17]). Later on Levi-Civita [13] proved the existence of periodic traveling waves of infinite depth, while Struik [18] proved the case of finite depth. More recently Hammack *et al.* [9, 8] and Milewski and Keller [15] have studied traveling wave solutions of approximations to the three dimensional water wave equations. Hammack *et al.* began their research with wave tank experiments of two interacting traveling waves and noted hexagonal wave forms. They were able to reproduce these using exact solutions to the KP equation [10] which serves as a three dimensional analogue to the KdV equation. Milewski and Keller were also able to reproduce these and other shapes using their own model. Our research focuses on traveling wave solutions to the Zakharov–Craig–Sulem (ZCS) formulation of the water wave problem which is equivalent to the full water wave equations. Zakharov recognized that the water wave problem could be written as a Hamiltonian system [22] while

Craig and Sulem introduced the Dirichlet–Neumann operator into the formulation [5]. Craig and Sulem studied the time dependent two dimensional version of these equations [5] while many people have studied various aspects of traveling wave solutions of the two dimensional water wave problem, [2, 6, 12, 19, 20, 21, 23]. Schanz studied the three dimensional time dependent ZCS formulation [16], but little else has been done for the three dimensional full water wave problem.

The goal of this paper is to find hexagonal solutions of the water wave problem using numerical continuation methods applied to traveling wave solutions of the ZCS formulation. In Section 2 we state the water wave equations, show how they are equivalent to the ZCS formulation, introduce the Dirichlet–Neumann operator, and then derive the equations for traveling wave solutions. In Section 3 we discuss the spectral collocation discretization of the problem, briefly outline the continuation methods used, and then discuss our implementation of projections, filters, and parallelism. In Section 4 we mention a few of our numerical results in two and three dimensions. Most importantly we show three dimensional waves which exhibit the hexagonal shape discussed above.

2. TRAVELING WAVE SOLUTIONS OF THE WATER WAVE PROBLEM

The water wave problem consists of describing the evolution of the free surface of an incompressible, inviscid, irrotational fluid under the influence of gravity. Our version of the problem ignores the effects of surface tension, while it includes the effects of a bottom by fixing one at a constant depth of $-h$. The problem can be formulated in n dimensions, an $(n-1)$ dimensional bottom and one vertical dimension, but is clearly most useful in two and three dimensions for the modeling of ocean waves.

2.1. *Classical Equations and Zakharov–Craig–Sulem Formulation*

It is well known that inside the fluid region $S_\eta = \{(x, y) \in \mathbf{R}^n \mid -h \leq y \leq \eta(x, t)\}$, we can define a velocity potential $\varphi(x, y, t)$ which satisfies Laplace's equation,

$$\Delta\varphi = 0. \tag{2.1a}$$

Once the velocity potential is found the fluid velocity can be computed as $(u, v)^T = \nabla\varphi$. Boundary conditions for φ are

$$\partial_y\varphi = 0 \qquad \text{at } y = -h \tag{2.1b}$$

$$\partial_t\eta + \nabla_x\eta \cdot \nabla_x\varphi - \partial_y\varphi = 0 \qquad \text{at } y = \eta(x, t) \tag{2.1c}$$

$$\partial_t\varphi + \frac{1}{2}|\nabla\varphi|^2 + g\eta = 0 \qquad \text{at } y = \eta(x, t). \tag{2.1d}$$

For our problem the horizontal boundary conditions will be periodic. For the two dimensional problem the surface is parameterized by an interval, while in three dimensions the surface is parameterized by a parallelogram which is itself determined by a lattice Γ. We recall that a lattice $\Gamma \subset \mathbf{R}^n$ generated by the non-singular matrix $A \in \mathbf{R}^{n\times n}$ acting on integer vectors $j \in \mathbf{Z}^n$ has a conjugate lattice $\Gamma' \subset \mathbf{R}^n$ generated by $2\pi(A^T)^{-1}$. A well-behaved function f periodic on the parallelogram $P(\Gamma) = \mathbf{R}^n/\Gamma$ can be expressed by its Fourier expansion,

$$f(x) = \sum_{k\in\Gamma'} \hat{f}(k)e^{ik\cdot x}. \tag{2.2}$$

These equations and boundary conditions are what we have denoted as the classical equations for the water wave problem. The following paragraph introduces the Zakharov–Craig–Sulem (ZCS) formulation which constitutes a surface integral formulation of the above equations [5, 22].

Since we are solving Laplace's equation in the fluid region and we have the boundary conditions at the other boundaries, once we have found the free surface η and the boundary values of the potential at the free surface we can theoretically solve for the potential. Zakharov's method consists of a reduction of the problem from one inside the entire fluid to one at the surface which reduces the dimension of the problem. With this in mind we introduce $\xi(x,t) = \varphi(x, \eta(x,t), t)$ as the potential evaluated at the surface. Upon inspecting the equations at the free surface, Eqs. (2.1d) and (2.1c), it is clear that the only reference to φ on the interior of the fluid domain is through its normal derivative at the surface. Craig and Sulem [5] found that a convenient way to formulate this is to introduce the Dirichlet–Neumann operator, $G(\eta)$, which is the operator that takes Dirichlet data at the fluid surface to Neumann data at the surface. Since the outward normal to the fluid is given by $N_\eta = (-\nabla_x \eta, 1)^T$ and the unit outward normal is given by $n_\eta = N_\eta / |N_\eta|$ we can express the Dirichlet–Neumann operator as

$$G(\eta)\xi = (1 + |\nabla_x \eta|^2)^{\frac{1}{2}} \nabla\varphi \cdot n_\eta. \tag{2.3}$$

We note that the right hand side of Eq. (2.3) is not exactly what one would expect for Neumann data since it is scaled by an extra factor of $|N_\eta|$; however, this definition is effectively equivalent and yields a cleaner set of equations, Eq. (2.5). Using the following relations, which follow from the chain rule,

$$\begin{aligned}
\nabla_x \varphi|_\eta &= \nabla_x \xi - (\partial_y \varphi|_\eta) \nabla_x \eta && (2.4a)\\
\partial_y \varphi|_\eta &= G(\eta)\xi + \nabla_x \varphi|_\eta \cdot \nabla_x \eta && (2.4b)\\
\partial_y \varphi|_\eta &= \frac{1}{|N_\eta|^2} \{G(\eta)\xi + \nabla_x \xi \cdot \nabla_x \eta\} && (2.4c)\\
\partial_t \varphi|_\eta &= \partial_t \xi - \partial_t \eta \partial_y \varphi|_\eta, && (2.4d)
\end{aligned}$$

it is not difficult to transform Eqs. (2.1d) and (2.1c) into the equivalent ZCS formulation of the water wave problem,

$$\begin{aligned}
\partial_t \eta &= G(\eta)\xi && (2.5a)\\
\partial_t \xi &= -g\eta - \frac{1}{2|N_\eta|}[|\nabla_x \xi|^2 - (G(\eta)\xi)^2 - 2(G(\eta)\xi)\nabla_x \xi \cdot \nabla_x \eta \\
&\quad + |\nabla_x \xi|^2 |\nabla_x \eta|^2 - (\nabla_x \xi \cdot \nabla_x \eta)^2]. && (2.5b)
\end{aligned}$$

2.2. *The Dirichlet–Neumann Operator*

It can be shown that the Dirichlet–Neumann operator is analytic as a function of η if the supremum norm and Lipschitz norm of η are bounded by a constant [4]. Therefore we can write $G(\eta)$ in terms of a convergent Taylor series expansion,

$$G(\eta) = \sum_{j=0}^{\infty} G_j(\eta), \tag{2.6}$$

where each term $G_j(\eta)$ is homogeneous of degree j. It is not difficult to show [5] that the zeroth order term, corresponding to the case of flat water ($\eta = 0$), is,

$$G_0\xi(x) = |D_x| \tanh(h|D_x|)\xi(x), \tag{2.7}$$

where $D_x = -i\nabla_x$. Higher order terms can be derived from a recursive formula obtained by Craig and Sulem [5]. This formula is derived by considering functions $\varphi_k(x, y) = e^{ik\cdot x}\cosh(|k|(y+h))$, $k \in \Gamma'$, which span the set of harmonic functions which also satisfy both the periodic and the bottom boundary conditions. Putting these functions into the relationship,

$$\partial_y\varphi_k - \nabla_x\eta \cdot \nabla_x\varphi_k = G(\eta)\varphi_k \qquad \text{at } y = \eta, \tag{2.8}$$

expanding the functions $\cosh(|k|(\eta + h))$ and $\sinh(|k|(\eta + h))$ about $\eta = 0$, and equating terms of corresponding degree in η we obtain for $j = 2r > 0$,

$$\begin{aligned} G_{2r}(\eta) = {}& \frac{1}{(2r)!} D_x \cdot \eta^{2r} D_x(|D_x|^2)^{r-1}G_0 \\ & - \sum_{s=0}^{r-1} \frac{1}{(2(r-s))!} G_{2s}(\eta)\left[\eta^{2(r-s)}(|D_x|^2)^{r-s}\right] \\ & - \sum_{s=0}^{r-1} \frac{1}{(2(r-s)-1)!} G_{2s+1}(\eta)\left[\eta^{2(r-s)-1}(|D_x|^2)^{r-s-1}G_0\right]. \end{aligned} \tag{2.9}$$

For $j = 2r - 1 > 0$ we have

$$\begin{aligned} G_{2r-1}(\eta) = {}& \frac{1}{(2r-1)!} D_x \cdot \eta^{2r-1} D_x(|D_x|^2)^{r-1} \\ & - \sum_{s=0}^{r-1} \frac{1}{(2(r-s)-1)!} G_{2s}(\eta)\left[\eta^{2(r-s)-1}(|D_x|^2)^{r-s-1}G_0\right] \\ & - \sum_{s=0}^{r-2} \frac{1}{(2(r-s-1))!} G_{2s+1}(\eta)\left[\eta^{2(r-s-1)}(|D_x|^2)^{r-s-1}\right]. \end{aligned} \tag{2.10}$$

For the current problem the real utility of the above formulae is in their application to spectral methods. Since the formulae consist of multiplications and applications of Fourier multipliers composed with one another, a natural choice for applying such an operator is via a spectral collocation method. The boundary conditions are periodic so we will use a Fourier collocation method which is a method of weighted residuals with sines and cosines as trial functions, and translated Dirac delta functions with poles at the equally spaced collocation points as the test functions [3]. Thus the method is to approximate the functions of interest, $\eta(x)$ and $\xi(x)$, by expanding them in a finite Fourier series, insert these expansions into the partial differential equation, project the equations onto the space of delta functions, and then minimize the residual. For our problem this amounts of evaluating all differential operators in Fourier space (this includes all Fourier multipliers) while performing all multiplications in physical space. For example, if we wish to apply the operator G_0 to a function ξ in physical space we transform ξ to Fourier space, apply the diagonal operator $|k|\tanh(h|k|)$

to the Fourier coefficients of ξ, and then transform back to physical space. To evaluate the more complicated operator $G_1(\eta)$ we use the formula

$$G_1(\eta) = D_x \cdot \eta D_x - G_0 \eta G_0, \tag{2.11}$$

and note that $D_x \cdot \eta D_x \xi = D_x \cdot [\eta D_x[\xi]]$ and $G_0 \eta G_0 \xi = G_0[\eta G_0[\xi]]$. To evaluate the first part of Eq. (2.11) acting on a function ξ we transform to Fourier space, multiply by the diagonal operator k, transform back to physical space, multiply by the diagonal operator η, transform to Fourier space, multiply by the diagonal operator k, and finally transform back to physical space. Computations of other such terms are completed in an analogous way. For speed all transforms are computed via the fast Fourier transform (FFT) algorithm.

Our procedure for computing an approximation to the Dirichlet–Neumann operator is therefore to choose a desired number of terms in the Taylor expansion, follow the above procedure for evaluating each of the terms individually, and then sum the result. We note that there are several drawbacks with such a procedure. First of all, there are the typical problems of aliasing, and growth of spurious modes. However, there is also the problem that to evaluate a term of order j requires evaluations of all terms of order $l, l < j$. Thus the procedure is highly recursive and impractical for very high order terms. In practice we have used expansions of order 4 and 5 which give excellent results with reasonable cost.

2.3. *Traveling Wave Solutions*

Our goal is to study traveling wave solutions to the water wave problem. Therefore we introduce the phase velocity vector $c \in \mathbf{R}^{n-1}$, where $n = 2$ or 3 corresponding to two and three dimensions, respectively, into the classic water wave equations and then, as with the ZCS formulation, use the Dirichlet–Neumann operator to express them as surface integral equations. The classic equations for a steady flow in a frame moving with velocity c are

$$\Delta\varphi = 0 \qquad \text{in } S_\eta \tag{2.12a}$$

$$\partial_y \varphi = 0 \qquad \text{at } y = -h \tag{2.12b}$$

$$[c \cdot \nabla_x]\varphi + \frac{1}{2}|\nabla\varphi|^2 + g\eta = 0 \qquad \text{at } y = \eta \tag{2.12c}$$

$$[c \cdot \nabla_x]\eta + \nabla_x \eta \cdot \nabla_x \varphi - \partial_y \varphi = 0 \qquad \text{at } y = \eta. \tag{2.12d}$$

By introducing the Dirichlet–Neumann operator into these equations (2.12), it can be shown that traveling wave solutions to the water wave problem satisfy $F(\eta, \xi, c) = 0$ where

$$\begin{aligned} F_1(\eta, \xi, c) &= g\eta + [c \cdot \nabla_x]\xi + \frac{1}{2|N_\eta|}[|\nabla_x \xi|^2 - (G(\eta)\xi)^2 - 2(G(\eta)\xi)\nabla_x \xi \cdot \nabla_x \eta \\ &\quad + |\nabla_x \xi|^2 |\nabla_x \eta|^2 - (\nabla_x \xi \cdot \nabla_x \eta)^2] \end{aligned} \tag{2.13a}$$

$$F_2(\eta, \xi, c) = -[c \cdot \nabla_x]\eta + G(\eta)\xi. \tag{2.13b}$$

Solutions of the system $F(\eta, \xi, c) = 0$ will be studied using a bifurcation analysis, and the central issue in such an analysis is the linearization of F about a branch of known solutions and its null space. In our case the trivial solutions $(\eta(x) = 0, \xi(x) = 0, c)$ form a two dimensional branch of known solutions, and the linearization of F about the trivial solutions is given by

$$A(c) = \partial_u F(\eta = 0, \xi = 0, c) = \begin{pmatrix} g & c \cdot \nabla_x \\ -c \cdot \nabla_x & G_0 \end{pmatrix}, \tag{2.14}$$

where $u = (\eta, \xi)^T$. When $A(c)$ is non-singular we expect vis-à-vis the implicit function theorem that the only solutions are the trivial ones. Therefore we must search at the singularities of $A(c)$ to find bifurcations to branches of non-trivial solutions. If we write u in terms of its Fourier series,

$$u(x) = \begin{pmatrix} \eta(x) \\ \xi(x) \end{pmatrix} = \sum_{k \in \Gamma'} \begin{pmatrix} \hat{\eta}(k) \\ \hat{\xi}(k) \end{pmatrix} e^{ik \cdot x}, \tag{2.15}$$

one can see that the action of $A(c)$ on the Fourier coefficients is given in 2×2 block diagonal form, with the kth block being

$$\hat{A}_k(c) = \begin{pmatrix} g & ic \cdot k \\ -ic \cdot k & |k| \tanh(h|k|) \end{pmatrix}. \tag{2.16}$$

Therefore, the matrix $A(c)$ is singular for c such that $\Delta(c, k) = g|k| \tanh(h|k|) - (c \cdot k)^2 = 0$ for some $k \in \Gamma'$. The corresponding null vectors are

$$\psi_1(c, k) = \begin{pmatrix} c \cdot k \cos(k \cdot x) \\ -g \sin(k \cdot x) \end{pmatrix} \tag{2.17a}$$

$$\psi_2(c, k) = \begin{pmatrix} c \cdot k \sin(k \cdot x) \\ g \cos(k \cdot x) \end{pmatrix}. \tag{2.17b}$$

If we consider the two dimensional problem and are given a $k \in \Gamma'$, there is a unique, up to sign, wave speed c which makes $\hat{A}_k(c)$ singular. At this point we have the null vectors $\psi_1(c, k)$ and $\psi_2(c, k)$ and we are free to travel up the bifurcation branch in any direction spanned by these two vectors. However, continuation codes, which we outline below, are much more robust at bifurcation points which have a one dimensional null space. In light of this we will choose to travel in the ψ_1 direction and completely project out the ψ_2 direction. It is equivalent, and useful for programming purposes, to demand that the mode $\hat{\eta}(k)$ be real. This is of course arbitrary, but enforcing this simply fixes the crest of the wave at $x = 0$ and it reduces the dimension of the null space to one.

In the three dimensional problem given a $k \in \Gamma'$ there are two parallel *lines* in the (c_1, c_2) plane of values which make $\hat{A}_k(c)$ singular. At an arbitrary point on these lines we can find a bifurcation branch consisting of solutions of the two dimensional problem. Interesting points occur at the intersection of two lines arising from different wave numbers. Therefore we choose two wave numbers, κ_1 and κ_2, and find the intersection points of the lines generated by these wave numbers. At any one of these points we have a null space spanned by four vectors, $\psi_1(\kappa_1)$, $\psi_2(\kappa_1)$, $\psi_1(\kappa_2)$, and $\psi_2(\kappa_2)$. Again, we wish to reduce the dimension of the null space to one, so we seek solutions which are a product of $\psi_1(\kappa_1)$ and $\psi_1(\kappa_2)$ with equal amplitude. This results in a product of cosines in the η component. Once we have moved onto the bifurcation branch and proceeded up a short distance, these solutions will form the hexagonal ones mentioned earlier. Equivalently, again for programming purposes, this choice amounts to demanding that the modes $\hat{\eta}(\kappa_1)$ and $\hat{\eta}(\kappa_2)$ be real and equal. One final degree of freedom that must be eliminated is due to the fact that there is a two dimensional parameter, c. We eliminate this by choosing to move along a line in (c_1, c_2) space.

3. NUMERICAL METHODS

Now that we have derived surface integral equations for traveling wave solutions to the water wave problem we must reduce this to a finite dimensional problem, $H(\overline{v}, c) = 0$, $H : \mathbf{R}^N \times \mathbf{R} \to \mathbf{R}^N$, via some discretization method. As mentioned in Subsection 2.2, we accomplish this using a spectral collocation method which gives spectral convergence for the analytic solutions which we seek. However, these methods have difficulties such as aliasing and spurious growth of high wave number modes which we must overcome in order to proceed any distance up the bifurcation branch. Once we have arrived at this finite dimensional problem we must implement a continuation method to move us along the surface of solutions once we have located it. For the most part we use the standard methods outlined in the book of Allgower and Georg [1]. These methods generally work well with some noteworthy modifications which are necessary to select a solution from among the many possible solutions. This ambiguity regarding possible solutions is due to the symmetries inherent to our formulation, in particular the boundary conditions, of the water wave problem, and manifests itself by allowing two dimensional bifurcation surfaces rather than one dimensional bifurcation branches.

3.1. *Spectral Collocation Implementation*

As we mentioned in Subsection 2.2 above we use a spectral collocation method to discretize the problem $F(u, c) = F(\eta, \xi, c) = 0$ into the problem $H(\overline{v}, c) = 0$. The trial functions X_N are the trigonometric polynomials of degree less than or equal to $\frac{N}{2}$ and thus we approximate u by $u_N \in X_N$. This function is represented by the truncated Fourier series,

$$u_N(x) = \begin{pmatrix} \eta_N(x) \\ \xi_N(x) \end{pmatrix} = \sum_{k \in \Gamma'_N} \begin{pmatrix} \hat{\eta}(k) \\ \hat{\xi}(k) \end{pmatrix} e^{ik\cdot x}, \tag{3.1}$$

where Γ'_N is the set of wave numbers less than or equal to $\frac{N}{2}$ in the norm of the lattice Γ'. The test functions Y_N are the Dirac delta functions $\delta(x - x_j)$ where the x_j are the N^{n-1} collocation points, equally spaced grid points within the parallelogram $P(\Gamma) = \mathbf{R}^n / \Gamma$. We substitute the function u_N into $F(u, c) = 0$, project this equation onto the set of test functions Y_N, and then demand that the residual be minimized. This is equivalent to demanding that the equation be satisfied at the collocation points,

$$F(u_N(x), c)|_{x=x_j} = 0. \tag{3.2}$$

If we introduce the vector $\overline{v}$ into the problem by letting $\overline{v}_j = u_N(x_j)$, where the x_j are the collocation points, then the discretization of the problem is $H(\overline{v}_j, c) = F(u_N(x), c)|_{x=x_j}$. We will need to be able to transform the $\overline{v}_j$ from physical space to Fourier space; this is accomplished by performing a discrete Fourier transform on the $\overline{v}_j$ which amounts to an approximation of the coefficients $\hat{u}(k)$ by the trapezoid rule. Derivatives and Fourier multipliers are computed by transforming to Fourier space and then applying the associated diagonal or near-diagonal operators, while multiplications are carried out in physical space in a pointwise fashion. Consequently all operations are diagonal or near-diagonal with fast transforms performed in between. However, problems due to aliasing and spurious growth of high wave numbers require special techniques to allow meaningful computations to proceed. These techniques are outlined in Subsection 3.4.

In the next section we will see that a continuation method with a Newton corrector requires the formation of the Jacobian matrix $\partial_u F$. We compute the linear operator $\partial_u F$ analytically (this requires computing the first variation of the Dirichlet–Neumann operator $\delta_n G(\eta)\xi$) and then approximate it numerically using the Fourier collocation method. This presents no new difficulties but we point out that due to the global nature of the basis functions this matrix *must* in general be dense. This presents a problem for the use of iterative methods which we discuss in Subsection 3.5.

For convenience we work primarily in Fourier space due to the greater ease of applying the projections which we outline in Subsection 3.3. The simplest continuation methods require that the continuation parameter be one dimensional. The two dimensional water wave problem satisfies this constraint, but the three dimensional problem has two parameters. In order to reduce the dimension we choose to travel along a straight line in (c_1, c_2) space passing through (c_1^*, c_2^*) with angle $\mathbf{\Psi}$. Therefore we have

$$c_1(\sigma) = c_1^* + \cos(\mathbf{\Psi})\sigma \tag{3.3a}$$

$$c_2(\sigma) = c_2^* + \sin(\mathbf{\Psi})\sigma, \tag{3.3b}$$

and we use σ as the parameter.

3.2. *Continuation Methods*

The numerical methods for solving simple continuation problems are well understood and quite sophisticated. We chose to write our own simple set of basic routines due to our special needs, and implemented the pseudocode found in the book by Allgower and Georg [1] which we outline here.

We wish to solve the nonlinear problem

$$H(\bar{v}, c) = 0, \tag{3.4}$$

where $H : \mathbf{R}^N \times \mathbf{R} \to \mathbf{R}^N$ and H is smooth. We will let $v = (\bar{v}, c)$ and assume that there is a point $v_0 \in \mathbf{R}^{N+1}$ such that $H(v_0) = 0$ and $H'(v_0)$ has maximal rank N. In this case the implicit function theorem gives us a solution curve $\gamma(\alpha)$ such that $H(\gamma(\alpha)) = 0$. We differentiate the previous equation to obtain

$$H'(\gamma(\alpha))\gamma'(\alpha) = 0. \tag{3.5}$$

Thus $\gamma'(\alpha)$ spans $\ker(H'(\gamma(\alpha)))$. We parameterize the curve with respect to arclength implying that $\|\dot{\gamma}(s)\| = 1$ where $\dot{\gamma}(s) = \frac{d\gamma}{ds}$ represents differentiation with respect to arclength. Finally, in order to uniquely specify our tangent we form the *augmented Jacobian*

$$J_{\text{aug}} = \begin{pmatrix} H'(\gamma(s)) \\ \dot{\gamma}(s)^* \end{pmatrix}, \tag{3.6}$$

where $*$ here represents the conjugate transpose, and require that $\det J_{\text{aug}} > 0$. This development inspires the following general definition,

DEFINITION 3.1. Let $A \in \mathbf{R}^{N\times(N+1)}$ with $\operatorname{rank}(A) = N$. The unique vector $t(A) \in \mathbf{R}^{N+1}$ such that $At(A) = 0$, $\|t(A)\| = 1$, and $\det\binom{A}{t(A)^*} > 0$ is called the *tangent vector induced by A*.

With this definition in mind we can think of the problem of following the solution curve as solving the following *defining initial value problem*:

$$\dot{v} = t(H'(v)) \tag{3.7a}$$

$$v(0) = v_0. \tag{3.7b}$$

It can be shown that the right hand side of Eq. (3.7a) is smooth on an open set and thus the classical existence and uniqueness results of ordinary differential equations apply resulting in a locally defined smooth solution $\gamma(s)$.

Our goal is to numerically trace $\gamma(s)$ by computing a sequence of points $\{v_j\}$ near the curve such that

$$\|H(v_j)\| \leq \varepsilon, \tag{3.8}$$

where $\varepsilon > 0$ is a user defined tolerance. Suppose that we have a point $v_j \in \mathbf{R}^{N+1}$ such that $\|H(v_j)\| \leq \varepsilon$. To obtain the next point we make a predictor step w_{j+1}, and then use a corrector to obtain v_{j+1}. The most common predictor in such methods in the *Euler predictor*,

$$w_{j+1} = v_j + \Delta s * t(H'(v_j)), \tag{3.9}$$

where $\Delta s > 0$ is the step size which we will vary based on various convergence criterion (see Allgower and Georg [1, Chap. 6]). For the corrector step we take advantage of the fact that $H(v, c) = 0$ and use a powerful Newton corrector. We wish to find $y \in \mathbf{R}^{N+1}$ such that

$$\|y - w_{j+1}\| = \min_y\{\|y - w_{j+1}\| \ \mid y \in \gamma(s)\}. \tag{3.10}$$

If v_j is close to $\gamma(s)$ and Δs is small enough then this y is unique. Our approximation to y will be v_{j+1}.

The final step is to implement this minimization procedure using Newton's method. If $H : \mathbf{R}^N \to \mathbf{R}^N$ then we could use the classic Newton's method. However, since $H : \mathbf{R}^{N+1} \to \mathbf{R}^N$ we must use the Moore–Penrose inverse in place of the inverse in a Newton-like scheme. If we are given a matrix $A \in \mathbf{R}^{N\times(N+1)}$ of maximal rank we denote the Moore–Penrose inverse by A^+ and recall that $A^+ = A^*(AA^*)^{-1}$. The following central lemma can be proven,

LEMMA 3.2. *If $A \in \mathbf{R}^{N\times(N+1)}$ has maximal rank and $t(A)$ is the induced tangent, then the following are equivalent for all $b \in \mathbf{R}^N$ and $x \in \mathbf{R}^{N+1}$,*

(1) *$Ax = b$ and $t(A)^*x = 0$.*
(2) *$x = A^+b$.*
(3) *x solves* $\min_y\{\|y\| \mid Ay = b\}$.

Now we wish to solve $\min_y\{\|w - y\| \ \mid H(y) = 0\}$. The method of Lagrange multipliers gives a necessary condition on the solution y which is that

$$H(y) = 0 \tag{3.11a}$$

$$y - w = H'(y)^*\lambda, \tag{3.11b}$$

where $\lambda \in \mathbf{R}^N$. The second condition is equivalent to $y - w \in \operatorname{ran}(H'(y)^*) = \{t(H'(y))\}^\perp$, thus y satisfies

$$H(y) = 0 \tag{3.12a}$$

$$t(H'(y))^*(y - w) = 0. \tag{3.12b}$$

To realize Newton's method we linearize the left hand side of Eq. (3.12) about w to get

$$H(y) = H(w) + H'(w)(y - w) + \mathcal{O}(\|y - w\|^2) \tag{3.13a}$$

$$t(H'(y))^*(y - w) = t(H'(w))^*(y - w) + \mathcal{O}(\|y - w\|^2). \tag{3.13b}$$

The solution of the linearized equation, $\mathcal{N}(w)$, therefore solves

$$H(w) + H'(w)(\mathcal{N}(w) - w) = 0 \tag{3.14a}$$

$$t(H'(w))^*(\mathcal{N}(w) - w) = 0. \tag{3.14b}$$

Using Lemma 3.2 we see that the following is equivalent,

$$\mathcal{N}(w) = w - H'(w)^+ H(w). \tag{3.15}$$

This is precisely Newton's method where the Moore–Penrose inverse replaces the inverse. Other topics such as adaptive step sizing, bifurcation point detection and location, and variable order predictors are all discussed in Allgower and Georg [1] and H. B. Keller [11].

3.3. *Projections*

Like most physical problems the water wave problem admits many symmetries. For example, if we have a solution $(\eta(x), \xi(x), c)$ to the two dimensional problem with periodic boundary conditions then $(\eta(x+\theta), \xi(x+\theta), c)$ is also a solution since this merely translates the origin. The presence of such symmetries makes continuation methods difficult since it introduces singularities into the Jacobian matrix $\partial_u F$ which must be inverted at every step. Overcoming these obstacles is not difficult and it is greatly aided by the fact that we work primarily on the Fourier side, but one must be careful to work in a consistent manner in order to obtain a solution with the correct rate of convergence. We will first discuss the types of symmetries and singularities that arise in the water wave problem and how we can theoretically overcome them, and then we outline how to implement these techniques into the algorithm.

The first singularity arises from an ambiguity in the average values of η and ξ. If we let $P(\Gamma)$ denote an n dimensional parallelogram in the lattice Γ, then the average value of a function f periodic on the lattice Γ is

$$\bar{f} = \frac{1}{|P(\Gamma)|} \int_{P(\Gamma)} f(x)\, dx = \hat{f}(0). \tag{3.16}$$

Since the velocity potential is unique only up to a constant, and the average value of the water surface is free to be chosen, we will have a singularity of degree two. We can eliminate this by specifying the values of $\bar{\eta}$ and $\bar{\xi}$, and consequently we demand that $\hat{\eta}(0) = \hat{\xi}(0) = 0$. Numerically this is achieved by demanding that a certain mode, the zeroth one, be zero at every step. Another case where it is useful to set an entire mode to zero is the highest wave number. We always use an even number of collocation points which implies that the highest wave number is purely real. This creates an inconsistency when applying differentiation operators since we can represent cosines but not sines of the highest wave number. Therefore the highest wave number is set to zero at each step.

Another singularity arises from the symmetries inherent to the water wave problem with periodic boundary conditions. In the two dimensional problem we recall that at a value c^* such that $\Delta(c^*, k) = 0, k \in \Gamma'$, the operator $A(c^*)$ has a two dimensional null space spanned by $\psi_1(c^*, k)$ and $\psi_2(c^*, k)$. We have decided to travel in the $\psi_1(c^*, k)$ direction which we implement by requiring that $\operatorname{Im}\{\hat{\eta}(k)\} = 0$ at every step. Similarly in the three dimensional problem at a parameter value $c^* = (c_1^*, c_2^*)$ such that $\Delta(c^*, \kappa_1) = \Delta(c^*, \kappa_2) = 0, \kappa_1, \kappa_2 \in \Gamma'$, the operator $A(c^*)$ has a four dimensional null space spanned by $\psi_1(c^*, \kappa_1)$, $\psi_2(c^*, \kappa_1)$, $\psi_1(c^*, \kappa_2)$, and $\psi_2(c^*, \kappa_2)$. We have decided to travel in the direction of the product of $\psi_1(c^*, \kappa_1)$ and $\psi_1(c^*, \kappa_2)$ which we implement by setting $\operatorname{Im}\{\hat{\eta}(\kappa_1)\}$ and $\operatorname{Im}\{\hat{\eta}(\kappa_2)\}$ to zero, and $\operatorname{Re}\{\hat{\eta}(\kappa_1)\}$ and $\operatorname{Re}\{\hat{\eta}(\kappa_2)\}$ equal to one another.

In practice implementing these procedures to avoid the singularities due to symmetry is straightforward. At the end of each iteration of Newton's method we simply set the component in question equal to zero or equal to another component. The only difficulty arises when considering the Jacobian matrix. Applying these procedures effectively projects out some of the dependent variables and so if we are to have a Jacobian of maximal rank we need to eliminate a corresponding number of equations. We project out equations which reduce the singularity of $A(c)$ at the bifurcation value c^* that we are near. After the projection we insert appropriate placeholders into the Jacobian matrix to maximize its rank. Such a procedure has worked well for us since we are only interested in solutions a short way up the branch.

3.4. *Filtering*

Two of the major sources of error in spectral methods for nonlinear differential equations are aliasing and spurious growth of high wave number errors. Effective methods to reduce aliasing error are computing de-aliased products, and filtering, while the effects of high wave number errors are usually controlled by filtering alone. Our efforts in reducing these types of errors have been focused almost exclusively on filtering and we have seen good results. We have implemented limited de-aliasing in evaluating the Dirichlet–Neumann operator where aliasing effects are most severe, but other products in the evaluation of F are aliased. More extensive de-aliasing techniques could be implemented, but we have chosen not to since aliasing errors decrease as the number of modes increases and our results have been satisfactory.

The first goal is to try to reduce the errors due to aliasing in evaluating the Dirichlet–Neumann operator. The jth term in the Dirichlet–Neumann operator involves nonlinearities of order $(j+1)$ so aliasing errors may accumulate quickly for large j. Complete de-aliasing of the Dirichlet–Neumann term of order j requires applying a perfect low-pass filter to both η and ξ of the form,

$$\Lambda(k, \nu) = \begin{cases} 1 & \text{for } |k| \leq \nu |k_{max}| \\ 0 & \text{for } |k| > \nu |k_{max}|, \end{cases} \tag{3.17}$$

with $\nu = \frac{1}{j+1}$.

Our second goal is to control the growth of errors in high wave numbers. It is well known that small errors are quickly amplified in spectral methods when differentiation operators are applied, and amplification is greatest in the highest modes. Filtering is the obvious method for overcoming such difficulties and we have used the ideal filter given in Eq. (3.17). However, applying the filter to η and ξ before each function and Jacobian evaluation resulted in solution paths which eventually diverged. As an alternative, we enforce the condition that wave numbers of modulus greater than $\nu = \frac{1}{2}$ be set to zero at every step. While intermediate calculations such as evaluation of F, $\partial_u F$, and $\partial_c F$ took place the higher wave numbers were allowed to fill which reduced aliasing errors, but at the end of the step the high wave numbers were set to zero. This was accomplished using the projection-placeholder zeroing out technique outlined above. This method does have the disadvantage that half of the modes are inactive; however, our method is spectral and our solutions are analytic so only a few modes are required to satisfy stringent conditions of convergence. Other non-ideal filters were also explored, such as the one used by Craig and Sulem [5], but the same divergence phenomenon was observed as with the application of the ideal filter. This is a phenomenon which needs to be investigated further, on a much simpler problem, and may be a numerical analogue of the techniques of alternate smoothing and Newton stepping found in Nash implicit function theorem techniques.

3.5. *Parallelization*

Although we have already stated that only a small number of modes are necessary to accurately resolve waveforms of interest, the problem is computationally challenging enough that it is worthwhile to program carefully and take advantage of any structure available. The challenge in the water wave problem arises due to the large number of total modes necessary to resolve the three dimensional problem, and the complicated nature of our equations.

A true Newton's method involves computing the Jacobian matrix at every iteration step and gives quadratic convergence, while a linear method would be to evaluate the Jacobian once at the beginning of the branch and then use this at all subsequent steps. Of course everything in between is possible and an appropriate choice of method is based on the accuracy required and the cost of evaluating and factoring the Jacobian matrix. For the water wave problem of small to moderate size the true bottleneck in the computation is the formation of the Jacobian matrix. We have decided to evaluate the Jacobian at the previous point along the branch, v_j, to determine the new tangent, $t(H'(v_j))$, and then evaluate once in the corrector phase using the Euler value, w_{j+1}. This does not result in a true Newton's method and it will not theoretically have super-geometric convergence rates, but nevertheless the convergence is very quick and the number of Jacobian evaluations is kept at two per step.

Since we have decided to evaluate the Jacobian at every step we ensure that the evaluation of the matrix is as fast as possible. Due to their recursive natures there are no direct formulae for either the Dirichlet–Neumann operator, $G(\eta)$, or its first variation, $\delta_n G(\eta)\xi$ as matrices. Therefore we are forced to form these matrices by evaluating them on each of the basis vectors and then placing the results in the appropriate columns. This is the most expensive operation in computing the Jacobian matrix and thus in the code for small to moderate problems. Not much can be done about this on a serial machine, but a simple way of speeding this process up on a parallel machine is to adopt a master-slave model [7] where the master runs most of the serial code on its own, and then parcels out this matrix evaluation work to the slave processes whenever $G(\eta)$ and $\delta_\eta G(\eta)\xi$ are needed. Clearly this is not the only way to parallelize this program, and for large numbers of modes it is obviously not the best especially as the cost of factoring the dense Jacobian matrix in serial becomes prohibitive. However, it is extremely easy to implement and debug and we have been able to run moderately sized problems, $N_1 = N_2 = 32$, on 32 nodes of the CCSTs SP at Argonne National Laboratory in reasonable amounts of time to get excellent results.

We have discussed two other ways to parallelize this code, each of which has its advantages. The first way would be to parallelize the evaluation of the Dirichlet–Neumann operator and its first variation. This would be based on the fact that when evaluating the jth term, we need to evaluate terms like $G_l(\eta)f_{j,l}(x)$ for $l = j, \ldots, 0$ where the $f_{j,l}$ are different for each value of j. We would again employ the master-slave model and invoke the help of the slaves only when evaluations of the Dirichlet–Neumann operator and its first variation are needed. Each one of the slaves would "specialize" in computing a certain order of G_l and would be passed appropriate $f_{j,l}$ for computation. This places a large burden on the slaves computing lower modes but it does parallelize the evaluation. Like the procedure we implemented this one will also suffer at higher number of modes due to the serial matrix factorization. The other procedure we considered would be to distribute the data over all of the nodes and then proceed as the serial code does, but with all FFTs, factorizations, etc., done in parallel. This has the obvious advantage that it adapts well to larger problems since it can store more data in memory and a larger portion of the code is in parallel (in particular the costly matrix factorization). However, this style of parallel programming is much more difficult and error-prone, and since our matrices are dense we cannot appeal to the iterative solution methods which have made large problems on parallel machines feasible. We plan to investigate both methods in the future.

4. NUMERICAL RESULTS

We have implemented the above procedure in both a serial and parallel version. As we have mentioned, a serial version is all that is necessary for even large two dimensional runs; however, a parallel version is required for moderately sized three dimensional runs. We have completed many runs for both problems and we briefly outline our progress below.

4.1. *The Two Dimensional Problem*

Much work has been done on the numerical solutions of the two dimensional water wave problem. There are many equations which approximate the two dimensional water wave equations which have all been studied in great detail. Craig and Sulem [5] have investigated the full time dependent water wave problem using our method and this is the precursor to our current work. Many people have investigated traveling wave solutions to the full equations and computed waves near the Stokes wave singularity (highest wave having a 120° angle at the crest) [2, 12, 19–21] (see also the summary by Miles [14]), breaking waves [6], and waves with various symmetry properties [23]. We do not attempt to compete with such computations since our formulation is not appropriate in all of these situations, but we present some of the our results concerning waves near the singularity. In Fig. 1 we see how the norm of our solution varies as a function of the parameter c along the non-trivial branch. In Fig. 2 we see the fastest-moving and most singular wave that we have computed thus far. In each case the results are from a computation with $N = 512$ collocation points and $m = 5$ Dirichlet–Neumann terms. The full run took roughly 7.4 h on 8 nodes of the Argonne SP.

Aside from traveling waves near the singularity we also would like to investigate secondary bifurcations from this primary branch of non-trivial solutions. A goal of such computations would be to find traveling waves with symmetry properties not yet seen, or a combination of various symmetry properties with the wave approaching a singularity. Both investigations will require more intensive computations, but they should be feasible given our parallel implementation.

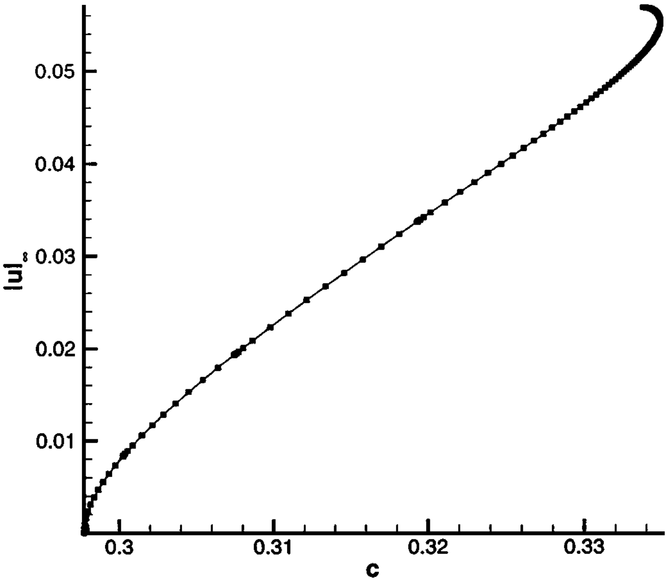

FIG. 1. Plot of $|u(x)|_\infty$ vs c $(h = \frac{1}{10})$.

4.2. *The Three Dimensional Problem*

In contrast to the two dimensional problem, very little work has been done on the three dimensional problem. The time dependent problem has been analyzed by Schanz [16], while traveling wave solutions to simplified versions of the three dimensional water wave equations have been studied by Hammack *et al.* [8, 9] and Milewski and Keller [15]. Hammack *et al.* constructed a wave tank where they were able to study two groups of

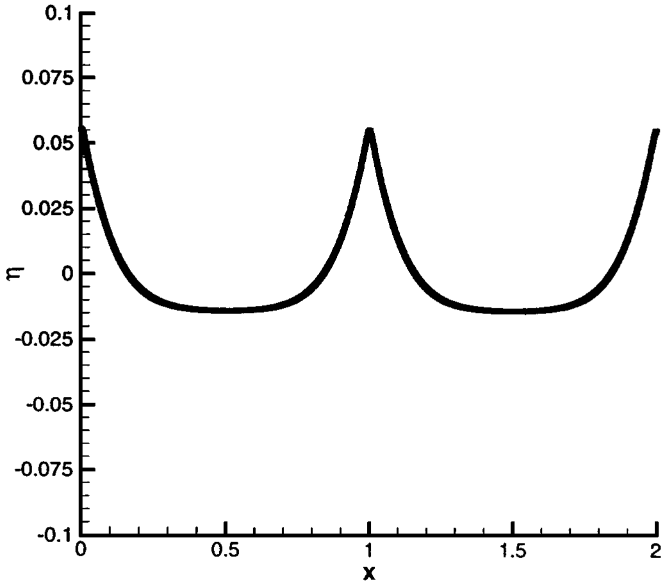

FIG. 2. Plot of $\eta(x)$ vs x $(h = \frac{1}{10})$.

solitary waves interacting where the angle between the wave train velocities was small. They produced pictures where hexagons were formed by the wave crests and noted that such behavior is exhibited by ocean waves in shallow water. They reproduced such solutions using the KP equation which models this behavior well. Milewski and Keller reproduced these and other solutions using their own formulation, and one of our goals in this paper is to reproduce these hexagonal solutions using the full water wave equations.

We have had great success in producing hexagonal solutions while traveling only a short distance up the non-trivial branch. The key seems to lie in choosing the fluid depth, the underlying parallelogram of periodicity, and the wave speed carefully. Since we allow the depth of the water to vary we can specify a general periodic problem using the matrix

$$A = \begin{pmatrix} 1 & R\cos\Theta \\ 0 & R\sin\Theta \end{pmatrix}, \tag{4.1}$$

to generate a lattice Γ. Choosing $\kappa_1, \kappa_2 \in \Gamma'$ one can find a parameter value $c^* = (c_1^*, c_2^*)$ such that $A(c^*)$ has a four dimensional null space. From this point c^* we move in a line with angle Ψ in the (c_1, c_2) plane. In all of our runs thus far we have chosen wave numbers corresponding to the points $j_1 = (1, 1)$, $j_2 = (1, -1)$ in the lattice, i.e.,

$$\kappa_1 = 2\pi(A^T)^{-1} j_1 \tag{4.2a}$$

$$\kappa_2 = 2\pi(A^T)^{-1} j_2. \tag{4.2b}$$

Our experience has shown that choosing shallow water on a periodic rectangle of high aspect ratio is effective in producing hexagonal wave patterns. Therefore we have selected $h = \frac{1}{100}$, $R = 11$, and $\Theta = \frac{\pi}{2}$ for the runs presented in this paper. For these values of R and Θ, c^* lies on the c_1 axis. We choose to move in a positive c_1 direction, i.e., $\mathbf{\Psi} = 0$, to amplify the hexagonal wave forming effect. In Fig. 3 we show the shape of the traveling wave moving with speed $c = (0.100512, 0.0)$. In Fig. 4 we show a contour plot of the same wave. It is easy to see in this figure that the wave crests are forming a hexagonal shape.

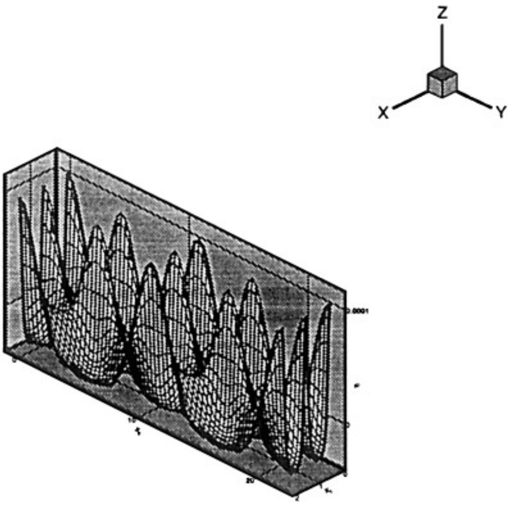

FIG. 3. Plot of $\eta(x)$ vs x ($h = \frac{1}{100}$, $R = 11$, $\Theta = \frac{\pi}{2}$).

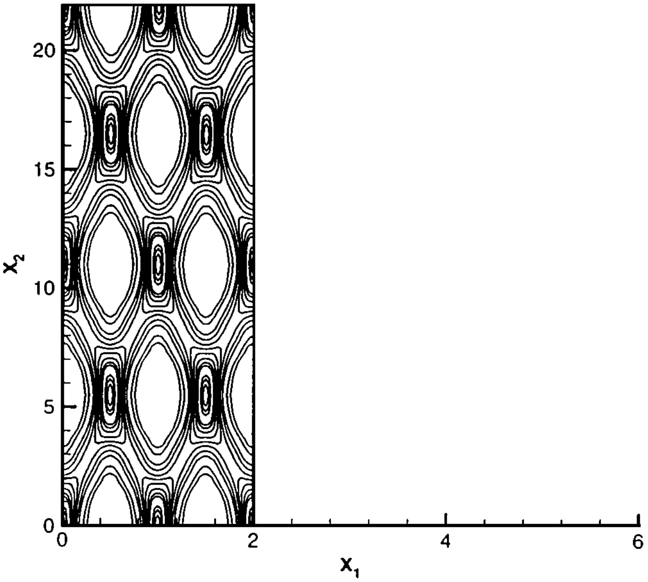

FIG. 4. Contour plot of $\eta(x)$ vs x ($h = \frac{1}{100}$, $R = 11$, $\Theta = \frac{\pi}{2}$).

One further direction for our three dimensional numerics is to try to reproduce some of the other wave forms seen by Milewski and Keller [15]. We would also like to reproduce their results concerning resonant interactions between primary waves and their subharmonics.

ACKNOWLEDGMENTS

The author thanks his advisor, Walter Craig, for suggesting this problem and providing encouragement and useful comments. He also thanks Paul Fischer and Catherine Sulem for much help along the way. He thanks the National Science Foundation (Graduate Fellowship) and the Division of Applied Mathematics at Brown University for supplying funding during this work. Finally, he thanks his wife Kristy for her undying love and support.

REFERENCES

1. E. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction* (Springer-Verlag, New York, 1990).
2. J. Byatt-Smith, An exact integral equation for steady surface waves, *Proc. R. Soc. London A* **315**, 625 (1970).
3. C. Canuto, Y. Hussani, A. Quarteroni, and T. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).
4. R. Coifman and Y. Meyer, Nonlinear harmonic analysis and analytic dependence, in *Proc. Sympos. Pure Math.* (Amer. Math. Soc., Providence, RI, 1985), Vol. 43, p. 71.
5. W. Craig and C. Sulem, Numerical simulation of gravity waves, *J. Comput. Phys.* **108**, 73 (1993).
6. D. Dommermuth and D. Yue, A high-order spectral method for the study of nonlinear gravity waves, *J. Fluid Mech.* **184**, 267 (1987).
7. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Programming with the Message Passing Interface* (MIT Press, Cambridge, MA, 1996).
8. J. Hammack, D. McCallister, N. Scheffner, and H. Segur, Two-dimensional periodic waves in shallow water. Part 2 Asymmetric waves, *J. Fluid Mech.* **285**, 95 (1995).
9. J. Hammack, N. Scheffner, and H. Segur, Two-dimensional periodic waves in shallow water, *J. Fluid Mech.* **209**, 567 (1989).

10. B. Kadomtsev and V. Petviashvili, On the stability of solitary waves in weakly dispersing media, *Sov. Phys. Dokl*. **15**, 539 (1970).
11. H. Keller, *Lectures on Numerical Methods in Bifurcation Problems* (Springer-Verlag, New York, 1987).
12. C. Lenau, The solitary wave of maximum amplitude, *J. Fluid Mech.* **26**, 309 (1966).
13. T. Levi-Civita, Détermination Rigoureuse des Ondes Permanentes d'Ampleur Finie, *Math. Ann.* **93**, 264 (1925).
14. J. Miles, Solitary waves, *Ann. Rev. Fluid Mech.* **13**, 11 (1980).
15. P. Milewski and J. Keller, Three-dimensional water waves, *Stud. Appl. Math.* **37**, 149 (1996).
16. U. Schanz, Ph.D. Thesis, University of Toronto, 1994.
17. J. Stoker, *Water Waves: The Mathematical Theory and Applications* (Interscience, New York, 1957).
18. D. Struik, Détermination Rigoureuse des Ondes Irrotationnelles Périodiques dans un Canal à Profondeur Finie, *Math. Ann.* **95**, 595 (1926).
19. J. Williams, Limiting gravity waves in water of finite depth, *Philos. Trans. R. Soc. London A* **302**, 139 (1981).
20. J. Witting, On the highest and other solitary waves, *SIAM J. Appl. Math.* **28**, 700 (1975).
21. H. Yamada, On the highest solitary wave, *Rep. Res. Inst. Appl. Mech. Kyushu Univ.*, 55 (1957).
22. V. Zakharov, Stability of periodic waves of finite amplitude on the surface of a deep fluid, *J. Appl. Mech. Tech. Phys.* **9**, 190 (1968).
23. J. Zufuria, Weakly nonlinear non-symmetric gravity waves on water of finite depth, *J. Fluid Mech.* **180**, 371 (1987).