UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Computing Periodic Gravity Waves on Water by using Moving Composite Overlapping Grids

N. Anders Petersson

January 1992

CAM Report 92-01

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Computing Periodic Gravity Waves on Water by using Moving Composite Overlapping Grids

N. Anders Petersson [1]
Department of Mathematics, UCLA
Los Angeles, CA 90024

January 15, 1992

[1] Supported by ONR grants N-00014-90-J-1695, N-00014-90-J-1382 and by the U.S. Department of Energy through Los Alamos National Laboratory.

Abstract

The composite overlapping grid method is applied to compute periodic gravity waves on water of finite constant depth. We make one component grid follow the free surface and let the remaining components be independent of the location of the surface. A pseudo-arclength continuation method is used to compute the solution as function of the phase velocity of the wave. The type of equation associated with some grid points and the number of equations in the discretized problem will change when the surface moves. We expound a stable way of switching composite grid during the continuation procedure which works close to limit points. An adaptive technique is also developed to efficiently resolve the solution where sharp gradients develop.

We present numerical examples that show very good agreement with existing results.

AMS Subject classifications: 65N50, 76B15.

Keywords: Adaptive grid, Composite overlapping grid, Pseudo-arclength continuation, Water wave.

1 Introduction

We consider two-dimensional irrotational gravity waves moving with constant phase velocity on water of finite constant depth. Both the infinitely deep case [2, 9] and the present case [4, 12] have previously been studied extensively. The aim of the research described here is to develop an accurate method that easily can be extended to compute the flow around an underwater obstacle, where we feel that the existing methods [8] still need improvement. This paper can therefore be seen as next step from [10] towards the solution of that problem. We prefer to first develop the method for the periodic case to be able to make comparisons with existing results.

We apply the composite overlapping grid method [3]. The basic idea is to make one component grid follow the free surface and let the remaining components be independent of the location of the surface. To resolve sharp gradients in the solution we also develop an adaptive technique. The adaptation is done locally on each component grid by changing the number of grid points and the stretching function. We would like to point out that the adaptive, moving grid approach developed here is general to problems where the shape of the domain is a function of the solution and/or the solution develops sharp gradients.

We use a pseudo-arclength continuation technique [7] to compute the solution as function of the phase velocity. When the surface moves, the type of equation associated with some grid points and the number of equations in the discretized problem change. The basic method therefore needs to be modified to allow for changes in the composite grid during the continuation procedure. A stable way of doing this which works close to limit points is devised.

The remainder of the paper is organized as follows. In §2 we scale the problem and present the governing equations. We discuss how to construct the composite overlapping grid in §3 and the problem is discretized in §4. Thereafter, in §5, we present the continuation method and in §6 we elaborate on the parts of the Jacobian matrix which are special for a moving composite grid. A stable way of switching composite grid is presented in § 7. Here, we also describe how the resolution of the grid is adapted to the solution. In §8 we discuss some implementational issues and we make numerical comparisons with existing results in §9. Very good agreement is found.

2 The governing equation

We describe the motion in Cartesian coordinates in a frame of reference fixed with respect to the wave, where the x-axis points opposite to the forward velocity and the z-axis is directed vertically upwards. We assume the motion to be steady in this coordinate system. Let the phase velocity be U and the wavelength be λ, which is defined as the shortest period of the wave. We scale the physical quantities by the length $\lambda/2\pi$ and the velocity $\sqrt{g\lambda/2\pi}$, where

g is the acceleration of gravity. In the scaled variables, $z = -d$ corresponds to the bottom and $z = 0$ to the undisturbed free surface. We split the total velocity potential into a free stream potential plus a perturbation potential: $\Phi(x,z) = \mu x + \phi(x,z)$, where $\mu = U/\sqrt{g\lambda/2\pi}$ is the scaled phase velocity. The perturbation potential is governed by, cf. Whitham [13],

$$\Delta\phi = 0,\ -d \le z \le \eta(x),\ -\infty < x < \infty, \tag{1}$$

where $\eta(x)$ is the elevation of the free surface. The perturbation potential is subject to the boundary condition

$$\phi_z(x,-d) = 0,\ -\infty < x < \infty. \tag{2}$$

Furthermore, at the free surface we require both the Bernoulli equation and the kinematic condition to be satisfied,

$$\mu\phi_x + \frac{1}{2}(\phi_x^2 + \phi_z^2) + \eta = 0,\ -\infty < x < \infty,\ z = \eta(x), \tag{3}$$

$$(\mu + \phi_x)\eta_x - \phi_z = 0,\ -\infty < x < \infty,\ z = \eta(x). \tag{4}$$

The choice of constant in the right hand side of Eq.(3) fixes the origin in z to the level of the free surface where the velocity, $\sqrt{(\mu+\phi_x)^2+\phi_z^2}$, equals the phase velocity, μ.

The problem is to find the perturbation potential, ϕ, and the surface elevation, η, as functions of the phase velocity, μ. We are interested in solutions that are l-periodic in the x-direction, with $l = 2\pi k$, $k = 1, 2, \ldots$. The surface elevation will be studied in the interval $0 \le x \le l$ and the perturbation potential in the domain $(x,z) \in \Omega$, $\Omega:\ 0 \le x \le l,\ -d \le z \le \eta(x)$. A convenient measure of a solution is its wave height, which is defined as the vertical distance between the highest crest and the deepest through of the wave.

A solution of Eqs.(1–4) at a fixed μ is not isolated. This is due to the Galilean invariance in the x-direction and the fact that ϕ is only determined up to a constant. Denote a solution (ϕ,η) by ψ and let $\mathcal{L}[\psi,\mu]$ be the operator described by Eqs.(1–4). Expanding $\mathcal{L}$ around a solution $\psi^{(0)}$ yields

$$\mathcal{L}[\psi^{(0)} + \psi', \mu] = \mathcal{L}[\psi^{(0)}, \mu] + \mathcal{L}_\psi[\psi^{(0)}, \mu]\psi' + \mathcal{O}(|\psi'|^2). \tag{5}$$

The two degrees of freedom correspond to two zero eigenvalues of $\mathcal{L}_\psi$. It is easy to see that the eigenfunction connected to the undetermined constant is $\phi' = const.$, $\eta' = 0$. The eigenfunction corresponding to the Galilean invariance is the x-derivative of the present solution, i.e. $\phi' = \phi_x^{(0)}$, $\eta' = \eta_x^{(0)}$.

3 The composite overlapping grid

The composite overlapping grid method is a general tool for solving PDEs on complex domains, cf. [3]. The basic idea is to divide the complex domain into

simple overlapping sub-domains, the union of which completely covers the region of interest. Each sub-domain is covered by a structured component grid and the set of component grids taken together is called the composite grid.

There are three different kinds of grid points in the composite grid: discretization, interpolation and unused grid points. The discrete version of the differential equation or the boundary condition operator is applied at a discretization point. The interpolation points are situated at an interior boundary where the sub-domains overlap. Here, the solution is interpolated from an overlapping component grid. The unused points are, as the name indicates, not used in the discretization of the differential equation. Henceforth, the discretization plus the interpolation points will be called the grid points used in the composite grid.

From previous investigations of the present problem it is known that the spatial scales of the solution can be very small close to the surface. The scales are also known to grow rapidly with the distance from the surface. To resolve the solution without wasting grid points, we will use a coarse component grid close to the bottom, a fine component grid close to the surface, and component grids of intermediate grid sizes in between.

The component grid that includes the surface will be called the surface grid. To make the discretization of the boundary conditions Eqs.(3,4) straightforward and accurate, we locate the uppermost grid line of the surface grid at the position of the surface, $\eta(x)$. In order to keep the dependence on η close to the surface and simplify the discretization, we choose the remaining component grids to be Cartesian and independent of η.

Let there be Q component grids and denote by Ω_q the sub-domain in the (x,z)-plane that corresponds to grid number q. The component grids are numbered beginning at the bottom and increasing towards the surface, see Fig. 1. The grid number equals the priority of that component in the composite grid. The composite grid will primarily use grid points from the component with the highest priority, secondarily from the grid with the second highest priority, and so on.

3.1 The mapping functions

For each sub-domain, we will design a one-to-one mapping function that transforms the unit square in the (r,s)-plane onto Ω_q. These mapping functions are used to construct the component grids by transforming the grid points in a Cartesian grid in the (r,s)-plane onto the (x,z)-plane. In the following, the (r,s)-plane will also be called the parameter space.

For the Cartesian component grids, the mapping functions are simply

$$x^q(r) = lr, \tag{6}$$

$$z^q(r,s) = a_q(1-s) + b_q s, \tag{7}$$

where a_q and b_q are constants; q is the grid number, $1 \leq q \leq Q-1$.

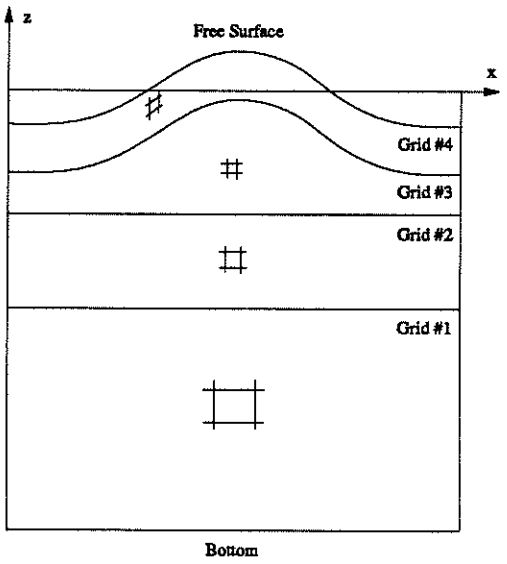

Figure 1: The numbering of the component grids. In this case, $Q = 4$.

To make the dependence of η in the transformed Laplace equation reasonably simple we choose the mapping function for the surface grid to be

$$x^Q(r) = lt_Q(r), \tag{8}$$

$$z^Q(r,s) = \eta(lt_Q(r)) - b_Q(1 - u_Q(s)). \tag{9}$$

Here, b_Q is the constant vertical thickness of the grid. The function $t_Q(r)$ clusters grid points in a layer around the crest of the wave and the function $u_Q(s)$ is used to concentrate grid points in a layer close to the surface. For example, $t_Q(r) = R^{-1}(r)$ where $R(t) = (t + U_p(t) - U_p(0))/(1 + U_p(1) - U_p(0))$ and

$$U_p(t) = \sum_{\nu=-\infty}^{\infty} U(t+\nu),\ U(t) = \frac{\alpha_r}{2}\tanh\beta_r(t-\gamma_r). \tag{10}$$

The stretching is concentrated around $t = \gamma_r$ and the parameter β_r determines the ratio between the smallest and the largest grid size. The constant α_r governs the width of the layer, i.e. the number of grid points that will be in the layer compared to the number outside it. The derivatives of the function $u_Q(s)$ are not required to be periodic. Therefore, the function U replaces U_p in the corresponding expression.

The grid points of grid q are given by $x_j^q = x^q(r_j^q)$ and $z_{j,k}^q = z^q(r_j^q, s_k^q)$,

where

$$r_j^q = (j-1)h_r^q, j = 1, 2, \ldots, N_q, \tag{11}$$
$$s_k^q = (k-1)h_s^q, k = 1, 2, \ldots, M_q, \tag{12}$$

with $h_r^q = 1/(N_q - 1)$ and $h_s^q = 1/(M_q - 1)$; $N_q > 1$ and $M_q > 1$ are natural numbers.

4 Discretizing the equations

In this section, we will describe how the Laplace equation and the boundary conditions are discretized by a second order accurate scheme on a composite grid. We adopt the mapping method which uses the previously constructed mapping functions to form the discrete set of equations on each component grid. The component grid problems are coupled to each other by interpolation relations at the interior boundaries where the sub-domains overlap.

We begin with some definitions. We define a component grid function on grid q by $g_{j,k}^q = g(x_j^q, z_{j,k}^q)$ and a surface grid function on grid Q by $f_j = f(x_j^Q)$. The forward, backward, and central divided difference operators in the r-direction on grid q are defined as

$$\begin{aligned} D_{+j} g_{j,k}^q &= \frac{g_{j+1,k}^q - g_{j,k}^q}{h_r^q}, \\ D_{-j} g_{j,k}^q &= D_{+j} g_{j-1,k}^q, \\ D_{0j} g_{j,k}^q &= \frac{1}{2}(D_{+j} + D_{-j}) g_{j,k}^q. \end{aligned}$$

A corresponding notation is used in the s-direction.

On the Cartesian grids, we discretize Eq.(1) by

$$\left(l^{-2} D_{+j} D_{-j} + (b_q - a_q)^{-2} D_{+k} D_{-k}\right) \phi_{j,k}^q = 0. \tag{13}$$

This stencil is applied to the discretization points with indices in the range $1 \le j \le N_q - 1$, $2 \le k \le M_q - 1$. The periodicity in the x-direction implies periodicity in the r-direction for all component grid functions. It is enforced by $\phi_{0,k}^q = \phi_{N_q-1,k}^q$ and $\phi_{N_q,k}^q = \phi_{1,k}^q$. We discretize the boundary condition at the bottom, Eq.(2), by the second order accurate one-sided formula

$$(b_1 - a_1)^{-1}(D_{+k} - \frac{h_s^1}{2} D_{+k} D_{+k})\phi_{j,k}^1 = 0, \tag{14}$$

for $k = 1$ and $1 \le j \le N_1 - 1$.

To discretize the Laplace equation on the surface grid, we transform it to the parameter space and replace the derivatives of ϕ in the resulting expression

by second order accurate divided differences. This yields

$$(\mathcal{A}_{j,k}D_{+j}D_{-j} + \mathcal{B}_{j,k}D_{0j}D_{0k} + \mathcal{C}_{j,k}D_{+k}D_{-k} + \mathcal{D}_{j,k}D_{0j} + \mathcal{E}_{j,k}D_{0k})\,\phi^Q_{j,k} = 0, \quad (15)$$

for the discretization points in $1 \le j \le N_Q - 1$, $2 \le k \le M_Q - 1$. The coefficients are, cf. [11],

$$\mathcal{A} = (r^Q_x)^2 + (r^Q_z)^2,\ \mathcal{B} = 2(r^Q_x s^Q_x + r^Q_z s^Q_z),\ \mathcal{C} = (s^Q_x)^2 + (s^Q_z)^2, \quad (16)$$

and

$$\mathcal{D} = r^Q_{xx} + r^Q_{zz},\ \mathcal{E} = s^Q_{xx} + s^Q_{zz}. \quad (17)$$

Here, r^Q and s^Q are the inverses of the mapping functions x^Q and z^Q, respectively, evaluated at the grid point in question. The subscripts on the metric quantities denotes partial differentiation, i.e. $r^Q_x = \partial r^Q/\partial x$ etc.

The boundary conditions at the free surface are discretized by the same technique as the Laplace equation. This yields

$$\mu(r^Q_x D_{0j} + s^Q_x D_{1k})\phi^Q_{j,k} + \frac{1}{2}\left((r^Q_x D_{0j} + s^Q_x D_{1k})\phi^Q_{j,k}\right)^2 + \frac{1}{2}\left((r^Q_z D_{0j} + s^Q_z D_{1k})\phi^Q_{j,k}\right)^2 + \eta_j = 0, \quad (18)$$

$$\left(\mu + (r^Q_x D_{0j} + s^Q_x D_{1k})\phi^Q_{j,k}\right) r^Q_x D_{0j}\eta_j - \left(r^Q_z D_{0j} + s^Q_z D_{1k}\right)\phi^Q_{j,k} = 0. \quad (19)$$

In these formulae we used $D_{1k} = D_{-k} + h_s D_{-k}D_{-k}/2$, which is a second order accurate approximation of the s-derivative. We apply Eqs.(18,19) to $k = M_Q$ and $1 \le j \le N_Q - 1$.

The component grid functions are coupled by interpolation relations. Our difference scheme is a second order accurate approximation of a second order equation and the component grids have an overlap which is proportional to the grid size. To get second order accuracy for the total solution, it is necessary to use (at least) third-order accurate interpolation, cf. [3]. For example, let $\phi^a_{n,m}$ be an interpolation point with interpolation location (j,k,b). This means that $\phi^a_{n,m}$ will be interpolated from $\phi^b_{j+p,k+q}$, $-1 \le p,q \le 1$. Let $(\tilde{r},\tilde{s})$ satisfy $x^a_n = x^b(\tilde{r})$ and $z^a_{n,m} = z^b(\tilde{r},\tilde{s})$. The biquadratic interpolation yields

$$\phi^a_{n,m} - \sum_{p=-1}^{1}\sum_{q=-1}^{1} \alpha_p(\tilde{r})\beta_q(\tilde{s})\phi^b_{j+p,k+q} = 0, \quad (20)$$

where α_p and β_q are quadratic polynomial base-functions.

5 Continuation in μ

We will use a pseudo-arclength continuation method to calculate (ϕ, η) as function of the phase velocity μ. The basic method described in [7] can not be used directly due to the two degrees of freedom that make the linearized operator singular. The method also needs to be modified because the number of used grid points in the composite grid depends on the surface elevation. However, the discussion of that aspect will be postponed to § 7.

To handle the two degrees of freedom, one can add two new degrees of freedom and two extra equations such that the Jacobian of the extended problem becomes non-singular [6]. However, it is possible to avoid adding extra unknowns and equations by instead enforcing the increments in the Newton iteration to be orthogonal to the nullspace connected to the two zero eigenvalues. This is equivalent to solving Eqs.(1–4) in the subspace orthogonal to the nullspace. The part of the solution in the nullspace is of no interest, since it only corresponds to adding a constant to the perturbation potential and shifting the solution in the x-direction.

To enforce the orthogonality of the increments in the Newton iteration, we also need the nullspace of the adjoint operator. Unfortunately, it is difficult to numerically compute accurate approximations of these nullspaces. To overcome this difficulty, we modify the boundary condition at the bottom to fix the constant part of the solution. Instead of Eq.(2), we use

$$\phi_z(x, -d) + \phi(0, -d) = 0, \ 0 \le x \le l. \tag{21}$$

The discrete boundary condition, Eq.(14), is modified in a corresponding way. The idea is that a solution with $\phi(0, -d) = 0$ satisfies the original boundary condition without introducing an undetermined constant. The linearized modified operator will therefore only have one zero eigenvalue. This makes an accurate computation of the corresponding nullspace much easier. Due to non-conservation, we cannot expect $\phi(0, -d)$ to be zero in the discrete approximation. Instead it will get a small value that indicates how well the continuous conservation property is satisfied. We regard this value as a measure of the accuracy of the discrete solution.

The discrete set of equations can be written in abstract form as $L[u, \mu] = 0$, where $L : \mathcal{X} \times \Re \to \mathcal{X}$. Here, $\mathcal{X}$ is an n-dimensional vectorspace and n is the number of grid points used in the composite grid plus the number of grid points along the surface. The vector u contains ϕ at every used grid point and η at the grid points along the surface. We define a scalar product and a norm for $x, y \in \mathcal{X}$ by

$$\langle x, y \rangle = \frac{1}{n} \sum_{i=1}^{n} x_i y_i, \ \|x\| = \sqrt{\langle x, x \rangle}. \tag{22}$$

The eigenvalue connected to the Galilean invariance will henceforth be called the shift-eigenvalue. Let e_r and e_l denote the right and left eigenvectors of

the Jacobian matrix, $\partial L/\partial u$, corresponding to the shift-eigenvalue. Let the eigenvectors be normalized to have $\|e_r\| = 1$ and $\langle e_l, e_r \rangle = 1$. The projection P maps $\mathcal{X}$ onto the eigenspace according to $Pf = \langle e_l, f \rangle e_r$, $f \in \mathcal{X}$.

We will consider solving

$$(I - P[u,\mu])L[u,\mu] = 0, \tag{23}$$

for the part of the solution in $(I-P)\mathcal{X}$. The part in $P\mathcal{X}$ corresponds to the phase of the solution and it will be determined during the solution procedure. A point (u,μ) that satisfies $(I - P[u,\mu])L[u,\mu] = 0$ will be called a solution point and we will call a point non-singular if all eigenvalues of $\partial L/\partial u$ except the shift-eigenvalue are bounded away from zero.

5.1 The non-singular case

Following [7], the solution (u,μ) will be considered as a function of the pseudo-arclength σ, $u = u(\sigma)$ and $\mu = \mu(\sigma)$. Assume that a non-singular solution point (u_0,μ_0) is known, and let it have pseudo-arclength σ_0. We define the pseudo-arclength relative to that point by

$$\sigma = \sigma_0 + \langle \dot{u}_0, u - u_0 \rangle + \dot{\mu}_0(\mu - \mu_0). \tag{24}$$

The tangent $(\dot{u}_0, \dot{\mu}_0)$ is the solution of

$$L_u[u_0,\mu_0]\dot{u}_0 = -L^I_\mu[u_0,\mu_0]\dot{\mu}_0. \tag{25}$$

Here we used the notation $L^I_\mu[u_0,\mu_0] = (I - P[u_0,\mu_0])L_\mu[u_0,\mu_0]$. Eq.(25) is solved under the side-condition $P\dot{u}_0 = 0$ and the normalization $\|\dot{u}_0\|^2 + |\dot{\mu}_0|^2 = 1$. We also require the scalar product between the previous and the present tangent to be positive.

To get the same number of equations as dependent variables, we augment $(I - P[u,\mu])L[u,\mu] = 0$ by the arclength equation $N[u,\mu;\sigma] = 0$, where

$$N[u,\mu;\sigma] = \langle \dot{u}_0, u - u_0 \rangle + \dot{\mu}_0(\mu - \mu_0) - (\sigma - \sigma_0). \tag{26}$$

We use the predictor $(u^0,\mu^0) = (u_0 + \dot{u}_0\Delta\sigma, \mu_0 + \dot{\mu}_0\Delta\sigma)$ as initial guess for the solution at $\sigma = \sigma_0 + \Delta\sigma$. The predictor is corrected by Newton's method on the augmented system, where the improvements of the solution are found by solving

$$\begin{pmatrix} L_u[u^k,\mu^k] & L^I_\mu[u^k,\mu^k] \\ N_u[u^k,\mu^k] & N_\mu[u^k,\mu^k] \end{pmatrix} \begin{pmatrix} \Delta u^k \\ \Delta\mu^k \end{pmatrix} = - \begin{pmatrix} L^I[u^k,\mu^k] \\ N[u^k,\mu^k] \end{pmatrix} \tag{27}$$

by the bordering algorithm, under the side-condition $P[u^k,\mu^k]\Delta u^k = 0$. The solution is then updated according to

$$u^{k+1} = u^k + \Delta u^k, \tag{28}$$

$$\mu^{k+1} = \mu^k + \Delta\mu^k. \tag{29}$$

We iterate until $\|u^{k+1} - u^k\| + |\mu^{k+1} - \mu^k| < \epsilon$.

If the iteration converges, we may repeat the procedure after the composite grid has been updated. The number of iterations that was required to get convergence is used to determine next step-size $\Delta\sigma$. However, if the iteration diverges, we halve the step-size and try again.

5.2 The singular case

The smallest eigenvalue, except for the shift-eigenvalue, is monitored along the solution curve to detect singular points. The pseudo-arclength method easily passes through limit points, but special care is necessary to switch branch at a bifurcation point. One can proceed as in [7]. In that method, the tangent to the solution curve on the other branch is computed by using second derivative information of L. The continuation then proceeds in the direction of the new tangent starting at the singular point. In the present work, we have instead used a Lyapunov-Schmidt reduction technique that does not use the second derivatives of L.

We only consider the situation when $\partial L/\partial u$ has exactly two small eigenvalues, of which one is the shift-eigenvalue. We further assume the eigenvectors of the two small eigenvalues to be linearly independent. Our task is to find one point on every solution curve (u, μ) that satisfy $(I - P[u, \mu])L[u, \mu] = 0$, close to the singular point.

Let e'_r and e'_l be the right and left eigenvectors of the second small eigenvalue, normalized to have $\|e'_r\| = 1$ and $\langle e'_l, e'_r \rangle = 1$. Let P' be the projection that maps $\mathcal{X}$ onto the second eigenspace. It is defined by $P'f = \langle e'_l, f \rangle e'_r$, $f \in \mathcal{X}$.

Let (u_0, μ_0) be a solution point close to the singular point and set $u = u_0 + x$ and $\mu = \mu_0 + \tau$. We split x into three parts, $x = x^I + x^{II} + x^{III}$, where $x^I = (I - P - P')x$, $x^{II} = P'x$, and $x^{III} = Px$. The part of x in P' can also be expressed as $x^{II} = \alpha e'_r$, $\alpha = \langle e'_l, x \rangle$, and the part in P will be set to zero, $x^{III} = 0$, since it only shifts the solution. We now split the problem according to

$$(I - P')(I - P)L[u, \mu] = 0, \tag{30}$$

$$P'(I - P)L[u, \mu] = 0. \tag{31}$$

Let $A = \partial L/\partial u[u_0, \mu_0]$ denote the Jacobian matrix. We can solve the linear system $Ax = b$ for $b \in (I - P - P')\mathcal{X}$ uniquely in $(I - P - P')\mathcal{X}$, where P and P' are evaluated at (u_0, μ_0). Therefore, Eq.(30) can be used to compute $x^I = x^I(\alpha, \tau)$ by iteration. First we choose some values of α and τ that will be fixed throughout the iteration. We take the initial guess to be $x^I_0 = 0$ and compute the subsequent iterates by solving

$$L_u[u_0, \mu_0]\Delta x_k = -(I - P - P')L[u_0 + x^I_k + \alpha e'_r, \mu_0 + \tau], \tag{32}$$

$$x^I_{k+1} = x^I_k + (I - P - P')\Delta x_k. \tag{33}$$

We truncate the iteration when $\|x^I_{k+1} - x^I_k\| < \epsilon$. The iteration will converge for sufficiently small α and τ because, by assumption, $\partial L/\partial u$ has only two small eigenvalues and the corresponding parts of the solution are kept constant by the projections. The iteration only requires L_u to be factored once; it is therefore relatively inexpensive compared to the Newton iteration.

The second relation, Eq.(31), is equivalent to the scalar equation $g(\alpha, \tau) = 0$, where

$$g(\alpha, \tau) = \langle e'_l, L[u_0 + x^I(\alpha, \tau) + \alpha e'_r, \mu_0 + \tau] \rangle. \tag{34}$$

This equation can for instance be solved by the following two step technique. We first approximately locate the zeros of g by evaluating it at a number of points on the small circle $\alpha^2 + \tau^2 = r^2$, $r \ll 1$. For each zero, we solve $g = 0$ by bisection with the approximate location as initial guess.

We can proceed with the non-singular continuation approach starting from a solution point sufficiently far away from the singularity in the (α, τ)-plane. We trace out all solution curves that connect at the singular point by doing this for one point on each branch.

6 The Jacobian matrix

The continuation method requires the knowledge of the Jacobian matrix $\partial L/\partial u$ and the derivative $\partial L/\partial \mu$. Only the boundary conditions at the surface, Eqs.(18, 19), depend explicitly on μ. Hence, $\partial L/\partial \mu$ will only be non-zero along the surface. The Jacobian matrix is most naturally split into the two parts $\partial L/\partial \phi$ and $\partial L/\partial \eta$. L depends non-linearly on ϕ only in Eqs.(18,19), so $\partial L/\partial \phi$ is very straight-forward to form. However, the part $\partial L/\partial \eta$ is somewhat involved to set up and that motivates a more detailed description. The dependence of η is located to the surface boundary conditions, the Laplace equation on the surface grid, and to the interpolation relations which involve the surface grid function.

The Laplace equation on the surface grid, Eqs.(15–17), and the boundary conditions at the surface, Eqs.(18,19), contain metric properties that are functions of the position of the surface. In order to find how they depend on η we express them in terms of the r and s-derivatives of the mapping function, Eqs.(8,9), by inverting its Jacobian. This yields,

$$r_x = \frac{1}{x_r}, \; s_x = -\frac{z_r}{x_r z_s}, \; r_z = 0, \; s_z = \frac{1}{z_s}. \tag{35}$$

The second derivatives are $r_{zz} = 0$, $s_{zz} = -z_{ss}/z_s^3$, $r_{xx} = -x_{rr}/x_r^3$ and

$$s_{xx} = \frac{x_{rr} z_r z_s^2 - z_{rr} x_r z_s^2 - z_{ss} x_r z_r^2}{z_s^3 x_r^3}. \tag{36}$$

By inspecting the mapping function for the surface grid, we see that z^Q_r and z^Q_{rr} depend on η' and η''. We approximate the derivatives of η with second order

accurate divided differences, $z_r^Q(r_j, s_k) \approx D_{0j}\eta_j$ and $z_{rr}^Q(r_j, s_k) \approx D_{+j}D_{-j}\eta_j$. This makes the corresponding parts of the Jacobian matrix sparse, since the dependence of η will be local to the grid point in question.

The interpolation relations which involve grid points in the surface grid depend on η through $\tilde{s}$ in Eq.(20). The first case occurs for the interpolation points in the surface grid. The surface grid is always overlapped by grid $Q-1$. Let Eq.(20) applied to the interpolation point (n,m) in grid Q be denoted $L_{n,m}^Q = 0$, where

$$L_{n,m}^Q =: \phi_{n,m}^Q - \sum_{p=-1}^{1}\sum_{q=-1}^{1} \alpha_p(\tilde{r})\beta_q(\tilde{s}(\eta_n))\phi_{j+p,k+q}^{Q-1}. \tag{37}$$

The mapping function for grid $Q-1$, Eq.(7), yields $d\tilde{s}/d\eta_n = (b_{Q-1} - a_{Q-1})^{-1}$ whence

$$\frac{\partial L_{n,m}^Q}{\partial \eta_n} = -\sum_{p=-1}^{1}\sum_{q=-1}^{1} \frac{\alpha_p(\tilde{r})\beta_q'(\tilde{s})}{b_{Q-1} - a_{Q-1}}\phi_{j+p,k+q}^{Q-1}. \tag{38}$$

The second case takes place at those interpolation points in grid $Q-1$ that are overlapped by the surface grid. Let Eq.(20) applied to the interpolation point (n,m) in grid $Q-1$ be $L_{n,m}^{Q-1} = 0$, where

$$L_{n,m}^{Q-1} =: \phi_{n,m}^{Q-1} - \sum_{p=-1}^{1}\sum_{q=-1}^{1} \alpha_p(\tilde{r})\beta_q(\tilde{s})\phi_{j+p,k+q}^{Q}. \tag{39}$$

In principle, $\tilde{s}$ depends globally on η, but we will only introduce third order errors in $(\tilde{r}, \tilde{s})$ if we locally approximate the mapping function for the surface grid, Eqs.(8,9), with the biquadratic mapping,

$$\tilde{x}^Q(\tilde{r}) = \sum_{p=-1}^{1} \alpha_p(\tilde{r})x_{j+p}^Q, \tag{40}$$

$$\tilde{z}^Q(\tilde{r}, \tilde{s}) = \sum_{p=-1}^{1}\sum_{q=-1}^{1} \alpha_p(\tilde{r})\beta_q(\tilde{s})z_{j+p,k+q}^Q. \tag{41}$$

The coordinates $(\tilde{r}, \tilde{s})$ in Eq.(39) are the solution of $\tilde{x}^Q(\tilde{r}) = x_n^{Q-1}$, $\tilde{z}^Q(\tilde{r}, \tilde{s}) = z_{n,m}^{Q-1}$ and the grid points in the surface grid have $z_{j,k}^Q = \eta_j - b_Q(1 - u_Q(s_k))$. Therefore, $\tilde{s}$ depends only locally on η, which implies a sparse Jacobian matrix. Differentiating Eq.(39) gives

$$\frac{\partial L_{n,m}^{Q-1}}{\partial \eta_\gamma} = -\sum_{p=-1}^{1}\sum_{q=-1}^{1} \left(\alpha_p'(\tilde{r})\beta_q(\tilde{s})\frac{\partial \tilde{r}}{\partial \eta_\gamma} + \alpha_p(\tilde{r})\beta_q'(\tilde{s})\frac{\partial \tilde{s}}{\partial \eta_\gamma}\right)\phi_{j+p,k+q}^Q. \tag{42}$$

In order to make $\tilde{x}^Q(\tilde{r},\tilde{s}) = x^{Q-1}_{n,m}$ and $\tilde{z}^Q(\tilde{r},\tilde{s}) = z^{Q-1}_{n,m}$ when η_γ is varied, we must have $d\tilde{x}^Q/d\eta_\gamma = 0$ and $d\tilde{z}^Q/d\eta_\gamma = 0$. We get $\partial\tilde{r}/\partial\eta_\gamma$ and $\partial\tilde{s}/\partial\eta_\gamma$ by differentiating Eqs.(40,41) and solving the linear system

$$\begin{pmatrix} \tilde{x}^Q_r & 0 \\ \tilde{z}^Q_r & \tilde{z}^Q_s \end{pmatrix} \begin{pmatrix} \partial\tilde{r}/\partial\eta_\gamma \\ \partial\tilde{s}/\partial\eta_\gamma \end{pmatrix} = - \begin{pmatrix} \partial\tilde{x}^Q/\partial\eta_\gamma \\ \partial\tilde{z}^Q/\partial\eta_\gamma \end{pmatrix}. \tag{43}$$

Now, $\partial\tilde{x}^Q/\partial\eta_\gamma = 0$ and $\partial\tilde{z}^Q/\partial\eta_\gamma \neq 0$ for $\gamma = j-1, j, j+1$, so $\partial\tilde{s}/\partial\eta_\gamma = -\partial\tilde{z}^Q/\partial\eta_\gamma/\tilde{z}^Q_s$ and $\partial\tilde{r}/\partial\eta_\gamma = 0$.

7 Changing composite grid

When the surface moves during the continuation procedure, the amount of overlap between the sub-domains Ω_Q and Ω_{Q-1} will change. Component grid Q has the highest priority so all its grid points will always be used in the composite grid. The situation is different for grid $Q-1$. After the surface has moved, some grid points that previously were used might no longer be needed, while some formerly unused grid points might be required in the composite grid. Therefore, the type of equation associated with a grid point and the number of used grid points in grid $Q-1$ will depend on the location of the surface. Furthermore, even if the type of equation remains unchanged for all grid points, the interpolation locations might be different for some interpolation points. For these reasons, the continuation method must be modified to allow for changes in the composite grid.

The basic idea is to fix the interpolation locations and the type of equation associated with each grid point during the Newton iteration and only update the mapping function for the surface grid. This means that for instance the discretization points are prevented from changing to interpolation points or unused points. For stability reasons, the amount the surface is allowed to move between two solution points is restricted by the amount of overlap, i.e. we do not allow for extrapolation in the interpolation relations. With the notation of Eq.(20), extrapolation occurs if at least one of the following inequalities is satisfied: $\tilde{r} < r^b_{j-1}$, $\tilde{s} < s^b_{k-1}$, $r^b_{j+1} < \tilde{r}$ or $s^b_{k+1} < \tilde{s}$. If extrapolation should occur in any interpolation relation, we stop the Newton iteration, decrease the step size $\Delta\sigma$ and restart the continuation from the previous solution point.

Once the iteration has converged to a solution where there is no extrapolation in the interpolation relations, we verify if the composite grid is consistent with the position of the surface. This is done by checking if the interpolations are sufficiently centered. The criterion

$$r^b_j - 0.6h^b_r \leq \tilde{r} \leq r^b_j + 0.6h^b_r,\ s^b_k - 0.6h^b_s \leq \tilde{s} \leq s^b_k + 0.6h^b_s, \tag{44}$$

has been found to work well in practise. We have a valid solution if all interpolation points satisfy Eq.(44); we may then proceed with the continuation. Otherwise, the composite grid and the solution must first be corrected.

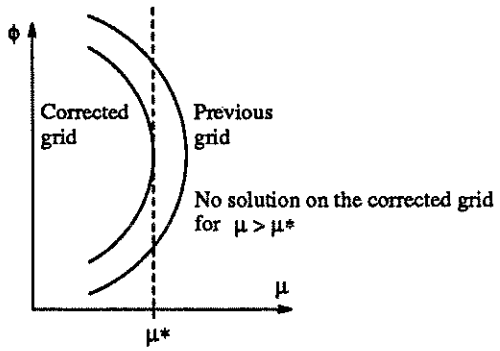

Figure 2: The location of a limit point might depend slightly on the grid.

7.1 The correction step

We begin by constructing a corrected composite grid. We mention in passing that this can be done at a much lower computational cost than would be involved in constructing a completely new grid. Thereafter, we repeat the last continuation step on the corrected grid. We get u_0 on the corrected grid by interpolation from the previous grid. With this approach, it is possible to correct the grid close to a limit point. The simpler idea to solve $(I - P[u,\mu])L[u,\mu] = 0$ on the corrected composite grid but fixing μ at the current value might fail close to a limit point because its location depends slightly on the grid, see Fig. 2. The correction step has been found to be very stable with the present restriction on the movement of the surface.

The reason for allowing the interpolation points to be slightly non-centered is to make sure that the corrected solution will satisfy Eq.(44) on the corrected grid. The difficulty that otherwise might arise is illustrated in Fig. 3. Consider the situation when 0.6 is replaced by 0.5 in Eq.(44). Let some interpolation point have $\tilde{s} = s_k^b + 0.5h_s^b + \epsilon$, $0 < \epsilon \ll 1$, before the correction step. The corrected composite grid will use more grid points in grid b to make $\tilde{s} = s_{k+1}^b - 0.5h_s^b + \epsilon$. The problem is that the solution on the corrected grid might be slightly different than the solution on the previous grid. In particular, the corrected location of the surface can make $\tilde{s} = s_{k+1}^b - 0.5h_s^b - \tilde{\epsilon}$, $0 < \tilde{\epsilon} \ll 1$, which would call for correcting the composite grid back to the initial state. This problem does not occur if a less restrictive tolerance is used. Instead, both composite grids in Fig. 3 would be valid. The history of the previous locations of the surface determines which of the possible composite grids that will be used. This is acceptable because we can expect the solutions on the different grids to be very similar if the solution is well resolved.

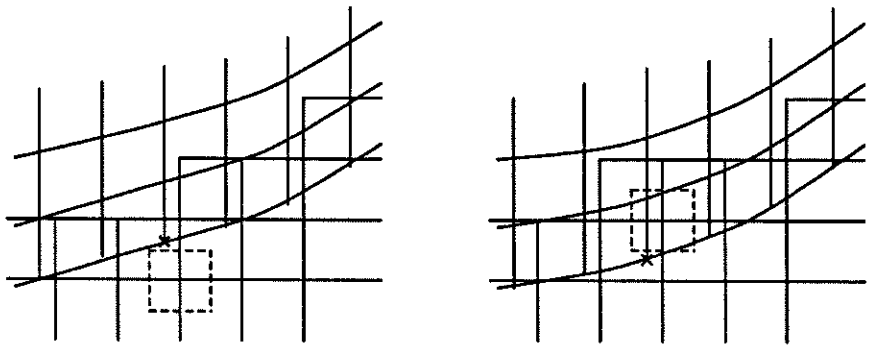

Figure 3: The oscillation between two very similar composite grids that can be avoided by allowing for slightly non-centered interpolation points.

7.2 Adapting the grid to the solution

The solution becomes steeper as the wave height increases. In particular, the derivative of the surface elevation tends to a discontinuity at the crest of the wave. A very fine grid is therefore necessary to resolve the solution close to that point, but it is difficult to a priori estimate how fine the grid needs to be for a certain wave height. It is also very uneconomical to use the fine grid all the way from the trivial solution. For these reasons, we have developed an adaptive technique where we allow the resolution in each component grid and the stretching in the surface grid to change during the continuation procedure.

Ideally, one would like to monitor the truncation error of the discrete solution, and increase the resolution when and where it is necessary. In the present work, we have used a simpler approach, namely to monitor how well the solution is resolved on the grid. In particular, for the perturbation potential we look at the largest difference in each direction, i.e.

$$\text{diff}_r \phi^q := \max |h_r^q D_{+j} \phi_{j,k}^q|, \ \text{diff}_s \phi^q := \max |h_s^q D_{+k} \phi_{j,k}^q|. \tag{45}$$

For the surface elevation, we are concerned with resolving the sharp gradient in η_x that develops when the wave height gets close to its limiting value, see Fig. 4. We therefore monitor $\text{diff}_r \eta_x$.

At each valid solution point, we calculate $\text{diff}_r \phi^q$, $\text{diff}_s \phi^q$ and $\text{diff}_r \eta_x$. The idea is to keep these quantities in the predetermined ranges

$$A_\phi < \text{diff}_r \phi^q, \text{diff}_s \phi^q < B_\phi \ \text{ and } \ A_\eta < \text{diff}_r \eta_x < B_\eta \tag{46}$$

by changing the number of grid points and the parameters in the stretching function. The constants A_ϕ, B_ϕ, A_η and B_η will be called the resolution thresholds.

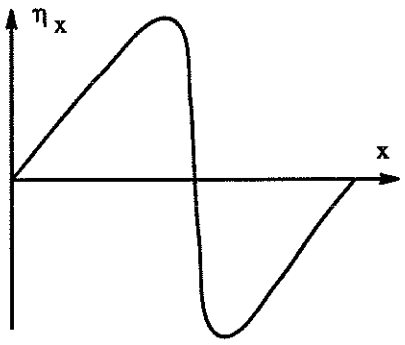

Figure 4: The x-derivative of the surface elevation close to the maximum wave height.

Initially, the solution is trivial, and we can use a coarse grid to start the continuation. During the continuation, some difference will eventually exceed its limit and we then need to construct a grid with better resolution. For example, let $\text{diff}_r \phi^q > B_\phi$. We increase the number of grid points in the r-direction of grid q by changing N_q such that the solution on the new grid approximately satisfies $\text{diff}_r \phi^q = A_\phi$.

We control the stretching by monitoring $\eta_x(x)$. To avoid wasting grid points, the strength β_r and the location γ_r of the stretching function Eq.(10) are chosen such that

$$\max h_r^q D_{+k} \eta_x(x_k^Q) \approx -\min h_r^q D_{+k} \eta_x(x_k^Q). \tag{47}$$

When the number of grid points and the stretching have been properly adjusted, we perform a correction step (§ 7.1) to update the grid and get a valid solution on the refined grid. To make the process stable and reliable, the solution must be reasonably well resolved on the grid before we try to change the resolution. If it is not, we decrease the step size $\Delta\sigma$ and compute an intermediate, more well resolved, solution before we change the resolution. Without this precaution, the correction step might diverge. Another important rule is to avoid changing the resolution too drastically. Practically, we found that the correction step converges quickly if the number of grid points changes with less than 20 percent in each direction on each component grid, and the stretching strength β_r changes with less than 5.0.

8 Implementation

In the present implementation we have used the software package CMPGRD, cf. [1], to generate the composite grids. A composite grid is created by calling

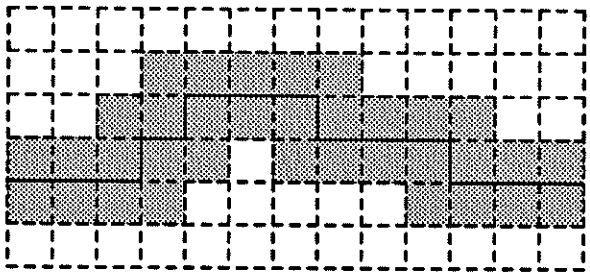

Figure 5: The band in the (r, s)-plane that is searched for the location of the interpolation points in the corrected grid. The solid line indicates the position of the interpolation points in the previous grid.

CMPGRD as a subroutine with an input consisting of the component grid-transformations, the number of grid points in each direction on each grid, and the priority between the component grids. On output, each grid point is labeled according to how it will be used and the interpolation locations are also given for the interpolation points.

When CMPGRD is called to correct an existing grid, the previous composite grid and the corrected mapping function for the surface grid are given as input. For the case when the resolution has been changed, the new number of grid points is also supplied. The fast algorithm for correcting the composite grid only works if the new interpolation points can be found at most one grid point away from some previous interpolation point in the same component of the previous composite grid, cf. Fig. 5. This restriction is consistent with the aforementioned requirement to disallow for extrapolation in the interpolation relations. However, we have encountered situations when the fast algorithm fails after the stretching has been changed drastically. In those rare cases, a new composite grid has to be constructed from scratch.

The information supplied by CMPGRD is sufficient to form the discrete set of equations for a composite grid that is composed of an arbitrary number of component grids, each having an arbitrary number of grid points. The only restriction on the component grids is that they must overlap sufficiently. However, the implementation of a general solver in FORTRAN-77 requires a dynamic memory allocator. For this purpose, we have used the DSK-package, which is an autonomous part of CMPGRD.

The emerging linear systems of equations were solved by the YALE sparse matrix package, cf. [5]. We found experimentally that the accuracy of the solution was improved when the equations corresponding to the boundary conditions at the surface were inserted early in the system matrix. Actually, this trick was necessary to make the Newton iteration converge properly for solutions with

A_ϕ	$\mu = 1.04247$	$\mu = 1.07029$	$\mu = 1.09184$
3.0×10^{-2}	0.092107	0.117823	0.135427
2.0×10^{-2}	0.092481	0.118119	0.136440
1.0×10^{-2}	0.092278	0.117885	0.136362
[9]	0.091809	0.117572	0.136178

Table I: h/l as function of the resolution for the depth $d = 2\pi$. The results obtained by [9] for infinite depth are given at the bottom of the table.

wave heights close to the limiting value. The reason for this behavior might be that this package only pivots with respect to the structure of the non-zero elements in the matrix. Our main motivation for using this package despite this deficiency is that it is fast.

9 Numerical results

We begin by studying a single wave, i.e. $l = 2\pi$, on deep water. We compare our results with those reported by [2] and [9] for the infinitely deep case. To investigate the effect of the finite depth in our computation, we performed two sets of calculations with $d = 2\pi$ and $d = 4\pi$, respectively. To also investigate the dependence of the grid size, we varied the resolution thresholds. For simplicity, we used $A_\phi = A_\eta$ and $B_\phi = B_\eta = A_\phi + 5 \times 10^{-3}$. We used a composite grid with 4 components and the mapping functions for the component grids had the parameters $a_1 = -d$, $b_1 = -2.5$, $a_2 = -3.0$, $b_2 = -0.7$, $a_3 = -1.2$, $b_3 = 0.65$ and $b_4 = 0.4$. In the case $l = d$, we used the following number of grid points in the start grid: $N_1 = 14$, $M_1 = 8$, $N_2 = 23$, $M_2 = 11$, $N_3 = 39$, $M_3 = 13$, $N_4 = 63$ and $M_4 = 7$. For the deeper case, we used $M_1 = 20$.

For the continuous problem, the bifurcation from the trivial solution occurs at $\mu_0 = \sqrt{\tanh d}$. With the present start grid, it was displaced by $\mathcal{O}(10^{-4})$.

Let h denote the wave height. In table I and II we present h/l as function of the resolution thresholds. The three different μ-values correspond to solutions where the maximum slope of the surface is approximately 17.0, 22.4 and 27.8 degrees, respectively. For $A_\phi = 10^{-2}$, the relative difference between our results and those reported by [9] are of the order $\mathcal{O}(10^{-3})$. We conclude that the discrepancies are mainly caused by truncation errors and not by the finite depth. Both the accuracy and the efficiency of the present method would probably be improved by using a higher order difference scheme.

The value of $\phi(0, -d)$, which occurs in Eq.(21), indicates how well the conservation property of the continuous problem is satisfied. By starting from the

A_ϕ	$\mu = 1.04247$	$\mu = 1.07029$	$\mu = 1.09184$
3.0×10^{-2}	0.092083	0.117908	0.135623
2.0×10^{-2}	0.092557	0.118228	0.136455
1.0×10^{-2}	0.092257	0.117877	0.136353

Table II: h/l as function of the resolution for the depth $d = 4\pi$.

trivial solution, it is initially zero. For $A_\phi = 10^{-2}$ it was of the order $\mathcal{O}(10^{-6})$ along the solution curve.

To demonstrate the result of the adaptation technique, we present the start grid in Fig. 6 and the grid corresponding to $A_\phi = 3 \times 10^{-2}$, $\mu = 1.09184$ in Fig. 7.

In Figs. 8 and 9, we plot h/l as function of the phase velocity for the depth $d = l$ and the resolution $A_\phi = 10^{-2}$. During the computation of this solution curve, the step-size in the pseudo-arclength was chosen to make the Newton iteration converge in approximately 4 steps. We calculated 43 solution points and the resolution was changed 24 times. The composite grid was corrected 43 times due to non-centered interpolations; 6 of those were done after the resolution had been changed. The fast algorithm for correcting the composite grid did not fail during any correction step. This implies that the overhead for changing the resolution corresponded to 30 out of totally 110 continuation steps, i.e. approximately 27 %. To give an example of the amount of work that was saved by using an adaptive grid, we report the number of equations as function of h/l in Fig. 10. The sparse matrix solver requires of the order $\mathcal{O}(n^2)$ operations to perform one Newton step. This means that one Newton iteration with 10^4 grid points requires the same effort as performing $\mathcal{O}(10^2)$ iterations with 10^3 grid points. These figures clearly shows the benefit of using an adaptive grid.

Next, we investigate the subharmonic bifurcation reported by [2] for the infinitely deep case. In this computation, we started with two waves in the domain, $l = 4\pi$. The depth was $d = 2\pi$ and the grid sizes for the start grid were the same as in the single wave case. We used the resolution threshold $A_\phi = 10^{-2}$. The smallest eigenvalue except for the shift-eigenvalue was monitored along the solution curve to detect singular points. Initially, the solution has period $l/2$ and it is identical to two adjacent single wave solutions. A solution with this property will be called regular.

In close agreement with [2], we found a singular point in the vicinity of $\mu = 1.08414$, $h/l = 0.06447$. By applying the technique described in §5.2 we detected a bifurcating solution curve that connects to the regular solution curve at the singular point. The eigenfunction corresponding to the second

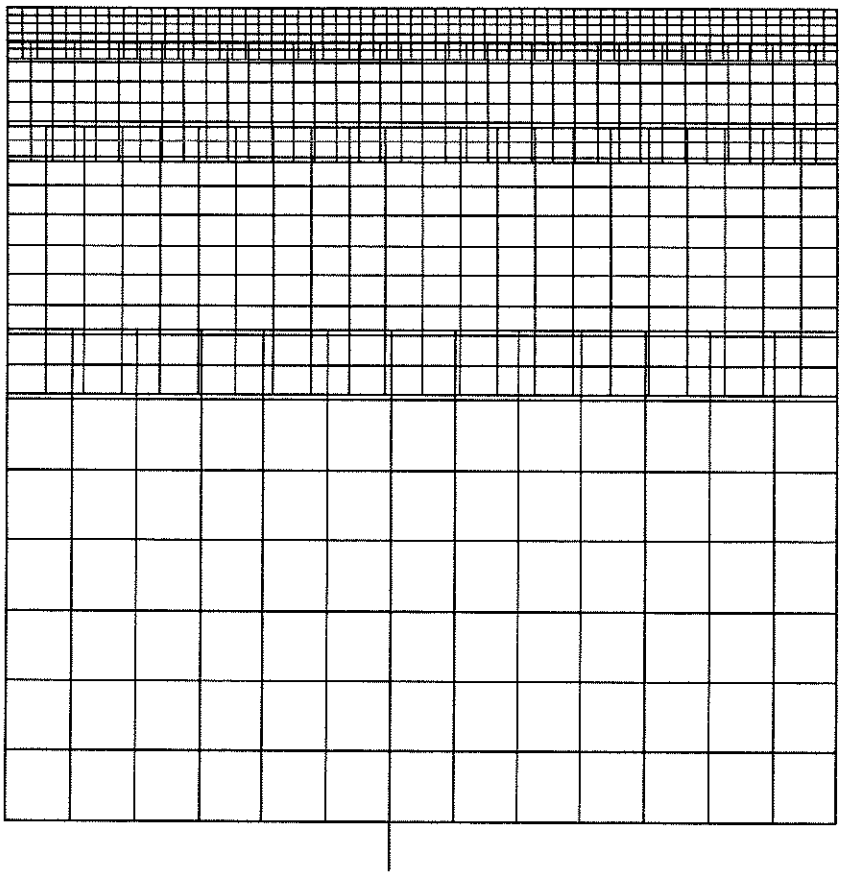

Figure 6: The start grid.

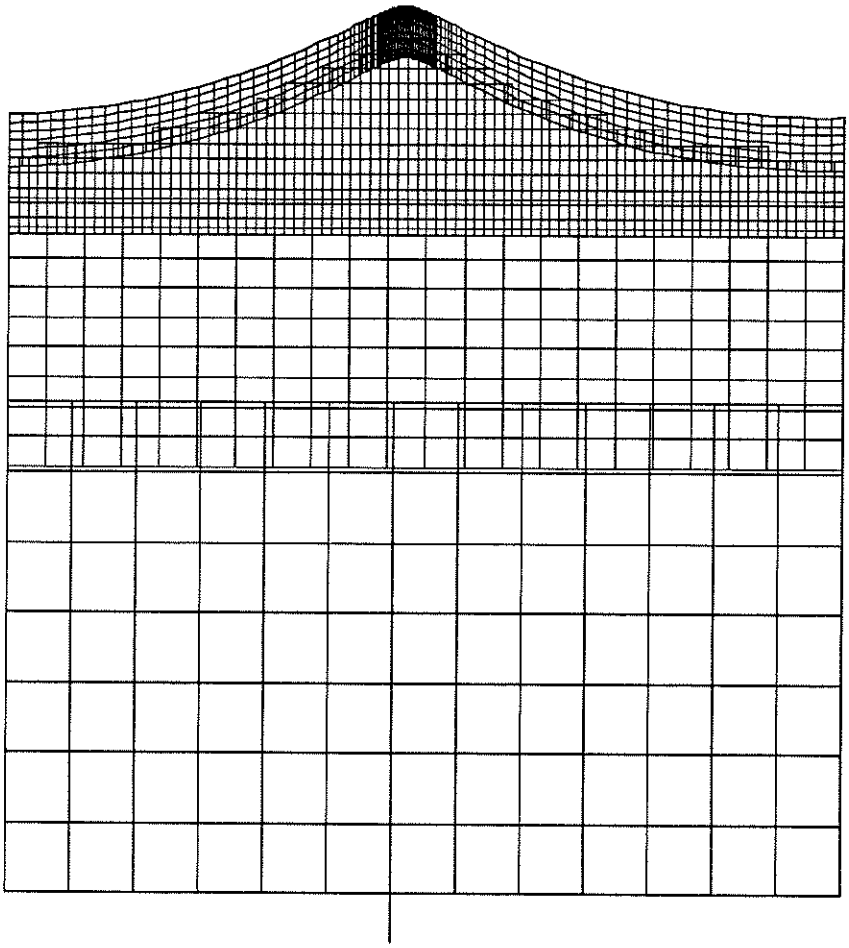

Figure 7: The grid at $\mu = 1.09184$ with the resolution $A_\phi = 3 \times 10^{-2}$.

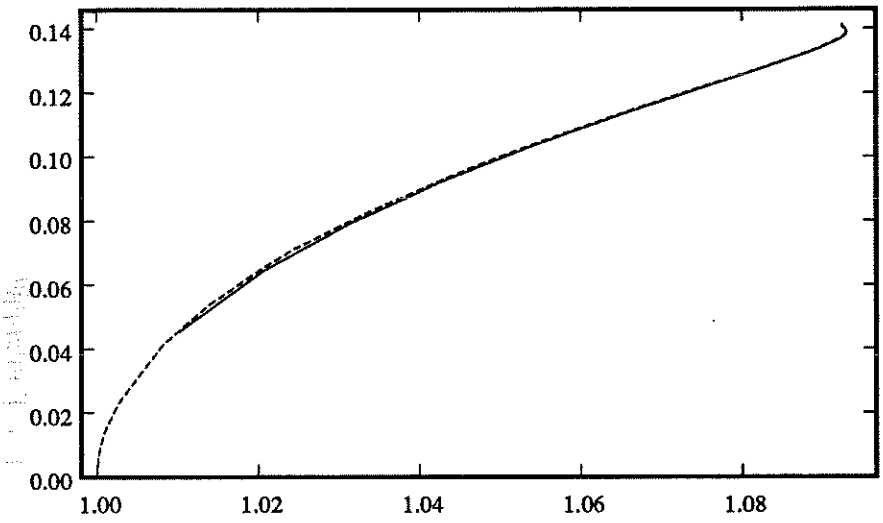

Figure 8: h/l as function of the phase velocity for the case $d = 2\pi$ and $A_\phi = 10^{-2}$. The solid line represents the result reported by [9]; the dashed line corresponds to the present work.

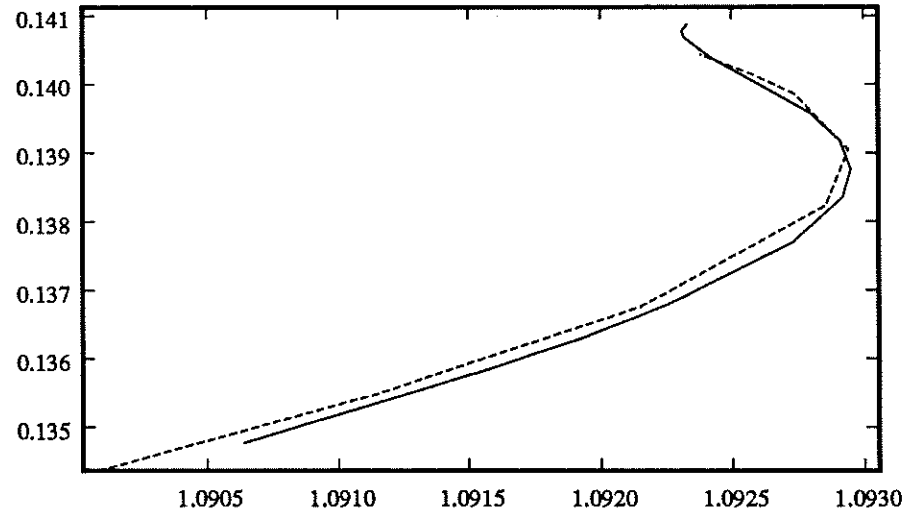

Figure 9: The same case as in Fig. 8, close to the solution of maximum height. Here, the solid line corresponds to the results of [2].

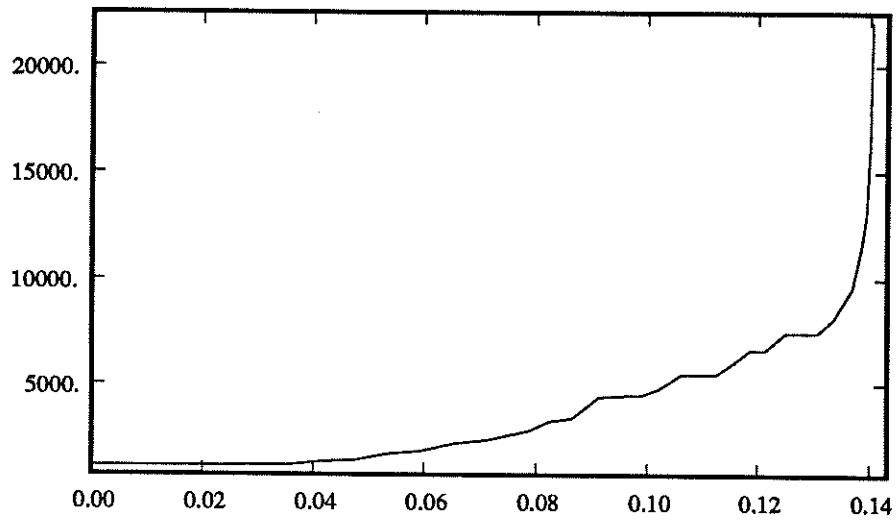

Figure 10: The number of equations as function of h/l for the case $d = 2\pi$ and $A_\phi = 10^{-2}$. The first grid had 1,086 points and the last grid had 22,053 points.

small eigenvalue is given in Fig. 11. It has period l in the x-direction, which implies that the solution on the bifurcated curve also has period l, i.e. twice the period of the regular solution. The bifurcated solution curve was traced out with the non-singular continuation method starting on the bifurcated branch close to the singular point. On the bifurcated solution curve, one of the crests becomes sharp while the other stays rounded when the wave height increases, see Fig. 12. Obviously, the two possible locations of the sharp crest correspond to identical solutions shifted by $l/2$; they are found by proceeding in opposite directions along the solution curve. In Fig. 13, we give h/l as function of the phase velocity for the bifurcated solution. The continuation was truncated when the number of equations exceeded 25,000.

Unfortunately, a direct comparison with previous results for the shallow water case [4, 12] was not possible because those results were obtained by keeping the flux, μd, constant during the continuation. In the present method, we instead keep the constant in the Bernoulli equation, Eq.(3) fixed, which implies that the flux varies slightly along our solution curves. The average depth in the two approaches will therefore be different at the same velocity for non-zero wave heights.

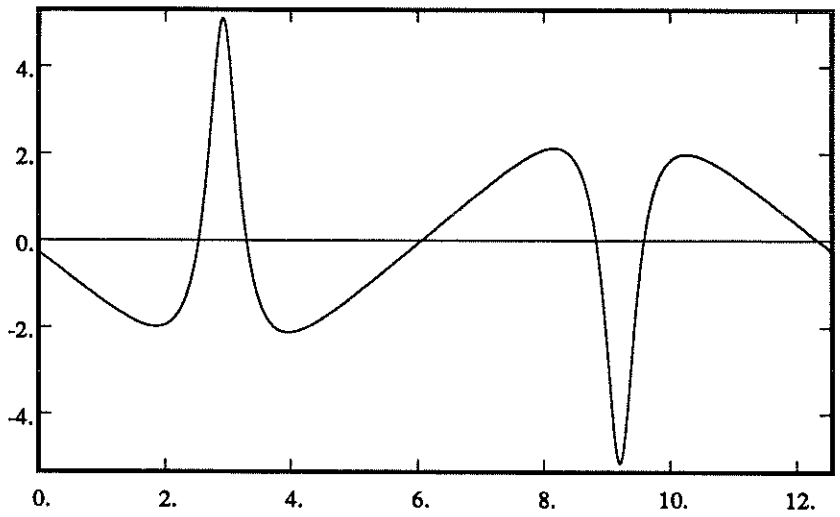

Figure 11: The surface component of the eigenfunction connected to the second small eigenvalue, close to the bifurcation point.

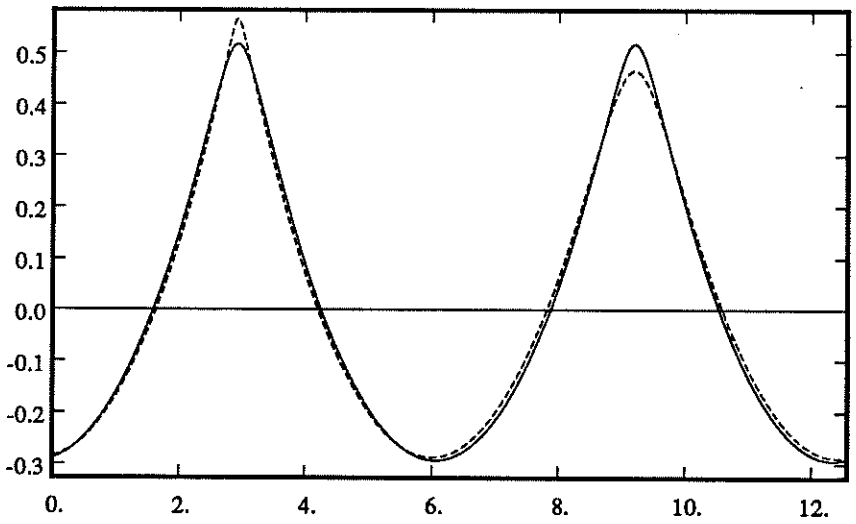

Figure 12: The surface elevation as function of x. The solid line represents the regular solution close to the bifurcation point ($\mu = 1.08414$, $h/l = 0.06447$) and the dashed line corresponds to the bifurcated solution at $\mu = 1.08132$, $h/l = 0.06786$.

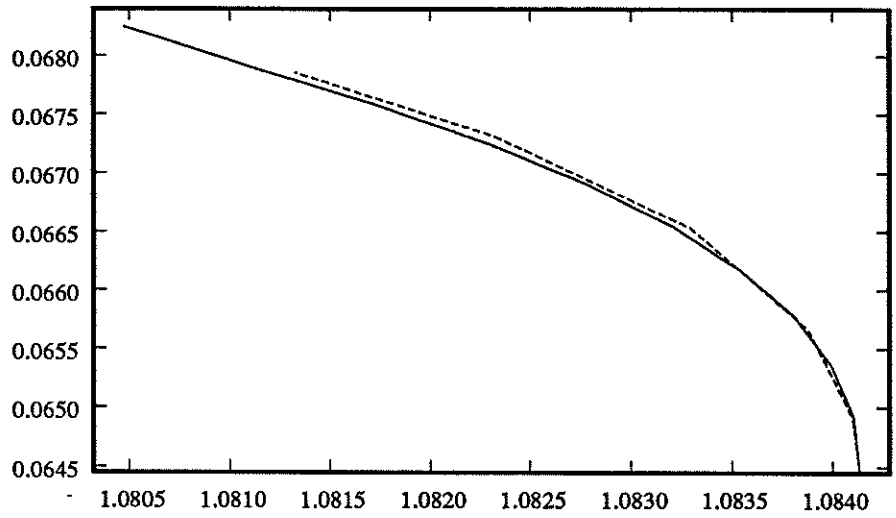

Figure 13: h/l as function of the phase velocity for the bifurcated solution. Here, $d = 2\pi$ and $A_\phi = 10^{-2}$. The solid line represents the infinitely deep case reported by [2] and the dashed line corresponds to the present work.

10 Conclusions

In this paper, it has been shown that composite overlapping grids together with finite difference methods can be used to accurately calculate steep periodic water waves. It has been indicated that an adaptive grid is necessary to achieve well resolved solutions close to the wave of maximum height. We have shown that the overhead connected to changing the resolution in the grid is small compared to the cost involved in always using a fine grid.

An underwater obstacle could easily be introduced by adding a component grid close to the obstacle and replacing the periodicity in the x-direction with appropriate in and out-flow boundary conditions. The solution is only asymptotically periodic behind the obstacle, and it would be interesting to see if the subharmonic bifurcation also appears here. This will be investigated in a future paper.

11 Acknowledgment

Professor H.-O. Kreiss is gratefully acknowledged for many inspiring discussions on the present problem. I am indebted to Dr. G. Chesshire who designed and implemented the algorithm for effectively constructing a corrected composite

grid. He and Dr. D. L. Brown are also thanked for sharing their thorough knowledge on composite overlapping grids.

References

[1] D. L. Brown, G. Chesshire, and W. D. Henshaw, "Getting Started with CMPGRD, Introductory User's Guide and Reference Manual," LA-UR-90-3729, Los Alamos National Laboratory, Los Alamos, NM (unpublished).

[2] B. Chen and P. G. Saffman, *Stud. Appl. Math.* **62**, 1 (1980).

[3] G. Chesshire and W. D. Henshaw, *J. Comput. Phys.* **90**, 1 (1990).

[4] E. D. Cokelet, *Philos. Trans. Roy. Soc. Ser. A* **286**, 183 (1977).

[5] S. Eisenstat, M. Gursky, M. Shultz, and A. H. Sherman, "The Yale Matrix Package II: The Non-Symmetric Case," Report 114, Dept. of Computer Science, Yale University, 1977 (unpublished).

[6] A.D. Jepson and H.B. Keller, "Steady State and Periodic Solution Paths: their Bifurcations and Computations," *Bifurcation: Analysis, Algorithms and Applications.* T. Kupper, H.D. Mittelmann and H. Weber eds., (Birkhauser ISNM series, no. 70, 1984), pp. 219–246.

[7] H. B. Keller "Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems," *Applications of Bifurcation Theory*, P. H. Rabinowitz, ed. (Academic Press 1977), pp. 359–384.

[8] Y. H. Kim and T. R. Lucas in *Proceedings of the 18'th Symposium on Naval Hydrodynamics, Ann Arbor, Michigan, 1990.*

[9] M. S. Longuet-Higgins, *Proc. Roy. Soc. London Ser. A* **342**, 157 (1974).

[10] N. A. Petersson and J. F. Malmliden, "Computing the Flow Around a Submerged Body using Composite Grids," Royal Institute of Technology, TRITA-NA-9028, 1990 (to appear in J. Comput. Phys.).

[11] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, 1985), p. 126.

[12] J.-M. Vanden-Broeck and L. W. Schwartz, *Phys. Fluids* **22(10)**, 1868 (1979).

[13] G. B. Whitham, *Linear and Nonlinear Waves*, (Wiley-Interscience, 1974), p. 434.