Solution of the mild-slope wave problem by iteration

VIJAY G. PANCHANG AND BRYAN R. PEARCE

Civil Engineering Department, University of Maine, Orono, ME 04469, USA

GE WEI

Civil Engineering Department, University of Maine, Orono, ME 04469. Now at: Civil Engineering Department, MIT, Cambridge, MA 02139, USA

BENOIT CUSHMAN-ROISIN

Thayer School of Engineering, Dartmouth College, Hanover, NH 03755, USA

Iterative solution procedures for solving the complete mild-slope wave (combined refraction-diffraction) equation are developed. Existing models for investigating wave refraction–diffraction in coastal areas have one of two main problems: (i) Some of the physics is lost as they resort to approximate solutions (e.g. parabolic approximations). Thus they are inappropriate in many situations. (ii) If all of the physics is to be incorporated, the problem defies computer solution except for extremely small domains (approximately 10 wavelengths), chiefly because the matrix equation associated with numerical discretization of the complete problem does not normally lend itself to solution by iteration. This paper describes the construction of iterative models that overcome both problems. First, a modified equation with an identical solution but which lends itself to iterative procedures is formulated, and the conjugate gradient method is used. A second, more rapidly converging algorithm is obtained by preconditioning.

It is shown that the algorithms can be conveniently implemented on regions much larger than those handled by conventional models, without compromising the physics of the equation. Further, they can be efficiently run in either the linear or nonlinear mode. Comparisons of model results with laboratory data and other numerical and analytical solutions are found to be excellent for several cases. The procedures thus enable the engineer to expand the scope of the mild-slope equation. As an example, an experiment is performed to demonstrate the sensitivity of the wavefield to the location of a breakwater in a region with complex bathymetry.

1. INTRODUCTION

The combined refraction–diffraction or the 'mild-slope' wave equation, derived by Berkhoff[1,2], enables the engineer to calculate wave heights in coastal regions. This equation overcomes the difficulty associated with caustics obtained while performing wave refraction calculations (ray-tracing) in regions of complex bathymetry. In recent years, it has been shown to be extremely useful in modeling surface wave propagation in a wide variety of situations (since it passes, in the limit, to the deep and shallow water equations). Examples include studies of long and short wave propagation in the vicinity of islands[2–6], wave propagation in areas of high energy dissipation such as marine vegetation[7], wave forces on floating docks[6], spectral propagation in bays[8], the construction of breakwater-gap diffraction diagrams[9], etc.

Paper accepted November 1990. Discussion closes February 1992.

Despite its immense usefulness, the refraction–diffraction model cannot be applied to several situations, particularly when tackling short wave propagation (periods on the order of seconds) in most coastal areas of significant dimensions (a few kilometers). The reason is that the equation is an inseparable elliptic partial differential equation, and the solution makes prohibitive demands on computer memory and time. It is not surprising, therefore, that in some of the applications mentioned above, approximate methods have been used to solve the equation.

Consider, for example, the case of wave propagation in Homer Spit (Alaska), studied by Ebersole *et al.*[10] [using an approximate method] on a computational domain containing roughly 100×80 nodes. If the complete elliptic boundary-value problem is to be solved for such a case, one is confronted with the task of solving 8000 simultaneous equations in complex variables. In this example the grid sizes used by Ebersole *et al.*[10] are of the order of the deepwater wavelengths. The requirements

are more stringent for shallow-water waves with shorter wavelengths. Also, in many situations much finer resolution is required (say 5 points per wavelength), increasing the number of equations to 200,000. As this system of equations does not lend itself to iterative solution (indirect methods), Gaussian elimination is used, necessitating the storage of a huge system matrix. Although Houston[3] has successfully solved the complete problem for over 10,000 nodes, it involved considerable difficulty, and it is clear that the need to use direct methods severely restricts the size of the domain in which wave propagation can be simulated with the mild-slope equation.

For most applications, then, one resorts to approximate methods of solving the mild-slope equation. These approximate methods consist of: (i) the parabolic equation method[11–13], (ii) the RCPWAVE model[10,14], and (iii) the EVP model[15]. These models avert computational problems, but the computational convenience is obtained at the cost of some of the physics contained in the governing equation. In particular, they have one or both of the following restrictions:

a. The waves must have a principal propagation direction (say x), and diffraction effects are restricted to the y-direction only (for Cartesian grid models).
b. The reflected (backscattered) component of the wave potential in the negative x-direction should be negligibly small.

Thus the approximate models are inappropriate when the bathymetry or structures such as harbor walls, breakwaters, etc. reflect energy in the $(-x)$ direction, and/or when the scattering is at large angles. The above restrictions thus limit the applicability of these methods. Further, these methods encounter considerable difficulties when the domain is non-rectangular or when the angle of incidence is varied substantially.

Another class of recent models[16–19], is based on the solution of the time-dependent hyperbolic equations associated with the mild-slope equation. Although no approximations (such as *a* or *b* above) are necessary, integration over an extremely large number of wave periods with small timesteps (usually governed by the Courant condition) is required for model spin-up. An inordinate amount of computer time is required for large domains, and, moreover, the treatment of open boundaries for the time-dependent equation is difficult. Dong and Al-Mashouk[20] report that in their study the hyperbolic approach did not fare as well as the elliptic problem.

In situations where the above models are inadequate, the complete elliptic mild-slope equation must be solved. For small domains (approximately 10 wavelengths) this can be done fairly efficiently using direct Gaussian elimination. For larger domains, the computer requirements of Gaussian elimination are prohibitive, despite recent advances in sparse matrix technology. Storage requirements of iterative procedures, on the other hand, are extremely modest, even for large domains. These procedures also accommodate non-rectangular domains easily. However, as already indicated, the mild-slope equation cannot be solved by standard iterative procedures, since the conditions required for convergence of the iterations are not usually satisfied; the details are described in Section 2. In this paper, we modify the basic matrix equation associated with finite-difference treatment of the mild-slope equation in a manner that allows the use of an iterative procedure. The iterative method then uses the conjugate-gradient algorithm, which is described in detail in Section 3. Further, a second algorithm is implemented, based on a modification of the conjugate-gradient algorithm[21], which considerably enhances the rate of convergence. In Section 4, the procedures are used to obtain simulations of wave refraction–diffraction and reflection in rectangular and non-rectangular coastal regions, using both linear and nonlinear dispersion relations. The results are compared with laboratory data and other existing solutions. Concluding remarks are offered in Section 5.

2. MATHEMATICAL PROBLEM STATEMENT

a. Governing equations, boundary conditions

The combined refraction–diffraction equation[2,22] that describes the propagation of periodic, small-amplitude, surface gravity waves over an arbitrarily varying, mild-sloped sea-bed is:

$$\nabla.(CC_g\nabla\Phi)+\frac{C_g}{C}\sigma^2\Phi=0, \tag{1}$$

where

$\Phi(x, y)$ = complex surface elevation function, from which the wave height can be estimated.
σ = wave frequency under consideration,
$C(x, y)$ = phase velocity $=\sigma/k$
$C_g(x, y)$ = group velocity $=\partial\sigma/\partial k$
$k(x, y)$ = wavenumber $(=2\pi/L)$, related to the local depth $d(x, y)$ through the dispersion relation:

$$\sigma^2=gk\ \mathrm{Tanh}(kd). \tag{2}$$

A detailed derivation of equation (1) may be found in Berkhoff[2]. Essentially, equation (1) represents the vertically integrated form of Laplace's equation for linear wave propagation, with the vertical variation for Φ assumed to be the form that corresponds to a horizontal bed. As such it is valid when the bottom has a 'mild-slope', characterized by $\nabla d/kd=0(\varepsilon)\ll 1$. Instead of working with equation (1), it is convenient to work with the following wave equation:

$$\nabla^2\phi+K^2(x, y)\phi=0, \tag{3}$$

which is obtained from equation (1) through the transformation:[12]

$$\phi=\Phi(CC_g)^{0.5}\text{ and }K^2=k^2-\frac{\nabla^2(CC_g)^{0.5}}{(CC_g)^{0.5}}. \tag{4}$$

In this formulation, ϕ is a modified wave potential function and K is a modified wavenumber. We note that the above transformation is merely for convenience of demonstration of the procedure; all operations performed here on the reduced wave equation (3) can be applied to Berkhoff's original equation (1). Similarly, a rectangular domain is chosen for demonstration. (The method can be easily applied to non-rectangular regions, with internal boundaries when necessary, as shown in Section 4.) The domain, coordinate axes, incident wave direction, etc. are shown in Fig. 1.

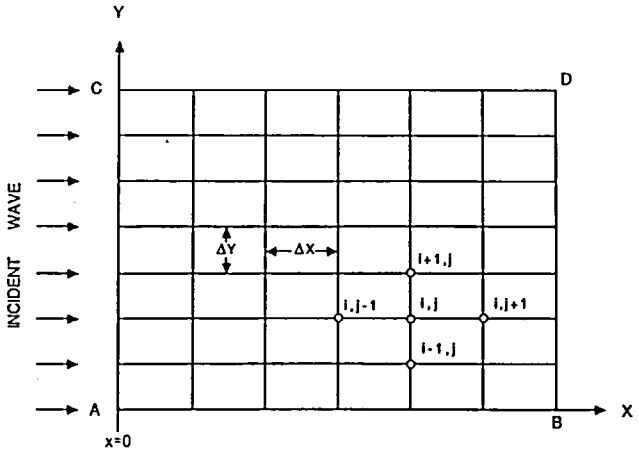


Fig. 1. Model grid

Boundary conditions have to be imposed along AC, BD, CD, and AB, and in general, they may be written as:

$$L(\phi) = 0. \tag{5}$$

More specifically, the boundary conditions are mostly of the Robbins type. Along AC, the incident wave is $\phi_{in} = A_{in} \exp(iKx)$, and A_{in} is specified as unity ($i = \sqrt{-1}$). There exists also a backscattered component, which may be approximated by $\phi_r = B[\exp(-iKx)]$. (For more detail see Ref. 23.) Since B is not necessarily known, we can differentiate,

$$\frac{\partial \phi}{\partial x} = iK(\phi_{in} - \phi_r) = iK(\phi_{in} - (\phi - \phi_{in})), \tag{6}$$

and obtain

$$\frac{\partial \phi}{\partial x} = iK(2 - \phi), \tag{7}$$

since $\phi_{in} = 1$ at $x = 0$. Equation (7) can be one form of the boundary condition along AC. Alternatively, the wavemaker condition

$$\frac{\partial \phi}{\partial x} = \text{Constant} \tag{8}$$

can be used, as in Ref. 24. If wave conditions along AC are known or calculated using some other coarser model, the Dirichlet condition may be used.

The other boundaries may represent a seawall, an open boundary, or a coastline. In such cases the following condition may be used:[25]

$$\frac{\partial \phi}{\partial n} - iK\alpha\phi = 0, \tag{9}$$

where n is the direction normal to the boundary, and α is a reflection coefficient that varies with the type of boundary and may have to be determined empirically. Boundary conditions similar to those described above or variations thereof are also used[7,26,27]

b. Finite-difference representation

The domain (Fig. 1) is discretized into grids of size Δx and Δy. If ϕ_j^i is used to denote the grid-point value of the potential, standard discretization of equation (1) using second-order finite-differences (for $\Delta x = \Delta y$) yields:

$$\phi_j^{i-1} + \phi_j^{i+1} + \phi_{j-1}^i + \phi_{j+1}^i + [(K\Delta x)^2 - 4]\phi_j^i = 0. \tag{10}$$

The conventional approach consists of writing such equations for all points in the domain. The resulting system of equations may be expressed in matrix form as:

$$[A]\{\phi\} = \{f\}, \tag{11}$$

where $[A]$ is the system matrix, $\{\phi\}$ is the unknown vector (of the desired grid-point values of the wave potential), and $\{f\}$ is a vector that contains information from the discretized boundary conditions (5). The brute-force approach consists of solving equation (11) for $\{\phi\}$ using Gaussian elimination, which requires storage for the matrix $[A]$. Note that even when there are as few as 1000 unknowns, $[A]$ contains 1000×1000 complex elements. Although, with considerable effort, advantage can be taken of the sparsity of $[A]$ to store it in a 'packed form', Houston[3] has noted that 'even memory requirements of the packed form ... are excessive'. Experience indicates that solution by direct Gaussian elimination is practically impossible for regions larger than about ten times the wavelength squared, due to computer storage problems.

3. SOLUTION BY ITERATION

Unlike direct methods (e.g. Gaussian elimination), indirect (iterative) methods do not require the storage of $[A]$, and hence can be used even on large domains. For convergence of such procedures, the coefficient matrix $[A]$ must usually be strictly diagonally-dominant, or it must be symmetric and positive–definite[28]. However, equation (10) does not lead to a diagonally-dominant matrix: the sum of the coefficients of the non-diagonal elements in the matrix ($= 4$, resulting from the first four terms) is usually larger, in absolute value, than the coefficient of the diagonal element, (viz, $(K\Delta x)^2 - 4$). Also, the presence of complex quantities in the boundary conditions usually renders $[A]$ non-Hermitian, and hence not positive–definite. Conventional iterative methods (e.g. Jacobi, Gauss–Seidel, SOR, etc.) therefore do not guarantee convergence when applied to equation (11). Of particular interest to us is the conjugate–gradient (CG) method for solving large sets of simultaneous equations; this iterative procedure has been shown to converge several orders of magnitude faster than many other schemes[29,30], for instance, for the finite-differenced Laplace equation. However, the CG method converges only when the system matrix is symmetric and positive–definite. Although our system matrix $[A]$ is symmetric, it is not positive-definite, and the basic conjugate gradient method will not work without some adaptation. A remedy is to use the Gauss transformation, i.e. multiply equation (11) by $[A^*]$, the complex conjugate transpose of $[A]$:

$$[A^*][A]\{\phi\} = [A^*]\{f\}. \tag{12}$$

The new coefficient matrix $[A^*][A]$ is always symmetric and positive–definite, and the modified CG procedure for equation (12) will converge. The algorithm, which we shall refer to as algorithm 1, is as follows[31]:

1. Select trial values ϕ_0 (i.e. $i = 0^{\text{th}}$ iteration) for all grid points where the solution is desired.
2. Compute for all points $r_0 = f - A\phi_0$ and $p_0 = A^* r_0$.

3. Compute for the ith iteration:

$$\alpha_i = \frac{|A^* r_i|^2}{|Ap_i|^2}$$

4. Update $\phi_{i+1} = \phi_i + \alpha_i p_i$ for all points.
5. Check for convergence of solution.
6. Compute, for each grid point, $r_{i+1} = r_i - \alpha_i A p_i$.
7. Compute for the ith iteration:

$$\beta_i = \frac{|A^* r_{i+1}|^2}{|A^* r_i|^2}$$

8. Compute $p_{i+1} = A^* r_{i+1} + \beta_i p_i$.
9. Set $i = i + 1$, and go to step 3.

Note that each premultiplication of any vector (say g) by $[A]$ is simply equivalent to obtaining the finite-difference approximation to $\nabla^2 g + K^2 g$ for each grid point. The g values on the boundaries needed for the product Ag are determined using the finite-difference form of the boundary conditions applied to g, i.e. g(boundary point)$= \zeta$. g(internal point). Note that the homogeneous forms of the boundary conditions have to be used while computing Ag. For $Ag - f$ needed in step 2, the actual boundary conditions are used, i.e. g(boundary point) $= \zeta$. g(internal point) $+ \delta$. As $[A^*]$ differs from $[A]$ only in the complex diagonal elements (arising from the complex boundary conditions), multiplication of any vector g by $[A^*]$ is the same as multiplying it by $[A]$, except that g(boundary point) $= \zeta^*$. g(internal point), where ζ^* is the complex conjugate of ζ. The factors α and β involve the product $|\mathrm{d}|^2 = (\mathrm{d}, \mathrm{d})^2 = \Sigma\, \mathrm{d}_j . \mathrm{d}_j^*$, where d_j represents elements of the vector d.

The procedure is convenient to implement even on non-rectangular domains, since the algorithm simply hops from one grid point to the next, and no matrix needs to be stored. In fact, only storage for the three arrays r, p, and ϕ are needed, i.e. if there are 1000 unknowns, storage is required for only 3000 complex values. Convergence of the iterations is assured, and the solution should require a maximum of 1 iteration per grid-point[32]. The criterion for convergence used in our study was:

$$\frac{\Sigma |(\nabla^2 \phi + K^2 \phi)|^2}{\Sigma |\phi|^2} < \varepsilon,$$

where the summations extend over all grid-points and ε is a prescribed tolerance. Checks for convergence were made every 100 iterations.

The procedure described above requires very little programming manpower, but convergence, although guaranteed, is rather slow. This is due to the coefficient matrix $[A^*][A]$ of the transformed equation (12) having a far wider spectral range than the original matrix $[A]$. The convergence properties can be significantly enhanced by using a real matrix Q (to be specified shortly) to precondition (12):

$$[Q]^{-1}[A]\{\phi\} = [Q]^{-1}\{f\},$$

or

$$([Q]^{-1}[A][Q]^{-T})\,([Q]^T\{\phi\}) = [Q]^{-1}\{f\},$$

which may be written as

$$[A']\{\phi'\} = \{f'\}. \tag{13}$$

where

$$[A'] = ([Q]^{-1}[A][Q]^{-T}),\ \{\phi'\} = ([Q]^T\{\phi\},\ \{f'\} = [Q]^{-1}\{f\}. \tag{14}$$

Now performing the transformation on the preconditioned equation (13), we have

$$[A'^*][A']\{\phi'\} = [A'^*]\{f'\}. \tag{15}$$

In terms of the primed variables, the solution algorithm for equation (15) is identical to algorithm 1 for equation (12). In terms of the original variables (after some manipulations described in Appendix A), the resulting algorithm (algorithm 2) is:

1. Select trial values ϕ_0(i.e. $i = 0^{th}$ iteration) for all grid points where the solution is desired.
2. Compute for all points $R_0 = A^* M^{-1}(f - A\phi_0)$ and $P_0 = M^{-1} R_0$. (The matrix M^{-1} is defined in equation (16) below.)
3. Compute for the ith iteration:

$$\alpha_i = \frac{(R_i, M^{-1} R_i)}{(Ap_i, M^{-1} Ap_i)}$$

4. Update $\phi_{i+1} = \phi_i + \alpha_i P_i$ for all points.
5. Check for convergence of solution.
6. Compute, for each grid point, $R_{i+1} = R_i - \alpha_i A^* M^{-1} A p_i$.
7. Compute for the ith iteration:

$$\beta_i = \frac{(R_{i+1}, M^{-1} R_{i+1})}{(R_i, M^{-1} R_i)}$$

8. Compute $P_{i+1} = M^{-1} r_{i+1} + \beta_i P_i$.
9. Set $i = i + 1$, and go to step (3).

In this algorithm,

$$M = QQ^T \quad \text{or} \quad M^{-1} = Q^{-T} Q^{-1} \tag{16}$$

The procedure described above follows the approach of Bayliss *et al.*[21] who rigorously obtained the preconditioner Q required to substantially improve the convergence rate. We choose a minor modification proposed by Axelsson *et al.*[33] that yields:

$$Q = [(2 - \omega)\omega]^{-1/2}(D_0 + \omega L_0) D_0^{-1/2}, \tag{17}$$

where D_0, L_0, and U_0 are the diagonal, lower, and upper triangular matrix components of the matrix A_0, which is obtained by setting $K = 0$ in the matrix A. (i.e. $D_0 + L_0 + U_0 = A_0$, where A_0 corresponds to the discrete Laplacian.) The factor ω is a scalar relaxation parameter analogous to that encountered in SOR. The implementation of algorithm 2 is not very different from that of algorithm 1, except for the computations involving $[M]$. Specifically, we have to calculate products of $[M^{-1}]$ with several other vectors. If Y denotes the product $M^{-1}R$, then the use of equations (16) and (17) yields:

$$Y = (D_0 + \omega U_0)^{-1} D_0 (2 - \omega)(D_0 + \omega L_0)^{-1} \omega R. \tag{18}$$

(since A_0 is symmetric and thus $L_0^T = U_0$). Equation (18) is first rewritten as:

$$Y' = (D_0 + \omega L_0)^{-1} \omega R \tag{19}$$

$$Y = (D_0 + \omega U_0)^{-1} D_0 (2 - \omega) Y' \tag{20}$$

The product, equation (18), is carried out in two steps.

First Y' can be determined, given R, through manipulation of equation (19) as:

$$Y' = D_0^{-1}\omega(R - L_0 Y'), \tag{21}$$

which is equivalent to sweeping the grid from corner to opposite corner (point (1, 1) to point (M, N)), constructing the vector

$$Y'(i,j) = \left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)^{-1} \omega\left[R(i,j) - \left(\frac{Y'(i-1,j)}{\Delta x^2} + \frac{Y'(i,j-1)}{\Delta y^2}\right)\right], \tag{22}$$

by using values calculated along the way. Y can then be obtained from equation (20) as:

$$Y = (2-\omega)Y' - D_0^{-1}\omega U_0 Y, \tag{23}$$

which is equivalent to sweeping the grid in reverse order, constructing the vector

$$Y(i,j) = (2-\omega)Y'(i,j) - \left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right)^{-1} \omega\left(\frac{Y(i+1,j)}{\Delta x^2} + \frac{Y(i,j+1)}{\Delta y^2}\right) \tag{24}$$

Thus, the product of $[M^{-1}]$ with any vector can be achieved by performing a forward and a backward sweep on the Laplacian, as in SSOR. Note that equations (22) and (24) apply to internal grid points. Boundary values of Y' required in the sweep are determined from R (and similarly those of Y from Y') using homogeneous forms of the original boundary conditions with $K = 0$. For the boundary conditions described in Section 2, these usually reduce to simply $\phi = 0$ or $\partial\phi/\partial n = 0$.

Although algorithm 2 requires greater effort to code and performs more arithmetic operations per iteration than algorithm 1, convergence is reached after substantially fewer iterations, resulting in considerable savings of computer time (see Section 4).

4. MODEL VERIFICATION

The models developed in the preceding section were used to simulate several cases of combined refraction–diffraction with and without significant reflections, and of wave reflection–diffraction in harbours. The results were compared with other mathematical solutions and laboratory data. First, we describe model simulations of wave propagation over varying bathymetry (in intermediate water depths). Three bathymetric configurations were considered: a circular shoal, an elliptic shoal (each surrounded by a region of constant depth), and more complicated bathymetry varying throughout the domain. Data for these cases were obtained in hydraulic models by Ito and Tanimoto[34], Vincent and Briggs[35], and Berkhoff *et al.*[24] respectively.

Both numerical algorithms described above were used and yielded almost identical results, as expected. These results matched the laboratory data extremely well in all three cases. For brevity, we present only one set of results, pertaining to wave propagation in the third case (over the complicated bathymetry used by Berkhoff *et al.*[24]). The bathymetry (Fig. 2) consists of an elliptic shoal

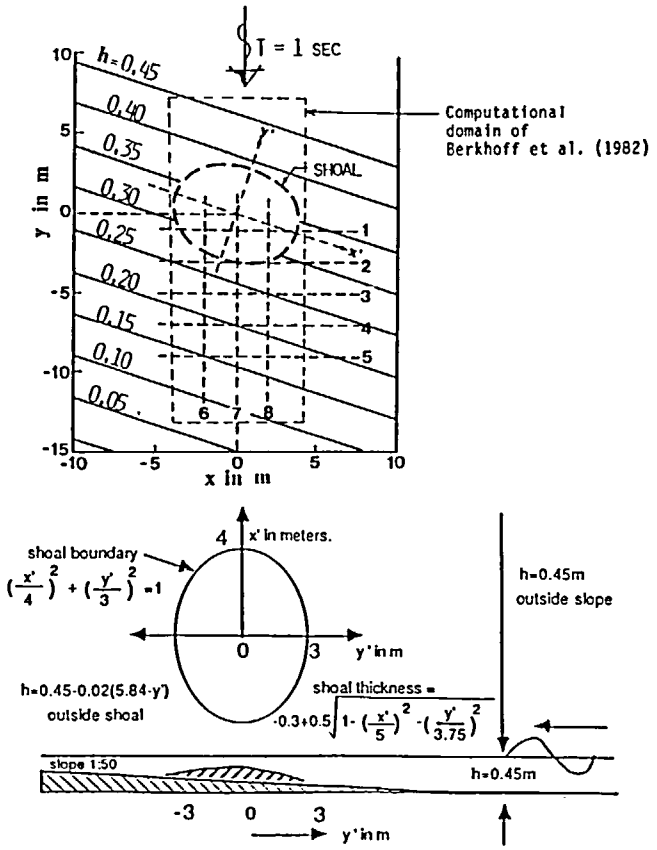


Fig. 2. Bathymetry from hydraulic model of Berkhoff et al. *(1982)*

situated on a bottom sloping at an angle to the incoming wave direction. This case is a perfect example that demonstrates the advantages of the iterative method over conventional models. In addition to obtaining laboratory data, Berkhoff *et al.*[24] constructed a finite-element model. However, thier computational domain (Fig. 2) covered only about 31% of the laboratory model, because the solution on the full model region would have required too many grid points. The iterative algorithms, on the other hand, can easily handle the entire domain (and larger domains as well). The results discussed here (Fig. 3a–h) were obtained with 220 × 200 = 44,000 nodes*. In Fig. 3(a–h) the numerical results and laboratory data of Berkhoff *et al.*[24] are compared with the iterative model results. Overall, the agreement is reasonably good. Most important, the iterative model results are extremely close to the results of the pseudospectral model developed by Panchang and Kopriva[18], which also entails no physical approximations. It is satisfying that two models containing similar physics produce almost identical results. (The disadvantage of the pseudospectral model relative to the present algorithm is that it is a hyperbolic model using a non-uniform Chebychev grid, and that the timestep is dictated by the smallest grid-size and the CFL criterion. The difficulty with hyperbolic models is noted in Section 1.)

* Recall that in the published works, the largest refraction–diffraction problem solved by conventional methods had roughly 10,000 nodes. Kostense *et al.*[36] also report the solution of large problems, but give no description of their procedure.

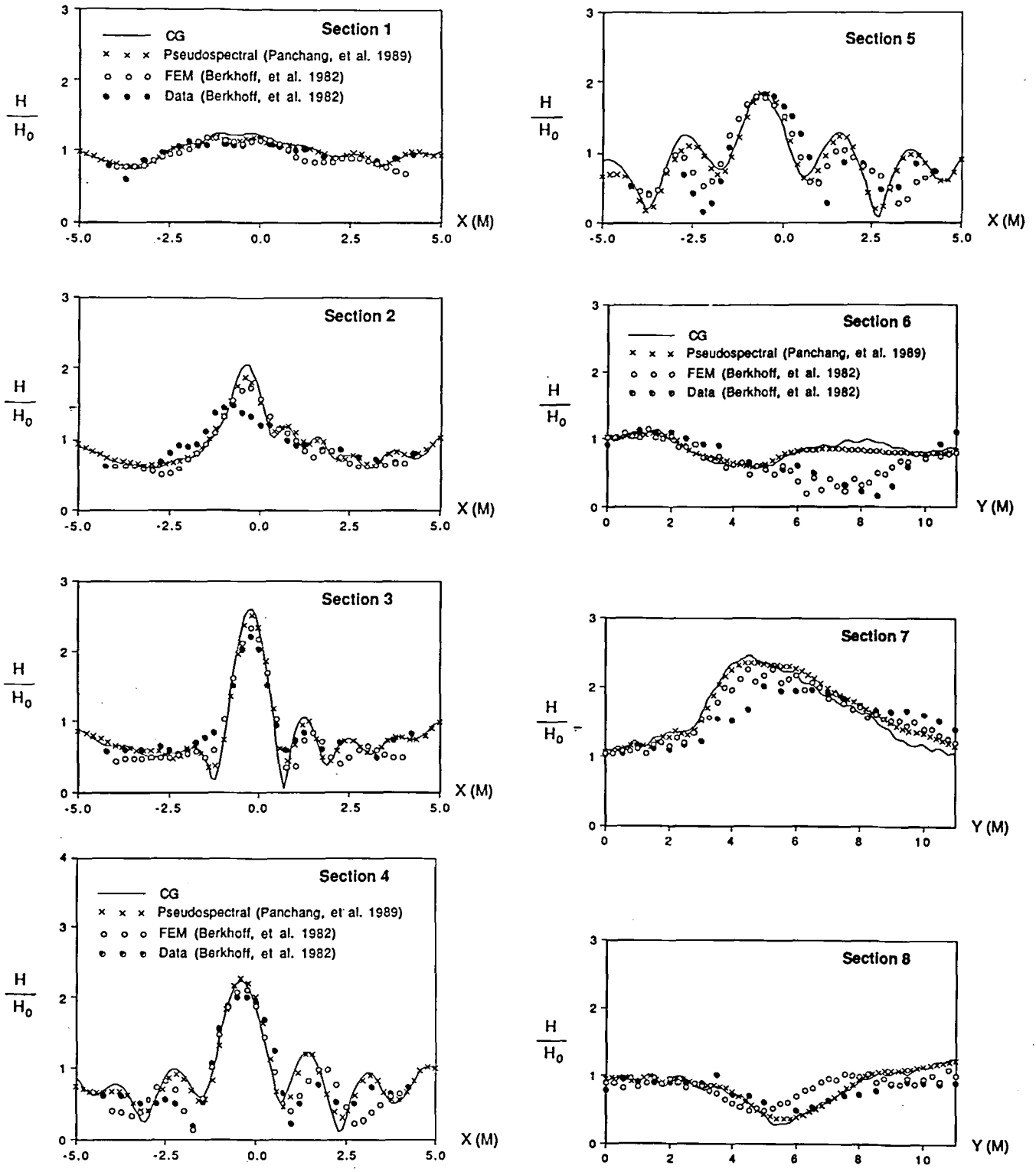


Fig. 3(a–h). Wave height comparisons for bathymetry of Fig. 2

Although the mismatch between the present numerical results and data seen in Fig. 3(a–h) is within the range of contemporary short-wave models, Kirby and Dalrymple[37] have shown that the inclusion of nonlinear effects can lead to a better fit. One approach proposed by Kirby and Dalrymple[38] consists of using the following nonlinear dispersion relation (instead of equation (2)) for calculating the wavenumbers:

$$\sigma^2 = gk[1 + (ka)^2 F_1 \operatorname{Tanh}^5(kd)] \operatorname{Tanh}[kd + (ka)F_2] \quad (25)$$

where

$$F_1 = \frac{\cosh(4kd) + 8 - 2\tanh^2(kd)}{8 \sinh^4(kd)} \quad \text{and} \quad F_2 = \left(\frac{kd}{\sinh(kd)}\right)^4$$

and a is the wave amplitude. This dispersion relation can be used for a wide range of water depths and wave conditions. Since equation (25) involves the unknown wave amplitude (the calculation of which is our objective), the solution process now requires a round of iterations for establishing the correct k. First, the elliptic problem is solved with the linear dispersion relation (2); these solutions are discussed above. The computed amplitudes are then used in equation (25) to obtain new estimates of k, and the elliptic problem is solved again using the iterative algorithms. This procedure is repeated 5 times, using equation (25) and the amplitudes computed in the previous round to update k. For a given set of k's, the iterations for the elliptic problem are stopped when the tolerance ε is 10^{-4}. Solving this problem is thus equivalent to simply solving the linear problem several times, with a different set of wavenumbers used each time. The iterative method is particularly efficient for handling this, since successively fewer iterations are required in each round to obtain a specified level of convergence; this is in contrast to the conventional direct method, where each round would require the same effort for constructing and inverting the matrix equation. The solutions obtained are compared with the previous linear solutions and data in Fig. 4(a–h). The agreement between the nonlinear solutions and data is excellent; the nonlinear results clearly yield a substantially better fit with data than the linear model results, particularly in the side-lobes of Fig. 3(c–e). This example shows that the present model can be efficiently used in either the linear or nonlinear mode. The comparisons of model results with laboratory data for the other two shoal cases were similarly very good, although only the linear case was examined.

Although it is possible to simulate wave propagation in the above cases using other simpler models (e.g. RCPWAVE, PEM, etc.), these models cannot be used in the presence of strong reflections and/or lateral scattering. Since the present model has no such limitations, it can be easily applied even when a shore-protection structure such as a seawall is placed at the downwave end of the domain of Fig. 2 (replacing the outgoing boundary condition used earlier); the results are presented in Fig. 5 as a contour diagram. These clearly indicate a standing-wave pattern due to the presence of the seawall, and consequently the wave heights are much larger. The asymmetry due to the bathymetric effects is also more pronounced than in Fig. 4(a–e). No experimental data are available for verifying these predictions, but the results of two other models that can handle reflections, the EVP and the pseudospectral models were very similar (although not identical, chiefly because of grid-spacing limitations of these latter models.)

Because the iterative algorithm is so versatile, two additional experiments were performed with this bathymetry by inserting a breakwater halfway across the domain, not too far from the seawall. Reflective boundary conditions were applied on each side of this breakwater, which was given a thickness equal to one grid size. The two experiments differed in the location of the breakwater; it was located at $y = -5$ m and $y = -4.4$ m in the two experiments. The small displacement was chosen to correspond to half of the local wavelength (π/K). When the results (Figs 6a and b) are compared and contrasted with the results in the absence of the breakwater (Fig. 5), important differences strike the eye.

First, the incoming beam wandering along the middle of the domain is intercepted by the tip of the breakwater; a part of it is reflected and another part goes forward, undergoing severe diffraction. Upwave from the breakwater (toward positive y), the reflected ray disperses somewhat and leads to a noticeable accumulation of energy to the left of the domain. On the right side, the local phases change considerably from run to run, but the overall pattern seems to remain the same, presumably because the incident beam hits the breakwater at an angle favoring reflection toward only one side (the breakwater side).

Behind the breakwater, the solution is governed by the diffraction of the incident beam upon encountering the breakwater tip. It is considerably affected not only by the presence of the breakwater but also by its exact position. Indeed, in one case (breakwater at $y = -5$; Fig. 6a), the semi-enclosed area is shielded by the breakwater, and the energy is relatively low, while in the other case (breakwater at $y = -4.4$; Fig. 6b), a large standing wave is present, indicating resonance. Since the wavelength is dependent on the local bathymetry, the determination of the exact breakwater position(s) for which resonance occurs could hardly have been predicted *a priori*. Energy supplied to the resonant wave is provided by the diffraction of the incident, central beam as it hits the breakwater tip, which is a complicated problem. Thus, an *a priori* determination of the height of the standing wave would have been impossible.

Finally, we describe model simulations of wave propagation in a rectangular harbor connected to the open ocean (Fig. 7), for different incident wavelengths (harbor resonance). Such a case can be conveniently handled by the boundary element method if the depth is constant. Alternatively hybrid finite-element formulations[25,39] may be applied if the domain is small. The iterative procedure developed here permits us to cover a much larger portion of the ocean region. This is particularly advantageous because the results are somewhat sensitive to the proper formulation of the outgoing boundary condition and hence also to the extent of the infinite ocean included in the simulation[17]. Here we approximate the infinite ocean outside the harbour by a finite rectangular region (Fig. 7) of approximately the same size as that used by Madsen and Larsen[17]. Equation (7) was used on the incoming boundary, and equation (9) on the artificial boundaries (with $\alpha = 0, 0.5, 1.0$ on the lateral boundaries). The computed amplification (for several different wavelengths) at the center of the backwall of the harbor (Fig. 7) are compared with the theoretical solution of Lee[40] in Fig. 8, and the simulation may clearly be considered successful.

The computer runs described above were made on the IBM3090 machines at the University of Maine and the Cornell Theory Center. The computer time required for model convergence depends, of course, on the starting values ϕ_0 and the prescribed tolerance ε. In all examples reported here ϕ_0 was assigned a value of $(0.5 + 0.5i)$ for uniformity, although better choices are possible in some cases. The value of ε had to be determined by experimentation, and it was found that 10^{-n}, where n is about 5, normally sufficed. The results shown in Fig. 3 were obtained with $\varepsilon = 10^{-7}$ and required about 1250 seconds of CPU time using the preconditioned algorithm. However, identical solutions were obtained with $\varepsilon = 10^{-4}$, requiring only 400 iterations and 220 seconds. Tables 1–3 compare the performance of the two algorithms (with a

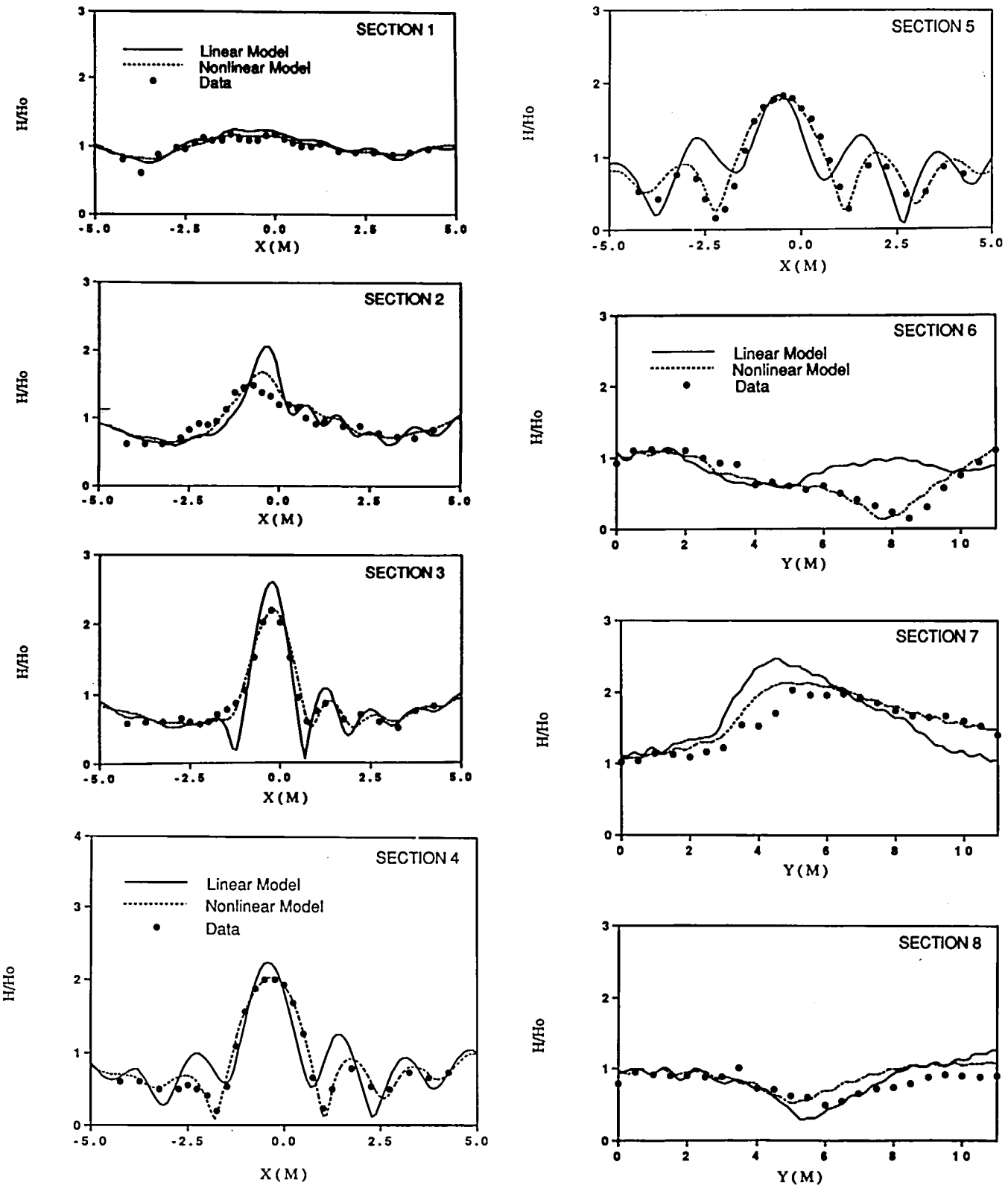


Fig. 4(a–h). Wave height comparisons for bathymetry of Fig. 2; linear and nonlinear models

much higher n chosen in some cases for illustration). The comparisons are made on the basis of either the computer time required to obtain a specified residual ε or the value of the residual after a certain number of iterations. It can be seen that Algorithm 2 is extremely effective in reducing the required computer time, and in some cases (the harbor problem), requires less than an eighth of the time required by the unpreconditioned Algorithm 1 (Table 3, $\omega = 1.75$). The optimum value of ω of course depends on the eigenvalues of the problem, and no rule exists to establish a best value *a priori*. However, it apears that substantial benefits would accrue with $\omega \approx 1.5$.

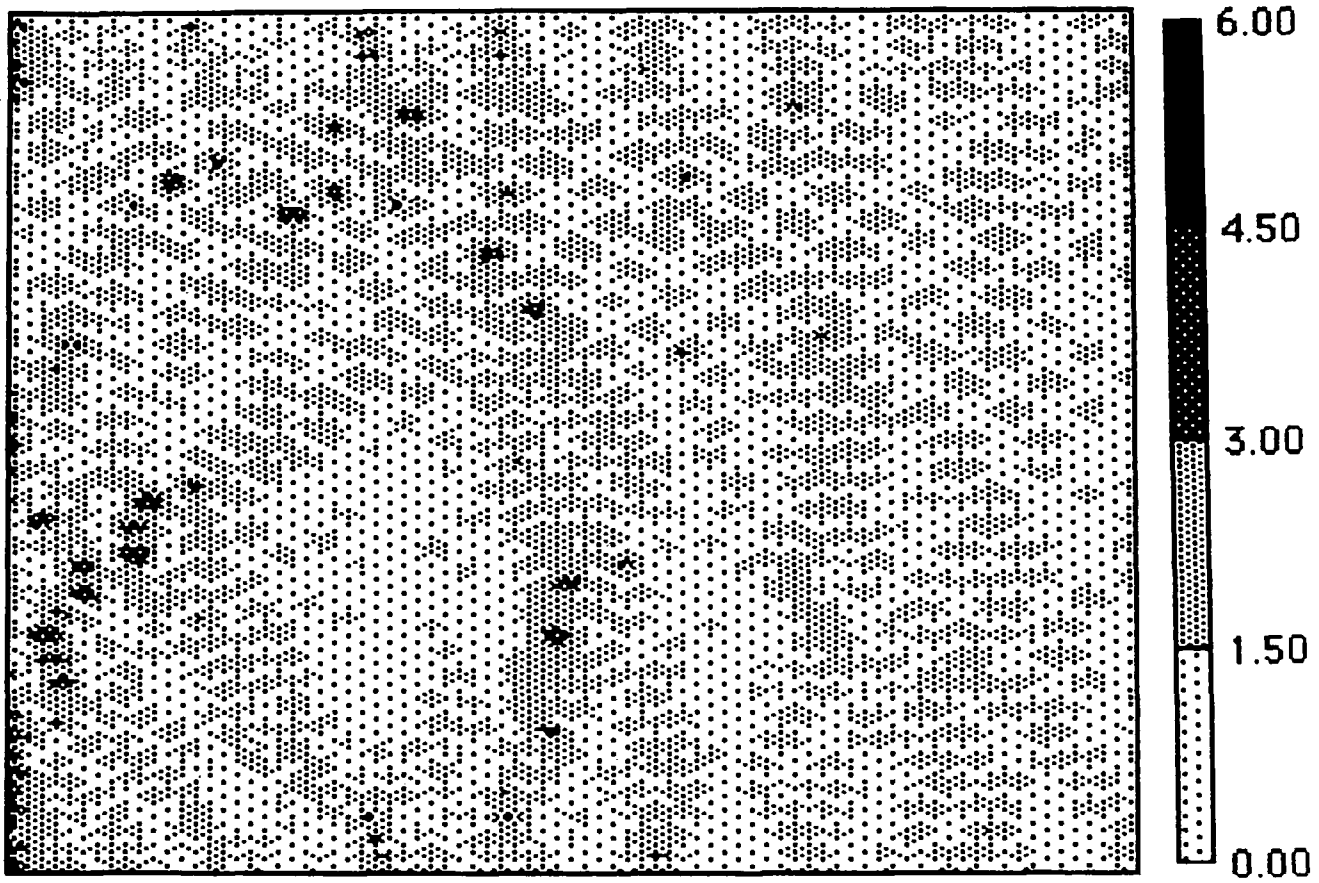


Fig. 5. Computed wave heights for bathymetry of Fig. 2 with a seawall

5. SUMMARY AND CONCLUDING REMARKS

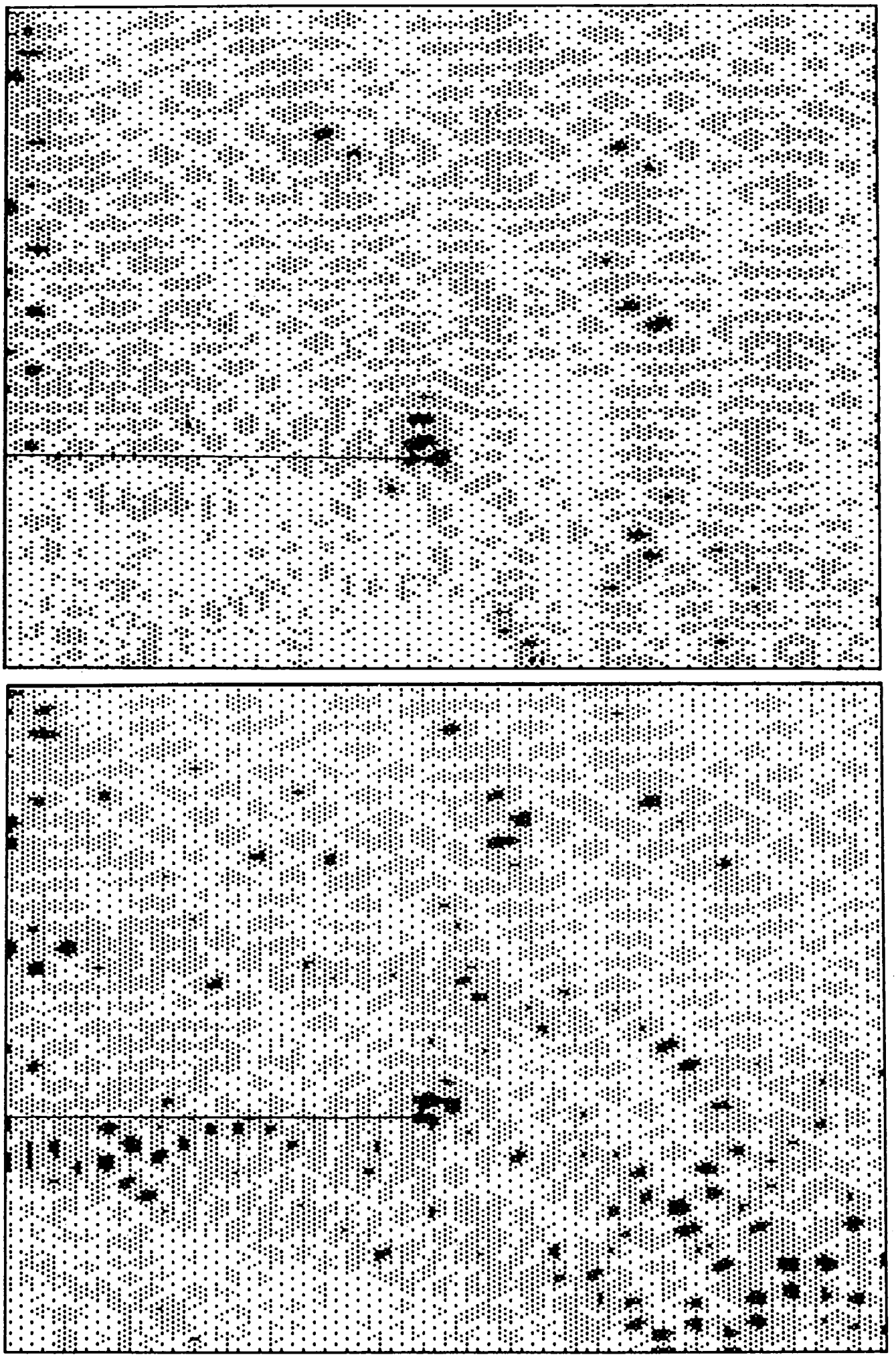
The usefulness of the mild-slope or combined refraction–diffraction equation in producing good simulations of wave behaviour in a wide variety of situations has been demonstrated by many investigators. Since it eliminates the limitations of conventional refraction analyses (viz. existence of caustics or sometimes a paucity of wave rays in regions known to have appreciable wave energy) and since it can be used for a wide range of ocean wave frequencies, its development may be considered to be a major advance in coastal engineering.

One disadvantage of the mild-slope equation is that until now it could be solved (with no physical approximations) only for small coastal domains (roughly 10 wavelengths). Prior to this study, the largest problem tackled with this equation contained 10,000 nodes and involved considerable difficulty[3]. This translates (for a square domain) to only 100 nodes in each direction. The difficulty stems from the exorbitant storage requirements associated with the solution of the elliptic governing equation by direct (non-iterative) methods.

For larger regions, approximate models[13–15,41] can be used when appropriate; these avert computational difficulties, but are applicable only if reflections by coastlines, seawalls, islands, and bathymetry are negligible, and/or the wave propagation exhibits paraxiality. These models are therefore inappropriate in several situations. For example, in coastal regions of the Gulf of Maine, the coastlines are rocky and irregular, the bathymetry is complex, and there are internal boundaries (rocky islands). The approximate models frequently cannot be used; on the other hand, the complete (non-approximate) problem is difficult because the domains of interest are large. The need for alternative models was also highlighted in a recent study by Panchang *et al.*[42], where the mild-slope equation was shown to yield realistic simulations of irregular sea-states. However, the approximate model used in that study is not sufficiently general to simulate the response to the different components that comprise the incident wave spectrum (depending on the bathymetry, domain geometry, the input spectrum, etc.), and it would be cumbersome to use different models for different components.

In this paper, two algorithms are presented by which the problem can be solved on large coastal domains and with no physical approximations. Storage problems that beset conventional models are completely eliminated through the development of the CG iterative algorithms applied to the Gauss-transformed equation. Algorithm 1 is simple, easy to program, and can be conveniently applied to non-rectangular regions. It is, however, slow. The preconditioned algorithm 2, on the other hand, is vastly superior in terms of its rate of convergence. Unlike direct methods, these algorithms require little preprocessing, and a minimum of computer storage. They are,

Fig. 6. Contour diagram of wave heights on the same domain as for Fig. 5, but with a reflective breakwater inserted half-way across the domain: (a) breakwater at $y = -5$ *m, (b) breakwater at* $y = -4.4$ *m. (See Fig. 5 for legend)*



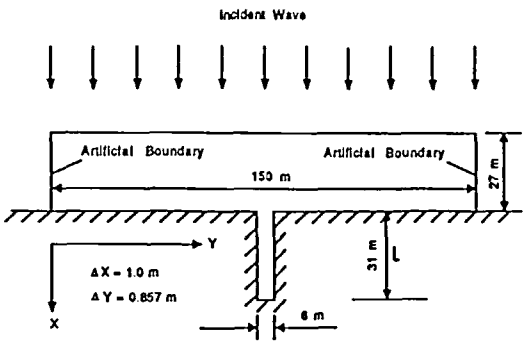


Fig. 7. Nonrectangular model for harbor resonance simulation

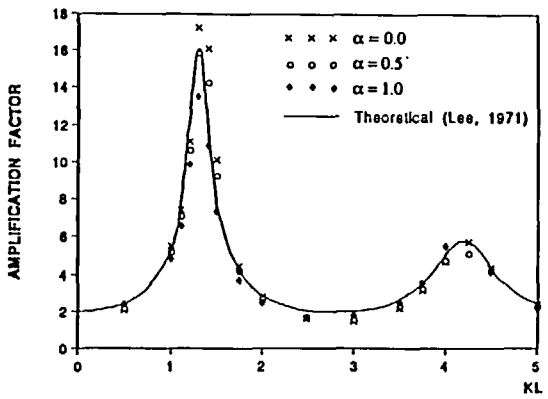


Fig. 8. Wave height comparison for harbor resonance

furthermore, guaranteed to converge. Section 4 also demonstrated that these methods are amenable to the inclusion of nonlinear effects. Prior to this, such effects have been included only in the framework of the parabolic approximation of the mild-slope equation[37,38,43]. For incorporating nonlinearities in the elliptic model, the iterative approach is particularly efficient, since each successive round requires less effort; this is further benefit relative to the direct method.

Using the iterative procedures described in Section 3, coastal wave propagation problems hitherto considered too large (e.g. 44,000 grids) have been successfully solved with reasonable computational effort. The procedures were applied to several configurations of bottom topography, with and without significant reflections, and for rectangular and non-rectangular domains. The results compared extremely well with laboratory data and other mathematical solutions. The insertion of a breakwater and the variation of its position illustrated the high degree of sensitivity of the wave energy distribution to reflection,

Table 1. Performance of iterative algorithms for bathymetry from Berkhoff et al. *(1982)*

Algorithm	ω	# of iterations	ε	CPU time (sec)
1		16,800	0.97×10^{-7}	2769
2	1.0	2,000	0.21×10^{-6}	1266
2	1.2	2,000	0.69×10^{-7}	1264
2	1.4	2,000	0.24×10^{-7}	1257
2	1.6	2,000	0.22×10^{-6}	1270
2	1.8	2,000	0.13×10^{-4}	1267

Table 2. Performance of iterative algorithms for bathymetry from Berkhoff et al. *(1982) in the presence of a seawall*

Algorithm	ω	# of iterations	ε	CPU time (min)
1		49,200	0.99×10^{-7}	140
2	1.0	5,500	0.33×10^{-5}	60
2	1.2	5,500	0.71×10^{-7}	60
2	1.4	5,500	0.81×10^{-7}	60
2	1.6	5,500	0.21×10^{-5}	60
2	1.8	5,500	0.49×10^{-4}	60

Table 3. Performance of iterative algorithms for harbor resonance problem

Algorithm	ω	# of iterations	ε	CPU time (sec)
1		3,262	1×10^{-10}	392
2	1.00	747	1×10^{-10}	131
2	1.25	494	1×10^{-10}	87
2	1.50	337	1×10^{-10}	60
2	1.75	255	1×10^{-10}	46
2	1.80	266	1×10^{-10}	48

diffraction, and resonance. Hence, it is imperative for the coastal engineer, always interested in the worst-case scenario, that many experiments be performed by varying the model geometry. The present algorithm provides the needed versatility. The procedures are therefore useful for estimating wave heights, and complement the existing suite of wave models used by engineers for various applications. Moreover, the iterative procedures can easily be generalized to include modifications of the mild-slope equation that include bottom dissipation[39], currents[44], or a partial relaxation of the 'mild-slope' requirements[25]. Finally, the basic iterative procedures are not specific to the finite–difference technique and can be used in conjunction with finite elements for better modelling complicated coastal boundaries and harbors.

ACKNOWLEDGEMENTS

This research was sponsored in part by NOAA, Department of Commerce, under Grant # NA86-AA-D-SG-047 (R/CE–181), through the Maine Sea Grant Program. G. W. and B. C. R. would also like to acknowledge research support received from the University of Maine Centre for Marine Studies and the Office of Naval Research (Grant N00014-89-J-1574), respectively. We gratefully acknowledge computing resources provided by the University of Maine Computer Center and the Cornell National Supercomputer Facility (of the Cornell Theory Center which receives major funding from NSF and IBM Corporation).

REFERENCES

1 Berkhoff, J. C. W. Computation of combined refraction–diffraction, *Proc. 13th International Conference Coastal Engng.*, Vancouver, 1972

2 Berkhoff, J. C. W. *Mathematical Models for Simple Harmonic Linear Water Waves. Wave Refraction and Diffraction*, Publ. 163, Delft Hydraulics Laboratory

3 Houston, J. R. Combined refraction and diffraction of short waves using the finite element method, *Applied Ocean Research*, 1981, 3(4), 163–170

4 Jonsson, I. G., Skovgaard, O. and Brink-Kjaer, O. Diffraction and refraction calculations for waves incident on an island, *J. Marine Research*, 1976, 343, 469–496

5 Kirby, J. T. and Dalrymple, R. A. Modeling waves in surfzones and around islands, *J. Waterway, Port, Coastal and Ocean Engng*, 1986, **112**(1), 78–93

6 Tsay, T.-K. and Liu, P. L.-F. A finite element model for wave refraction and diffraction, *Applied Ocean Research*, 1983, **5**(1), 30–37

7 Dalrymple, R. A., Kirby, J. T. and Hwang, P. A. Wave diffraction due to areas of high energy dissipation, *J. Waterway, Port, Coastal and Ocean Engng.*, 1984, **110**(1), 67–79

8 Pearce, B. R. and Panchang, V. G. A method for the investigation of steady state wave spectra in bays, *J. Waterway, Port, Coastal and Ocean Engng.*, 1985, **111**(4), 629–644

9 Pos, J. D. and Kilner, F. A. Breakwater gap wave diffraction: An experimental and numerical study, *J. Waterway, Port, Coastal and Ocean, Engng.*, 1987, **113**(1), 1–21

10 Ebersole, B. A., Cialone, M. A. and Prater, M. D. *Regional Coastal Processes Numerical Modeling System; Report 1: RCPWAVE – A Linear Wave Propagation Model for Engineering Use*, TR CERC-86-4, Coastal Engng Research Center, WES, Vicksburg, Miss 39180, 1986

11 Kirby, J. T. Higher-order approximations in the parabolic equation method for water waves, *J. Geophys. Res.*, 1986, **91**, C1, 933–952

12 Radder, A. C. On the parabolic equation method for water-wave propagation, *J. Fluid Mech.*, 1979, **95**, 159–176

13 Tsay, T.-K. and Liu, P. L.-F. Numerical solution of water-wave refraction and diffraction problems in the parabolic approximation, *J. Geophys. Research*, 1982, **87**, C10

14 Ebersole, B. A. Refraction–diffraction model for linear water waves, *J. Waterway, Port, Coastal and Ocean Engng.*, 1985, **111**(6)

15 Panchang, V. G., Cushman-Roisin, B. and Pearce, B. R. Combined refraction – diffraction of short waves for large domains, *Coastal Engng.*, 1988, **12**, 133–156

16 Copeland, G. J. M. A practical alternative to the 'mild-slope' wave equation, *Coastal Engng.*, 1985, **9**, 125–149

17 Madsen, P. A. and Larsen, J. An efficient finite–difference approach to the mild-slope equation, *Coastal Engng.*, 1987, **11**, 329–351

18 Panchang, V. G. and Kopriva, D. A. Solution to two-dimensional water-wave propagation problems by Chebyshev collocation, *Mathematical and Computer Modelling*, 1989, **12**(6), 625–640

19 Rygg, O. B. Nonlinear refraction–diffraction of surface waves in intermediate and shallow water, *Coastal Engng.*, 1988, **12**, 191–211

20 Dong, P. and Al-Mashouk, M. Comparison of transient and steady state wave models for harbor resonance, *Hydraulic and Environmental Modeling of Coastal, Estuarine and River Waters*, Falconer, R. A., Goodwin, P. and Matthew, R. G. S. (eds.), Gower Publishing Co., Brookefield, VT, USA, 1989

21 Bayliss, Q., Goldstein, C. I. and Turkel, E. An iterative method for the Helmholtz equation, *J. Comput. Phys.*, 1983, **49**, 443–456

22 Smith, R. and Sprinks, T. Scattering of surface waves by a conical island, *J. Fluid Mech.*, 1975, **72**, Part 2, 373–384

23 Booij, N. A note on the accuracy of the mild-slope equation, *Coastal Engng.*, 1983, **9**, 191–203

24 Berkhoff, J. C. W., Booy, N. and Radder, A. C. Verification of numerical wave propagation models for simple harmonic linear water waves, *Coastal Engng.*, **6**, 255–279

25 Tsay, T.-K., Zhu, W. and Liu, P. L.-F. A finite element model for wave refraction, diffraction, reflection and dissipation, *Applied Ocean Research*, 1989, **11**, 33–38

26 Kirby, J. T. Open boundary conditions in the parabolic equation method, *J. Waterway, Port, Coastal and Ocean Eng.*, 1986, **112**(3), 460–465

27 Kirby, J. T. A note on parabolic radiation boundary conditions for elliptic wave calculations, *Coastal Engng.*, 1989, **13**, 211–218

28 Goult, R. J., Hoskins, R. F., Milner, J. A. and Pratt, M. J. *Computational Methods in Linear Algebra*, John Wiley, New York, 1975

29 Kershaw, D. S. The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations, *J. Comput. Phys.*, 1978, **26**, 43–65

30 Khosla, P. K. and Rubin, S. G. A conjugate gradient iterative method, *Comput. Fluids*, 1981, **9**, 109–122

31 Peyret, R. and Taylor, T. D. *Computational Methods for Fluid Flow*, Springer-Verlag, New York, 1983

32 Beckman, F. S. The solution of linear equations by the conjugate gradient method, *Mathematical Methods for Digital Computers*, Ralston, A. and Wilf, H. S. (eds.) John Wiley, New York, 1960

33 Axelsson, O., Brinkemper, S. and Il'in, V. P. On some versions of incomplete block-matrix factorization iterative methods, *Linear Algebra and its Applications*, 1984, **58**, 3–15

34 Ito, T. and Tanimoto, K. A method of numerical analysis of wave propatation–application to wave diffraction and refraction, *Proc. 13th Int. Coast. Engng. Conf.*, Vancouver, Canada, 1972

35 Vincent, C. L. and Briggs, M. J. Refraction–diffraction of irregular waves over a mound. *J. Wtrway., Port, Coast. and Oc. Engng.*, 1989, **115**(2), 369–284

36 Kostense, J. K., Dingemans, M. W. and Van den Bosch, P. Wave–current interaction in harbours, *Proc. 21st International Conference Coastal Engng.*, Malaga, Spain, 1988

37 Kirby, J. T. and Dalrymple, R. A. Verification of a parabolic equation for propagation of weakly-nonlinear waves, *Coastal Engineering*, 1984, **8**, 219–232

38 Kirby, J. T. and Dalrymple, R. A. An approximate model for nonlinear dispersion in monochromatic wave propagation models, *Coastal Engineering*, 1986, **9**, 545–561

39 Chen, H. S. and Houston, J. R. *Calculation of Water Level Oscillation in Coastal Harbors*, Instructional Rept CERC-87-2, Coastal Engng Research Center, WES, Vicksburg, Miss 29180, 1987

40 Lee, J-J. Wave – induced oscillations in harbors of arbitrary geometry, *J. Fluid Mech.*, 1976, **45**, Part 2, 375–394

41 Kirby, J. T. Parabolic wave computations in non-orthogonal coordinate systems, J. Waterway, Port, Coastal and Ocean Engng., 1988, **114**, 673–685

42 Panchang, V. G., Ge, W., Pearce, B. R. and Briggs, M. J. Numerical simulation of irregular wave propagation over a shoal, *Jnl. Waterway, Port, Coastal and Ocean Engineering*, 1990, **116**(3), 324–340

43 Munasinghe, L. C. and Dalrymple, R. A. Split-step Fourier algorithm for water waves, *Jnl. Engineering Mechanics, ASCE*, 1990, **116**(2), 251–267

44 Kirby, J. T. A note on linear surface wave-current interaction over slowly varying topography, *J. Geophys. Res.* 1984, **89**(C1), 745–747

APPENDIX A

Algorithm 1 applied to equation (15) yields:

$$r_0 = f' - A'\phi'_0 = Q^{-1}(f - A\phi_0);$$

$$p_0 = A'^* r_0 = Q^{-1}A^*Q^{-T}r_0$$

$$\phi'_{i+1} = \phi'_i + \alpha_i p_i \Rightarrow \phi_{i+1} = \phi_i + \alpha_i Q^{-T}p_i;$$

$$r_{i+1} = r_i - \alpha_i A' p_i = r_i - \alpha_i Q^{-1}AQ^{-T}p_i$$

$$p_{i+1} = A'^* r_{i+1} + \beta_i p_i = Q^{-1}A^*Q^{-T}r_{i+1} + \beta_i p_i$$

where

$$\alpha_i = \frac{|A'^* r_i|^2}{|A' p_i|^2} = \frac{|Q^{-1}A^*Q^{-T}r_i|^2}{|Q^{-1}AQ^{-T}p_i|^2}$$

and a similar expression for β.

Creating new variables $s_i = Qr_i$ and $P_i = Q^{-T}p_i$, and recalling (16),

$$s_0 = f - A\phi_0;$$

$$P_0 = M^{-1}A^*M^{-1}s_0$$

$$\phi_{i+1} = \phi_i + \alpha_i P_i;$$

$$s_{i+1} = s_i - \alpha_i A P_i$$

$$P_{i+1} = M^{-1}A^*M^{-1}s_{i+1} + \beta_i P_i$$

where

$$\alpha_i = \frac{|Q^{-1}A^*M^{-1}s_i|^2}{|Q^{-1}AP_i|^2} \text{ and } \beta_i = \frac{|Q^{-1}A^*M^{-1}s_{i+1}|^2}{|Q^{-1}A^*M^{-1}s_i|^2}$$

Further simplification can be obtained if, as suggested by Bayliss *et al.*[21], we introduce the vector

$$R_i = A^*M^{-1}s_i$$

so that

$$\alpha_i = \frac{|Q^{-1}R_i|^2}{|Q^{-1}AP_i|^2} \text{ and } \beta_i = \frac{|Q^{-1}R_{i+1}|^2}{|Q^{-1}R_i|^2}$$

Finally, Q^{-1} can be eliminated from the α and β by recognizing that for some vector T,

$$|Q^{-1}T|^2 = (Q^{-1}T, Q^{-1}T) = (T, Q^{-T}Q^{-1}T) = (T, M^{-1}T)$$

which is the formulation used in Section 3.