



ELSEVIER

Coastal Engineering 46 (2002) 89–107

**Coastal
Engineering**
An International Journal for Coastal,
Harbour and Offshore Engineers

www.elsevier.com/locate/coastaleng

Modeling wave runup with depth-integrated equations

Patrick J. Lynett, Tso-Ren Wu, Philip L.-F. Liu*

*Joseph DeFrees Hydraulics Lab., School of Civil and Environmental Engineering, Cornell University,
223 Hollister Hall, Ithaca, NY 14853, USA*

Received 13 June 2001; received in revised form 14 March 2002; accepted 27 March 2002

Abstract

In this paper, a moving boundary technique is developed to investigate wave runup and rundown with depth-integrated equations. Highly nonlinear and weakly dispersive equations are solved using a high-order finite difference scheme. An eddy viscosity model is adopted for wave breaking so as to investigate breaking wave runup. The moving boundary technique utilizes linear extrapolation through the wet–dry boundary and into the dry region. Nonbreaking and breaking solitary wave runup is accurately predicted by the proposed model, yielding a validation of both the wave breaking parameterization and the moving boundary technique. Two-dimensional wave runup in a parabolic basin and around a conical island is investigated, and agreement with published data is excellent. Finally, the propagation and runup of a solitary wave in a trapezoidal channel is examined. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Wave runup; Breaking waves; Boussinesq equations

1. Introduction

The past decade saw the advent and widespread applications of Boussinesq-type equation models for studying wave propagation in one and two horizontal dimensions. The conventional Boussinesq equations (Peregrine, 1967) had two major limitations: (1) The depth-averaged model poorly described the frequency dispersion of wave propagation in intermediate depths, and (2) the weakly nonlinear assumption limited the largest wave height that could accurately be modeled. The dispersion properties of the conventional Boussinesq equation model have been improved by modifying the dispersive terms (Madsen and Sorensen, 1992) or using a reference velocity at a specified depth (Nwogu, 1993). These techniques yield a set of equa-

tions whose linear dispersion relation can be adjusted such that the resulting intermediate-depth dispersion characteristics are close to the true linear solution. Liu (1994) and Wei et al. (1995) presented a set of highly nonlinear Boussinesq-type equations that not only can be applied to intermediate water depth but also are capable of simulating wave propagation with strong nonlinear interaction. Wei et al. (1995) have also developed a high-order numerical scheme to solve these equations. All of these efforts successfully extended the usage of the Boussinesq-type equation model, such that wave evolution from relatively deep water to the breaking point could be accurately captured.

Wave propagation using Boussinesq-type equations is now well simulated and understood, but the process of runup and rundown is not. Shoreline boundaries may move significantly under the temporal influence of incident waves. A numerical model should be able

* Corresponding author.

E-mail address: pll3@cornell.edu (P.L.-F. Liu).

to take into account such variations correctly in order to obtain realistic flow patterns.

Researchers generally use a fixed grid, finite difference or finite element method to solve the Boussinesq-type equations. Using a fixed grid numerical model to solve a moving boundary problem can lead to difficulties related to the loss of mass conservation and instabilities in the computations (Leendertse, 1987) as a result of imposing discrete fixed increments to the extent of wetting and drying areas (Balzano, 1998). To reduce the computational instabilities near the wet–dry interface, some researchers added bottom friction into the momentum equations. However, a numerical model should be stable even without using bottom friction dissipation.

Zelt (1991) used a Lagrangian form of the Boussinesq-type equations to simulate shoreline movement. This model produced maximum runup values that compared well with experimental values, but the shape of the wave as it traveled up the slope did not compare as favorably. A handful of others have utilized Lagrangian techniques with depth-integrated equation models to simulate a moving shoreline (e.g., Petera and Nassehi, 1996; Gopalakrishnan, 1989). Another treatment of moving boundary problem is employing a slot or permeable-seabed technique (Tao, 1983, 1984). The first application of the permeable slot with a Boussinesq-type model (Madsen et al., 1997) yielded runup errors on the order of 10% of the maximum. Modifications have been made to this permeable slot technique (Kennedy et al., 2000), increasing the accuracy, but it was also shown that the empirical coefficients that govern the technique cannot be universally determined, due to numerical stability problems (Chen et al., 2000).

In this paper, we present a new moving boundary treatment for wave propagation models. The moving boundary algorithm is conceptually simple, easy to implement, and can be employed by different numerical schemes (i.e., finite difference and finite element) utilizing depth-integrated equations. The moving boundary technique utilizes linear extrapolation near the wet–dry boundary, thereby allowing the real boundary location to exist in-between nodal points. The model is compared with the classic Carrier and Greenspan (1958) solution for monochromatic long wave runup on a constant slope. As another one horizontal dimension test, the solitary wave runup

experiments of Synolakis (1986, 1987), which range from nonbreaking to breaking waves, are recreated numerically. To test the accuracy of two horizontal dimension moving boundary problems, three cases are examined: wave oscillations in a parabolic basin, solitary wave interaction with a conical island, and wave evolution in a trapezoidal channel.

2. Model equations and numerical scheme

The model equations to be utilized in this paper are the highly nonlinear, weakly dispersive wave equations, given in dimensional form (e.g., Liu, 1994):

$$\zeta_t + E = 0, \quad \mathbf{u}_{xt} + \mathbf{F} = 0 \quad (1)$$

where

$$E = \nabla \cdot [(h + \zeta)\mathbf{u}_x] - \nabla \cdot \left\{ (h + \zeta) \times \left[\left(\frac{1}{6}(\zeta^2 - \zeta h + h^2) - \frac{1}{2}z_x^2 \right) \nabla(\nabla \cdot \mathbf{u}_x) + \left[\frac{1}{2}(\zeta - h) - z_x \right] \nabla[\nabla \cdot (h\mathbf{u}_x)] \right] \right\} \quad (2)$$

$$\begin{aligned} F = & \mathbf{u}_x \cdot \nabla \mathbf{u}_x + g \nabla \zeta \\ & + \left\{ \frac{1}{2} z_x^2 \nabla(\nabla \cdot \mathbf{u}_{xt}) + z_x \nabla[\nabla \cdot (h\mathbf{u}_{xt})] \right\} \\ & + \{ [\nabla \cdot (h\mathbf{u}_x)] \nabla[\nabla \cdot (h\mathbf{u}_x)] - \nabla[\zeta(\nabla \cdot (h\mathbf{u}_x))] \\ & + (\mathbf{u}_x \cdot \nabla z_x) \nabla[\nabla \cdot (h\mathbf{u}_x)] \} \\ & + \left\{ z_x \nabla[\mathbf{u}_x \cdot \nabla(\nabla \cdot (h\mathbf{u}_x))] \right. \\ & \left. + z_x(\mathbf{u}_x \cdot \nabla z_x) \nabla(\nabla \cdot \mathbf{u}_x) + \frac{z_x^2}{2} \nabla[\mathbf{u}_x \cdot \nabla(\nabla \cdot \mathbf{u}_x)] \right\} \\ & + \nabla \left\{ -\frac{\zeta^2}{2} \nabla \cdot \mathbf{u}_{xt} - \zeta \mathbf{u}_x \cdot \nabla[\nabla \cdot (h\mathbf{u}_x)] + \right. \\ & \left. + \zeta[\nabla \cdot (h\mathbf{u}_x)] \nabla \cdot \mathbf{u}_x \right\} \\ & + \nabla \left\{ \frac{\zeta^2}{2} [(\nabla \cdot \mathbf{u}_x)^2 - \mathbf{u}_x \cdot \nabla(\nabla \cdot \mathbf{u}_x)] \right\} \quad (3) \end{aligned}$$

where ζ is the free surface elevation, h is the local water depth, and $\mathbf{u}_x = (u_x, v_x)$ is the reference horizontal

velocity. The velocity is evaluated at the elevation $z_\alpha = -0.531h$, as recommended by Nwogu (1993), based on optimum agreement of the governing equations with the linear dispersion relation. Wherever h is negative (initially dry land), this relationship is set to $z_\alpha = -h$, so as to avoid the evaluation of \mathbf{u}_α under the seafloor. Two dimensionless, characteristic coefficients can be given as

$$\varepsilon = a/h, \quad \mu = h/\lambda \quad (4)$$

where a is the wave amplitude and λ is the horizontal length scale. ε is indicative of the importance of nonlinearity and is assumed to be $O(1)$ in Eqs. (1)–(3); μ represents frequency dispersion and $O(\mu^2) \ll 1$. The order of magnitude of accuracy of these equations is $O(\mu^4)$. Note that the above momentum equation, Eq. (3), is slightly different from that presented by Wei et al. (1995). This difference is caused by the omission of some $O(\mu^2)$ terms in Wei et al. in their conversion of $1/2 \nabla(\mathbf{u}_\alpha^2)$ to $\mathbf{u}_\alpha \cdot \nabla \mathbf{u}_\alpha$. These neglected terms vanish only if $\nabla \times \mathbf{u}_\alpha = 0$, which, however, does not imply the irrotationality of the flow field. In fact, $\nabla \times \mathbf{u}_\alpha$ is of $O(\mu^2)$. A more mathematically detailed explanation can be found in Hsiao and Liu (in press).

The parameterizations, \mathbf{R}_f and \mathbf{R}_b , account for the effects of bottom friction and wave breaking, respectively. Bottom friction is described in the quadratic form:

$$\mathbf{R}_f = \frac{f}{h + \zeta} \mathbf{u}_\alpha |\mathbf{u}_\alpha| \quad (5)$$

where f is a bottom friction coefficient, typically in the range of 10^{-3} – 10^{-2} , depending on the Reynolds number and seafloor condition. To simulate the effects of wave breaking, the eddy viscosity model (Zelt, 1991; Kennedy et al., 2000) is used here. Readers are directed to Kennedy et al. (2000) for a thorough description and validation of the breaking model, and the coefficients and thresholds given therein are used for all the simulations presented in this paper.

The model used for all the simulations in this paper is nicknamed COULWAVE, for Cornell University Long and Intermediate Wave Modeling Package. This model has the ability to simulate a wide range of long wave problems, including interaction with porous coastal structures (Lynett et al., 2000), wave gener-

ation by seafloor movements such as landslides (Lynett and Liu, in press (a)), and internal wave propagation and evolution (Lynett and Liu, in press (b)). The numerical model utilizes a predictor–corrector time-stepping scheme, accurate to (Δt^4) , where Δt is the time step. Similar numerical schemes have been successfully employed by Wei et al. (1995) for modeling surface wave phenomena. Assume now that the numerical simulation is at time n , where all physical values (free surface and velocity) at time n , and previous times, are known. To determine the physical values at the next time step $n + 1$, the explicit predictor is first applied:

$$\zeta_{i,j}^{n+1} = \zeta_{i,j}^n - \frac{\Delta t}{12} (23E_{i,j}^n - 16E_{i,j}^{n-1} + 5E_{i,j}^{n-2}) \quad (6)$$

$$\begin{aligned} (\mathbf{u}_\alpha)_{i,j}^{n+1} = & (\mathbf{u}_\alpha)_{i,j}^n - \frac{\Delta t}{12} (23\mathbf{F}_{i,j}^n - 16\mathbf{F}_{i,j}^{n-1} \\ & + 5\mathbf{F}_{i,j}^{n-2}) \end{aligned} \quad (7)$$

where n represents the time index, i the x -space index, and j the y -space index (x and y make up the horizontal plane). Thus, in order to start a simulation, initial conditions from three time levels are required. Now, with an initial estimate of the physical values at the new time level, the implicit corrector equations can be applied:

$$\begin{aligned} \zeta_{i,j}^{n+1} = & \zeta_{i,j}^n - \frac{\Delta t}{24} (9E_{i,j}^{n+1} + 19E_{i,j}^n - 5E_{i,j}^{n-1} \\ & + E_{i,j}^{n-2}) \end{aligned} \quad (8)$$

$$\begin{aligned} (\mathbf{u}_\alpha)_{i,j}^{n+1} = & (\mathbf{u}_\alpha)_{i,j}^n - \frac{\Delta t}{24} (9\mathbf{F}_{i,j}^{n+1} + 19\mathbf{F}_{i,j}^n \\ & - 5\mathbf{F}_{i,j}^{n-1} + \mathbf{F}_{i,j}^{n-2}) \end{aligned} \quad (9)$$

These equations are solved with Jacobi iteration, so the calculation of E^{n+1} and \mathbf{F}^{n+1} is performed with the free surface and velocity values from the previous iteration. To implement the algorithm, an additional grouping of the mixed space and time derivatives in the dispersive terms is required. Wei et al.’s (1995) paper gives a good description and justification of this grouping procedure.

Spatial differencing in the numerical model employs centered finite differences. All first-order spatial derivatives are differenced with fourth-order ($\Delta x^4 = \Delta y^4$) accurate equations, which are five-point differences. Second-order spatial derivatives are approximated with three-point centered finite difference equations, which are second-order accurate. The second-order spatial derivatives are taken to lower-order accuracy because these derivatives only appear in dispersive terms. The “combined” dispersive-numerical error for the second-order derivatives is $O(\Delta x^2 \mu^2)$, which is less than the error associated with first-order spatial derivatives, $O(\Delta x^4)$, for all the grid spacings and wavelengths modeled in this paper.

3. Moving boundary algorithm

The development of the moving boundary algorithm presented in this paper began with a search for a scheme that allows for the wet–dry boundary to exist at any location, not restrictively at a node on a fixed grid. One method of achieving this is through dynamic regridding, using a Lagrangian approach. Methods such as this have been used in finite difference and finite element nonlinear shallow water (NLSW) and Boussinesq equation models (e.g., Petera and Nassehi, 1996; Zelt, 1991). Lagrangian moving boundary techniques require numerical flexibility, in terms of utilizing constantly changing space and time steps, to be implemented in conjunction with a Eulerian-type model. This flexibility is not present in the current numerical scheme, and is difficult to achieve due to the nature of the required high-order derivatives, and so a different approach is developed in this paper.

Owing to the significant number of derivatives calculated by the numerical model (~ 50 in 2D), it would be advantageous if the moving boundary scheme did not require any sort of special treatment of the derivatives near the wet–dry boundary (i.e., forward, backward, or low-order finite differences). To require, for example, directional differences at the boundary leads to abundant conditional statements, making the programming tedious and the runtime longer. Therefore, the five-point centered finite differences that are employed in the numerical model are desired to be used at all locations, including those

points near the shoreline, where neighboring nodes may be dry. With this in mind, the moving boundary scheme will employ a linear extrapolation of free surface displacement, ζ , and velocity components, u_x and v_x , from the fluid domain, through the wet–dry boundary, and into the dry region. Kowalik and Bang (1987) presented a similar approach of employing a linear extrapolation into the dry region, based on Sielecki and Wurtele’s (1970) earlier developments. Their model uses a leapfrog scheme to approximate the NLSW equations, and is limited to one-dimensional, nonbreaking problems. This paper will attempt to extend this idea to two horizontal dimension-breaking problems, using a high-order numerical model.

An extrapolation through the wet–dry boundary permits this boundary to exist in-between nodal points. Fig. 1 gives a numerical example of how the extrapolation is performed in a one-dimensional problem, showing a solitary wave interacting with a 1:20 slope. The free surface locations that are determined using the governing Eqs. (1)–(3) are shown by the solid line, whereas the linearly extrapolated points are shown by the dots. With extrapolated values of ζ and velocity components in the dry region, solving the model equations at wet nodes can proceed. When solving the model equations, five-point centered differences are employed to approximate the spatial derivatives. Although no derivatives are calculated at dry (extrapolated) points, the physical values of free surface and velocity at these points are used to evaluate derivatives at neighboring wet points. The determination of the location of the wet–dry boundary is performed once per time step, immediately after the predictor step. The moving boundary technique is numerically stable, and does not require any additional dissipative mechanisms.

The first step in the extrapolation boundary method is to determine a nodal boundary dividing an area where the model equations are to be solved (i.e., the wet region) and an area to be extrapolated (i.e., the dry region). The criteria employed to determine this dividing point is dependant on the total water depth, \mathcal{H} , where $\mathcal{H} = h + \zeta$. If $\mathcal{H} > \delta$, where δ is some threshold, the model equations will be applied at the node; otherwise, the physical variables at the node will be extrapolated from a neighboring node. The value of δ should be small; a value of $a_0/50$, where a_0

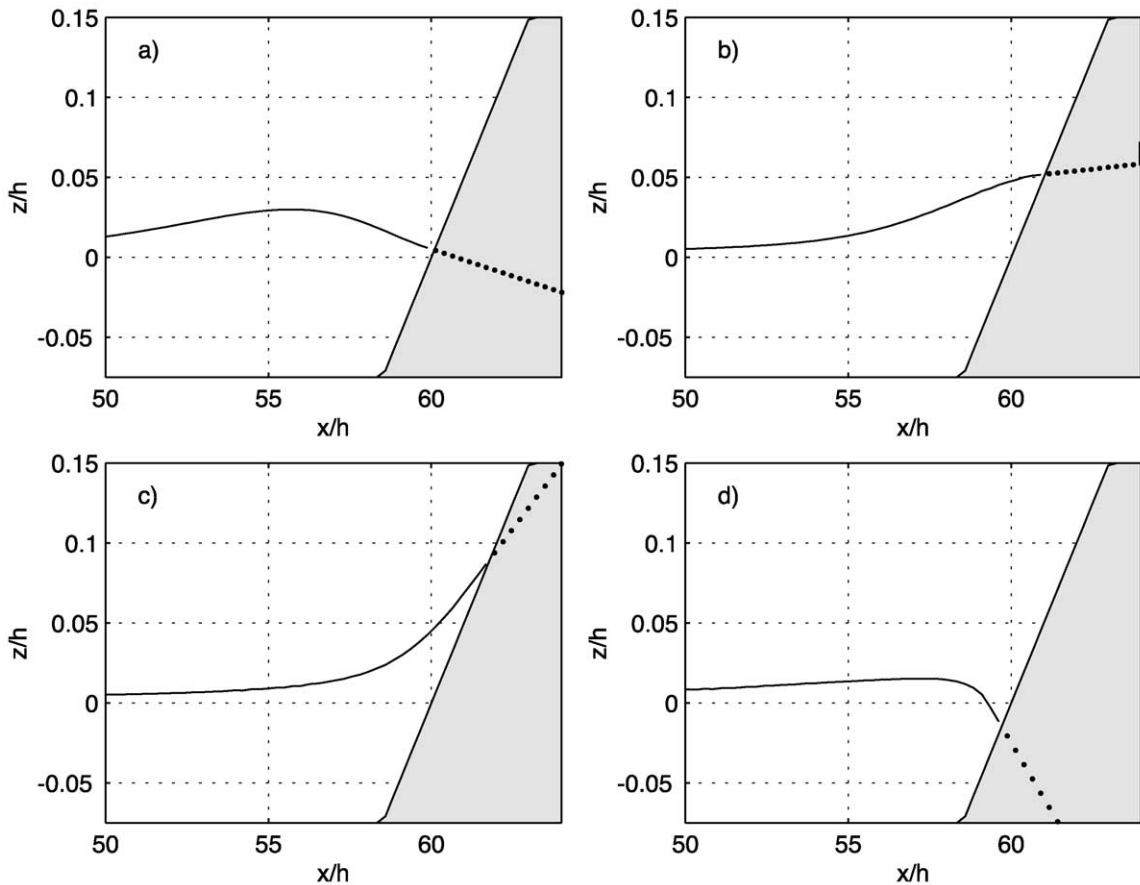


Fig. 1. Runup and rundown of a solitary wave, where extrapolated nodes are shown by the dots.

is the incident wave amplitude, was used for all simulations presented in this paper. This value is chosen based on stability. It was found that decreasing this value could occasionally cause stability problems, especially for simulations with strong wave breaking or ones that include bottom friction. The instability problems associated with these two parameterizations are due to the fact that they are inversely proportional to the total water depth. A very small total water depth may create an equally large dissipative momentum flux, which can lead to an overflow in the iterative numerical scheme. However, for nonbreaking simulations without bottom friction, a δ value of $a_0/5000$ could be stably employed. A convergence check, by changing δ , will be discussed briefly in the next section.

For the simple one-dimensional problem shown in Fig. 1, the extrapolation procedure is straightforward. Using the two wet points (where $\mathcal{H} > \delta$) nearest to the wet–dry boundary, a linear extrapolation into the dry (where $\mathcal{H} < \delta$) region is performed. For the two-dimensional case, the procedure is slightly more complex, but the logic is identical. The 2D extrapolation is performed by checking the surrounding eight points of a dry node. For each surrounding node that is wet, a 1D linear extrapolation is used to estimate the free surface at the dry node. Since more than one surrounding node can be wet, the free surface value at the dry node is taken to be the average of the 1D extrapolations. This procedure is simply repeated for the second layer of dry nodes, extrapolating from the just-extrapolated first layer of dry nodes. For both

1D and 2D cases, a four-point filter is passed over the extrapolated region, smoothing ζ , u_z , and v_z and eliminating possible slope discontinuities in the extrapolation. Additionally, there is one possible arrangement of wet and dry nodes that cannot be allowed to exist. When a wet node is grouped with dry nodes on both sides, i.e., if node i is wet and both $i - 1$ and $i + 1$ are dry, the extrapolation is impossible for both dry nodes. When this situation is developed, the wet node is no longer considered to be in the fluid domain, and its value will be extrapolated.

As the shoreline moves up and down the slope, the number of wet and dry points changes. For example, at time $n - 1$, node $i - 1$ is wet and node i is dry, and its free surface value has been extrapolated. Now, at time n , the new extrapolation for node i yields a total water depth greater than δ . Node i is therefore now a wet node, and its value is no longer extrapolated, but calculated by Eqs. (1)–(3).

As the extrapolated, dry points are solely a function of the neighboring wet points, the finite differences that incorporate these dry points cannot truly be thought of as centered finite differences. Let us consider a one horizontal dimension problem, and focus on six grid points, numbered from $i = -3$ to $i = 2$. At the time that we take a snapshot, the shoreline exists somewhere between points $i = 0$ and $i = 1$. Points to the left of this point are wet ($i = -3, -2, -1, 0$), and points to the right are dry ($i = 1, 2$). At the wet points, the governing equations, using the predictor–corrector scheme, are solved. At the dry points, the free surface and velocity are linearly extrapolated, and can be given as:

$$P_1 = 2P_0 - P_{-1} \quad (10)$$

$$P_2 = 3P_0 - 2P_{-1} \quad (11)$$

where P represents both ζ and u_z , and the subscripts represent the i -index. Substituting the extrapolated values of points $i = 1$ and $i = 2$ into the fourth-order first derivative difference equation:

$$\frac{\partial P_0}{\partial x} = \frac{P_{-2} - 8P_{-1} + 8P_1 - P_2}{12\Delta x} \quad (12)$$

yields, after some manipulation:

$$\frac{\partial P_0}{\partial x} = \frac{1}{6} \left[\frac{\partial P_0}{\partial x} \right]_{2B} + \frac{5}{6} \left[\frac{\partial P_0}{\partial x} \right]_{1B} \quad (13)$$

where $2B$ stands for the second-order backward (or upwind) finite difference:

$$\left[\frac{\partial P_0}{\partial x} \right]_{2B} = \frac{P_{-2} - 4P_{-1} + 3P_0}{2\Delta x} \quad (14)$$

and $1B$ stands for the first-order backward (or upwind) finite difference:

$$\left[\frac{\partial P_0}{\partial x} \right]_{1B} = \frac{-P_{-1} + P_0}{\Delta x} \quad (15)$$

Using the same approach, the derivative at $i = -1$ can be rewritten as:

$$\frac{\partial P_{-1}}{\partial x} = \frac{1}{2} \left[\frac{\partial P_0}{\partial x} \right]_{3T} + \frac{1}{3} \left[\frac{\partial P_0}{\partial x} \right]_{2C} + \frac{1}{6} \left[\frac{\partial P_0}{\partial x} \right]_{1B} \quad (16)$$

where $3T$ stands for the third-order tilted (in the backward direction) finite difference:

$$\left[\frac{\partial P_{-1}}{\partial x} \right]_{3T} = \frac{P_{-3} - 6P_{-2} + 3P_{-1} + 2P_0}{6\Delta x} \quad (17)$$

and $2C$ stands for the second-order centered finite difference:

$$\left[\frac{\partial P_{-1}}{\partial x} \right]_{2C} = \frac{-P_{-2} + P_0}{2\Delta x} \quad (18)$$

So clearly, hidden within the linear extrapolation, is leading order dissipation associated with the upwind differencing, even though a fourth-order centered difference is being taken. Note that the extrapolations are done for both free surface and velocity, so the moving boundary scheme will dissipate both momentum and mass.

The same analysis can be done for the second-order in space derivatives. At the point $i = 0$, the curvature is given as

$$\frac{\partial^2 P_0}{\partial x^2} = \frac{P_{-1} - 2P_0 + P_1}{\Delta x^2} \quad (19)$$

which is, with the linear extrapolation of P_1 , exactly zero at this point. Therefore, at the first wet point, all second-order differences disappear, and the governing Eqs. (1)–(3) reduce to the nonlinear shallow water wave equations. Now, looking at the whole picture of first- and second-order spatial derivatives, we see that numerical dissipation is not as great as it might

appear. It was shown that the first spatial derivative at the first wet point, $\partial P_0/\partial x$, is in large part approximated by the first-order upwind finite difference, $[\partial P_0/\partial x]_{1B}$. The leading numerical truncation error of the upwind difference is $(\Delta x/2)(\partial^2 P_0/\partial x^2)$, which is the source of the numerical dissipation in upwind schemes. However, at this first wet point, the second spatial derivative, $\partial^2 P_0/\partial x^2$, is forced to zero in the numerical model. The leading numerical truncation error of the second spatial derivative taken with a second-order centered difference formula is $(\Delta x^2/12)(\partial^4 P_0/\partial x^4)$. Therefore, the leading order, dissipative truncation error of the upwind difference at the first wet point is actually $(\Delta x/2)(\Delta x^2/12)(\partial^4 P_0/\partial x^4)$. At the second wet point, the first spatial derivative, $\partial P_{-1}/\partial x$, also incorporates upwind differencing (although its importance is five times less here as compared to the first wet point). As the second spatial derivative at this point is non-zero, there will at this point occur dissipation proportional to the second spatial derivative, equal to $(\Delta x/10)(\partial^2 P_{-1}/\partial x^2)$.

It is worth noting that these issues with leading order numerical dissipation associated with the linear extrapolation could be avoided by utilizing a higher-polynomial extrapolation. Unfortunately, these higher-order extrapolations created stability problems with breaking and near-breaking wave runup. As these waves approach the beach, typically the curvature of the free surface is large very near the shoreline. The large curvature created rapidly varying extrapolated values, which then led to numerical roundoff problems.

It would seem to be unnecessary to perform the linear extrapolation in the numerical model, as one could simply code a couple conditional statements, where if the current calculation node in the model is near the wet–dry boundary, use upwind differencing, instead of centered differencing. This too was attempted, but always resulted in $2\Delta x$ waves. It was found, through trial and error, that stability comes from the prediction of velocity in the dry region. In this numerical scheme:

$$(\mathbf{u}_x)^{n+1} = f[(\mathbf{u}_x)^{n+1}, (\mathbf{u}_x)^n, (\mathbf{u}_x)^{n-1}, (\mathbf{u}_x)^{n-2}] \quad (20)$$

as well as a function of numerous other parameters. Let us say that at time n , the point i was dry. Now, at time $n+1$, the point i is wet. What are the previous values of velocity, at times $n-2$, $n-1$, and n , to use

in the predictor–corrector scheme? An answer of zero velocity would be most obvious, because physically, there was no fluid. Using a zero velocity at these times in the numerical model led to $2\Delta x$ waves. So for this type of model, a zero velocity at previous times does not work. Using the linearly extrapolated velocities at the previous times of $n-2$, $n-1$, and n works well. It could be argued that the velocities at the previous “dry” times should not be zero, in fact they should not be anything—they are undefined. Thus, this model is simply taking a reasonable guess at what the undefined velocity should be in order to yield a stable and accurate numerical model. The linear spatial extrapolation is not just important as a spatial extrapolation (which is equivalent to some combination of upwind differencing)—it is especially important as a temporal extrapolation.

As a primary check of the algorithm, its ability to conserve mass is analyzed. Mass is defined as the integral of the free surface elevation, not the integral of the total water depth. A range of solitary waves, from $0.01 < \varepsilon < 0.4$ propagating up one-dimensional slopes of 1:10, 1:20, and 1:50 were checked for conservation of mass. Note that when referencing solitary waves, $\varepsilon = H/h$, where H is the solitary wave height. The solitary wave is generated using the analytic formulas presented in Wei and Kirby (1995), which are derived from the weakly nonlinear, “extended” Boussinesq equations. Fig. 2 summarizes the conservation properties of these cases. Shown in this figure is the fractional change in mass of the soliton, after completely exiting the nearshore region. Thus, these fractions represent the change in mass, scaled by the initial mass, after interaction with the shoreline is over. There are two clear trends: (1) for a given slope, the error in conservation is larger with larger wave heights and (2) for a given wave height, the error is larger with milder slopes. Both of these trends are consistent with the expectation that the numerical error is larger when the curvature near the shoreline is larger.

A question that arises with examination of Fig. 2 is whether the small conservation errors will accumulate during a regular wave simulation, destroying the simulation accuracy. A test case with a sine-wave shoaling up a 1/10 slope was simulated, and is summarized in Fig. 3. The incident wave has a wave height/water depth = 0.1 and a wavelength/water

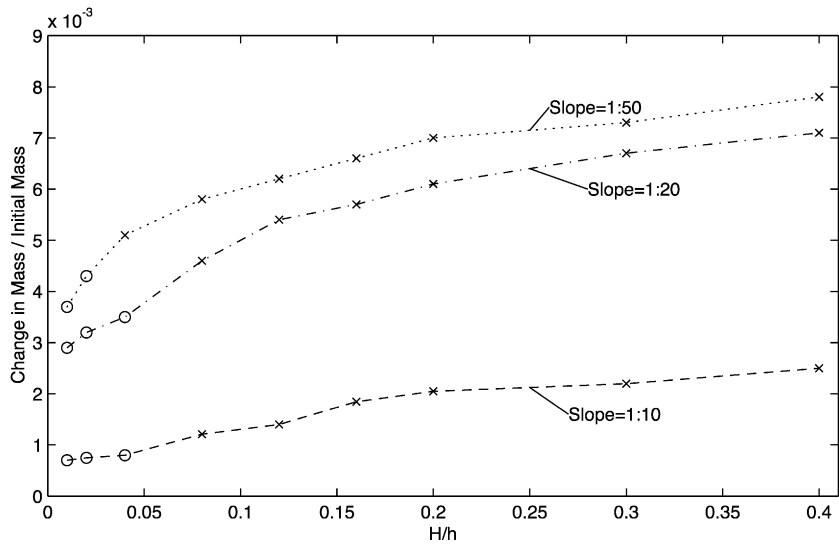


Fig. 2. Fractional change in mass for breaking and nonbreaking solitary waves interacting with three different planar slopes. Simulations where breaking occurs are indicated by the 'x's, nonbreaking results by the 'O's.

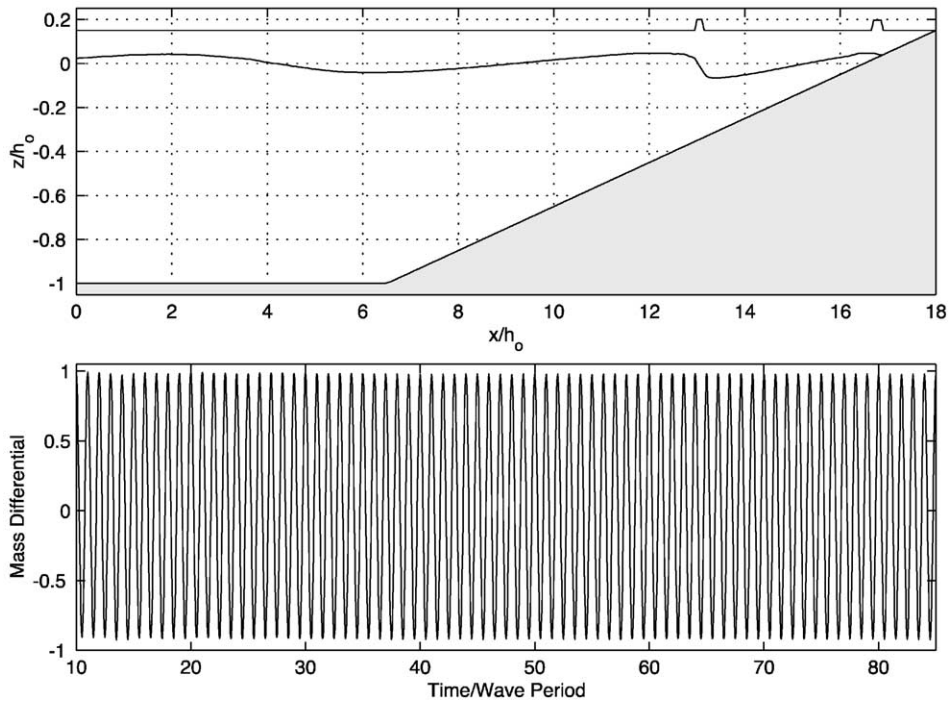


Fig. 3. Mass conservation check for a breaking period wave. The top subplot shows a spatial snapshot of the free surface, roughly 80 wave periods into the simulation, where the locations of breaking are given by the steps in the line plotted on $z/h_0=0.15$. The bottom plot gives the total mass fluctuation in the numerical domain as a function of time.

depth = 10, and breaks strongly while approaching the shoreline. The top plot of Fig. 3 is a snapshot of the free surface and also indicates the breaking locations. Note that a sponge layer is used as the left boundary. The bottom plot of the same figure shows the oscillation of mass in the entire numerical domain through 80 wave periods of time. The oscillation is due to the constant addition/subtraction of mass by the internal source wave generator. Clearly, no accumulation of errors occurs, as the oscillation remains regular for a large number of periods. In regard to this periodic, breaking simulation, numerical filtering was required for long-term stability. Each time a wave breaking event initiated, a small amount of noise was generated immediately behind the breaker. Eventually, this noise accumulated and caused the simulation to become unstable. This statement is not restricted to simulations that contain a shoreline, any simulation with periodic wave breaking suffers from this difficulty. For the setup used in Fig. 3, the simulation would overflow after 15 wave periods. To eliminate this

noise and the associated instability, a nine-point filter (see Kirby et al., 1998) was passed over the domain, filtering both the free surface and velocity, once every two wave periods. The use of the filter has very little effect on conservation, but gives a huge boost to numerical stability, allowing the simulation to run indefinitely. Filtering is only needed for periodic, breaking waves, and thus for all simulation results presented in this paper, except of course for those given in Fig. 3, no filtering is used.

4. Validation in one horizontal dimension

4.1. Sine wave runup

As a first check of the moving boundary model, a monochromatic wave train is let to runup and run-down a plane beach. This situation has an analytic solution derived by Carrier and Greenspan (1958). Their derivation makes use of the NLSW equations,

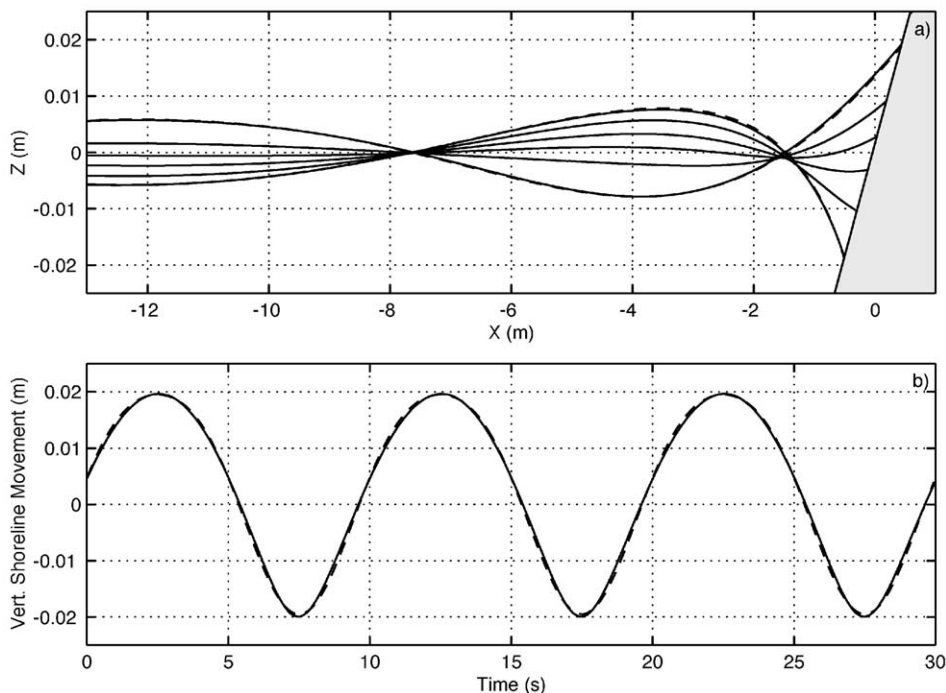


Fig. 4. Sine wave runup on a planar beach. (a) Numerical free surface at various times, analytic free surface is shown by the dashed line (---), and is only compared for the maximum and minimum shoreline movement profiles. (b) Comparison between analytical (---) and numerical (—) shoreline movement.

and thus for consistency the dispersive (μ^2) terms will be ignored in the numerical simulations for this comparison. The wave and slope parameters for this test case are identical to those used by Madsen et al. (1997) and Kennedy et al. (2000). A wave train with height 0.006 m and period of 10 s travels in a one-dimensional channel with a depth of 0.5 m and a slope of 1:25. For the numerical simulation, a grid size of 0.045 m and a time step 0.01 s are used; bottom friction is not included and the wave does not break. It should be noted that the grid size is an order of magnitude smaller than what is required for a convergent solution. This small grid size is used only to make certain that the boundary location travels a significant number of grid points (>10) during runup and rundown.

The results of the numerical simulation are shown in Fig. 4. Fig. 4a shows the numerical free surface at various times, along with two profiles of the analytic free surface. The comparison between analytic and

numerical horizontal shoreline movement is shown as Fig. 4b. The agreement is good. Also, as a check on the convergence properties of δ , an additional simulation with $\delta = a_0/5000$ was run. A comparison between the $\delta = a_0/50$ shows little difference, and is not given in this paper. The maximum deviation in shoreline at any time between the two δ runs is on the order of 0.01% of the maximum excursion.

4.2. Nonbreaking and breaking solitary wave runup

Solitary wave runup and rundown was investigated experimentally by Synolakis (1986, 1987). In his work, dozens of experimental trials were performed, encompassing two orders of magnitude of solitary wave height. The beach slope was kept constant at 1:19.85. Many researchers have used this data set to validate numerical models (e.g., Zelt, 1991; Lin et al., 1999). To compare with this data, solitary waves with heights in the range of $0.005 < \varepsilon < 0.5$ are made to

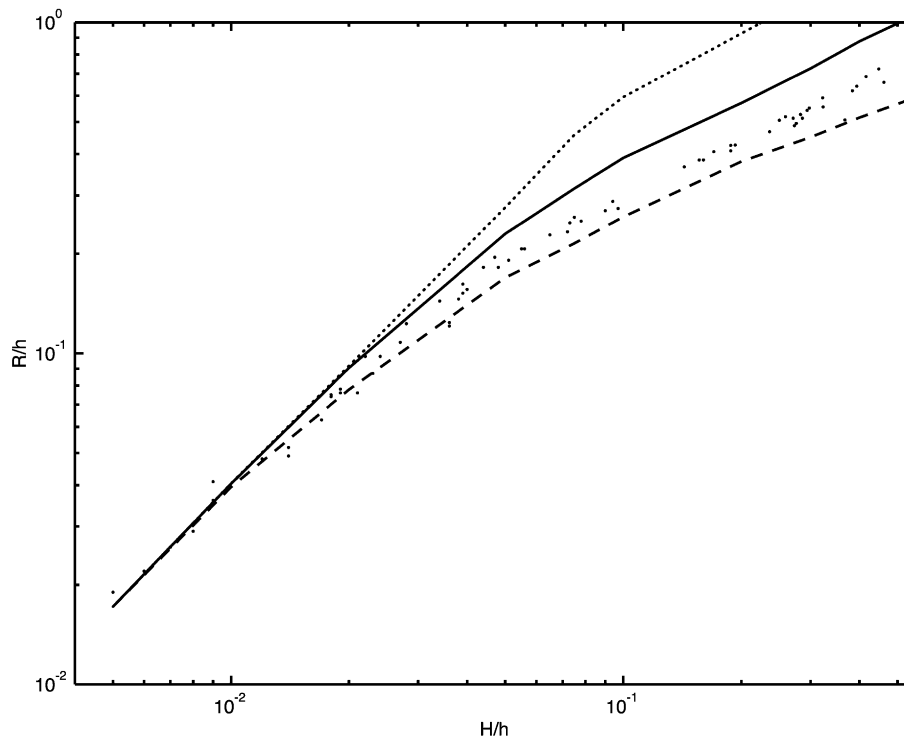


Fig. 5. Nondimensional maximum runup of solitary waves on a 1:19.85 beach versus nondimensional wave height. The points experimental data taken from Synolakis (1986), the dotted line is the numerical result with no bottom friction, the solid line is the numerical result with a bottom friction coefficient, f , of 10^{-3} , and the dashed line with $f = 10^{-2}$.

runup and rundown a slope and the maximum vertical runup is calculated. Note that this range includes both nonbreaking and breaking waves. For all simulations, $\Delta x/h=0.3$ and $\Delta t\sqrt{g/h}=0.03$. As a test of the sensitivity of wave runup to bottom friction, three sets of simulations were undertaken with different bottom friction coefficients, f . Set 1 was run with no

bottom friction, Set 2 with $f=10^{-3}$, and Set 3 with $f=10^{-2}$.

The numerical results are compared with the experimental data in Fig. 5, where maximum vertical runup is scaled by the water depth. For the smallest solitary waves ($\varepsilon < 0.01$) bottom friction does not affect the runup, as maximum runup is identical for

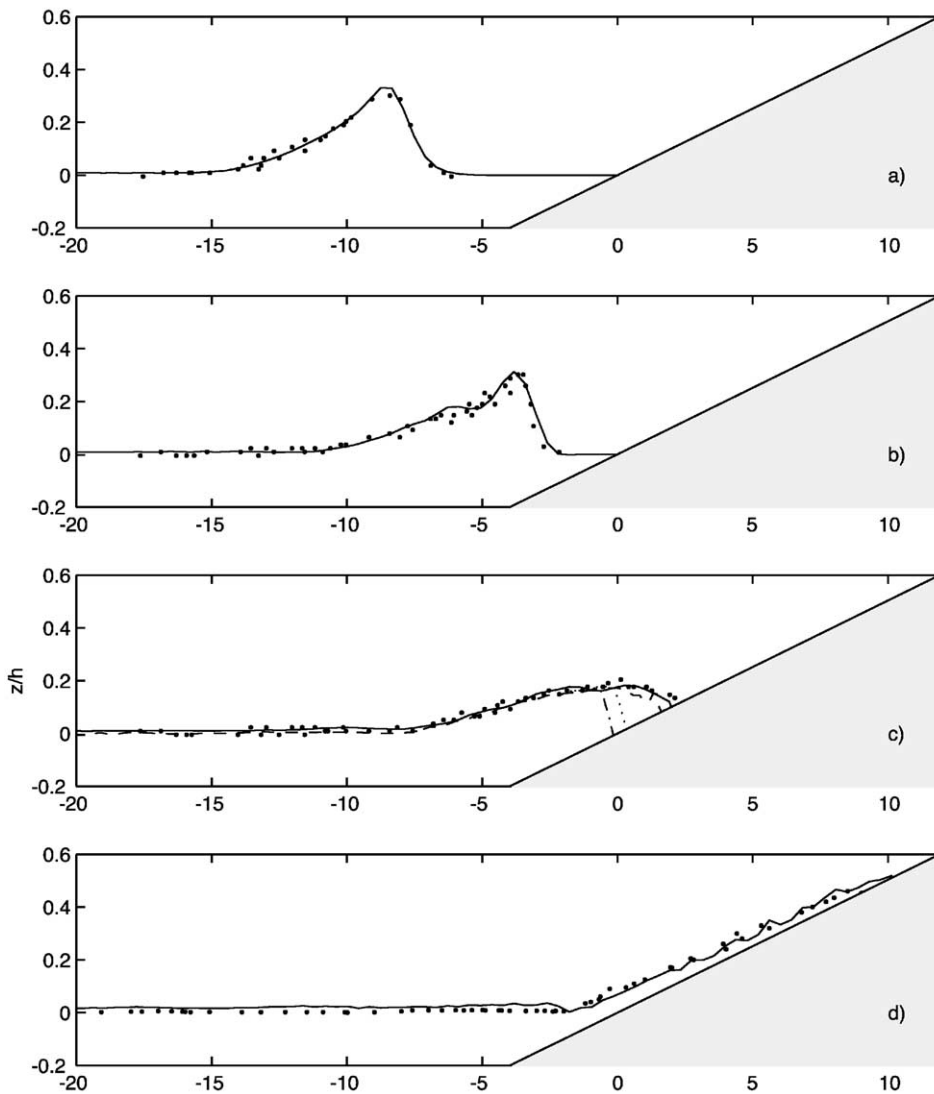


Fig. 6. Breaking solitary wave runup and rundown on a planar beach at $t(g/h)^{1/2}=(a) 15, (b) 20, (c) 25, (d) 45$. The solid line represents the numerical results and the points represent the experimental data. In (c), the dashed line represents numerical results by Lin et al. (1999) (closest to experiment and numerical results presented in this paper), the dotted line represents results by Zelt (1991), and the dashed–dotted line results by Titov and Synolakis (1995).

all three numerical sets. This is consistent with previous research (e.g., Liu et al., 1995), where it is shown that bottom friction effects are minor for non-breaking waves, and will typically alter the runup by <math><0.5\%</math> of the maximum. For larger wave heights, breaking is initiated, both experimentally and numerically, near $\varepsilon=0.04$. It is at this point that the numerical runup for Set 1 and Set 2 begins to diverge. Note that due to the log–log scale used in Fig. 5, the deviation in maximum runup may not be apparent. As an example, for $\varepsilon=0.3$, scaled runup with no bottom friction is 1.21, with $f=10^{-3}$ runup is 0.73, and with $f=10^{-2}$ is 0.45, which are significantly different. Use of $f=5 \times 10^{-3}$ yields the best agreement with experimental data for this particular case.

It would seem that inclusion of an accurate bottom friction parameterization becomes increasingly important with increasing degree of wave breaking. The probable reason is that as a broken wave runs up a mild slope, it travels up the slope as a fairly thin layer of water. As can be seen from Eq. (5), the smaller the total water depth, the more important bottom friction becomes.

Synolakis (1986) also photographed the waves during runup and rundown. One set of these snap-

shots, for $\varepsilon=0.28$, was digitized and compared with the numerical prediction, shown in Fig. 6. The numerical simulation shown in this figure uses $f=10^{-3}$. The wave begins to break between Fig. 6c and b, and the runup/rundown process is shown in Fig. 6c–d. In Fig 6c, numerical snapshots from three other models are plotted. The comparisons indicate a significant improvement over weakly nonlinear Boussinesq equation results of Zelt (1991) and the NLSW results of Titov and Synolakis (1995). Additionally, the numerical results presented in this paper compare favorably to the two dimensional (vertical plane) results of Lin et al. (1999), which makes use of a complex turbulence model.

5. Validation in two horizontal dimensions

5.1. Long wave resonance in a parabolic basin

Analytic solutions exist for few nonlinear, two horizontal dimension problems. One such solution is that for a long wave resonating in a circular parabolic basin. Thacker (1981) presented a solution to the

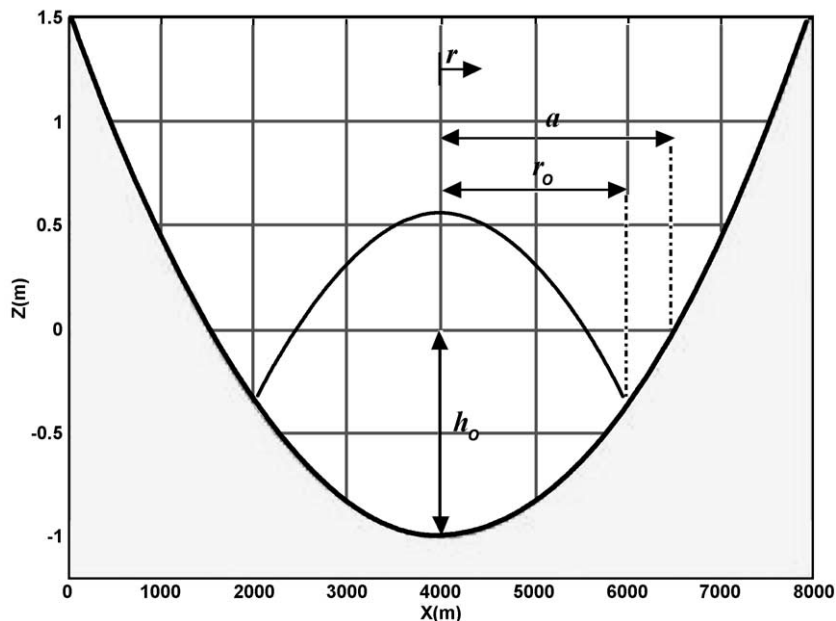


Fig. 7. Initial free surface and depth profile for parabolic basin test.

NLSW equations, where the initial free surface displacement is given as:

$$\zeta(r, t = 0) = h_0 \left[\frac{(1 - A^2)^{1/2}}{1 - A} - 1 - \frac{r^2}{a^2} \left\{ \frac{1 - A^2}{(1 - A^2)^2} - 1 \right\} \right] \quad (21)$$

and the basin shape is given by:

$$h(r) = h_0 \left(1 - \frac{r^2}{a^2} \right) \quad (22)$$

where

$$A = \frac{a^4 - r_0^4}{a^4 + r_0^4}, \quad (23)$$

h_0 is the center point water depth, r is the distance from the center point, a is the distance from the center point to the zero elevation on the shoreline, and r_0 is the distance from the center point to the point where the total water depth is initially zero. The numerical values used for this test are: $h_0 = 1.0$ m, $r_0 = 2000$ m, and $a = 2500$ m. The centerline initial condition and depth profile is shown in Fig. 7. Thacker showed the solution to this problem to be:

$$\zeta(r, t) = h_0 \left[\frac{(1 - A^2)^{1/2}}{1 - A \cos \omega t} - 1 - \frac{r^2}{a^2} \left\{ \frac{1 - A}{(1 - A \cos \omega t)^2} - 1 \right\} \right] \quad (24)$$

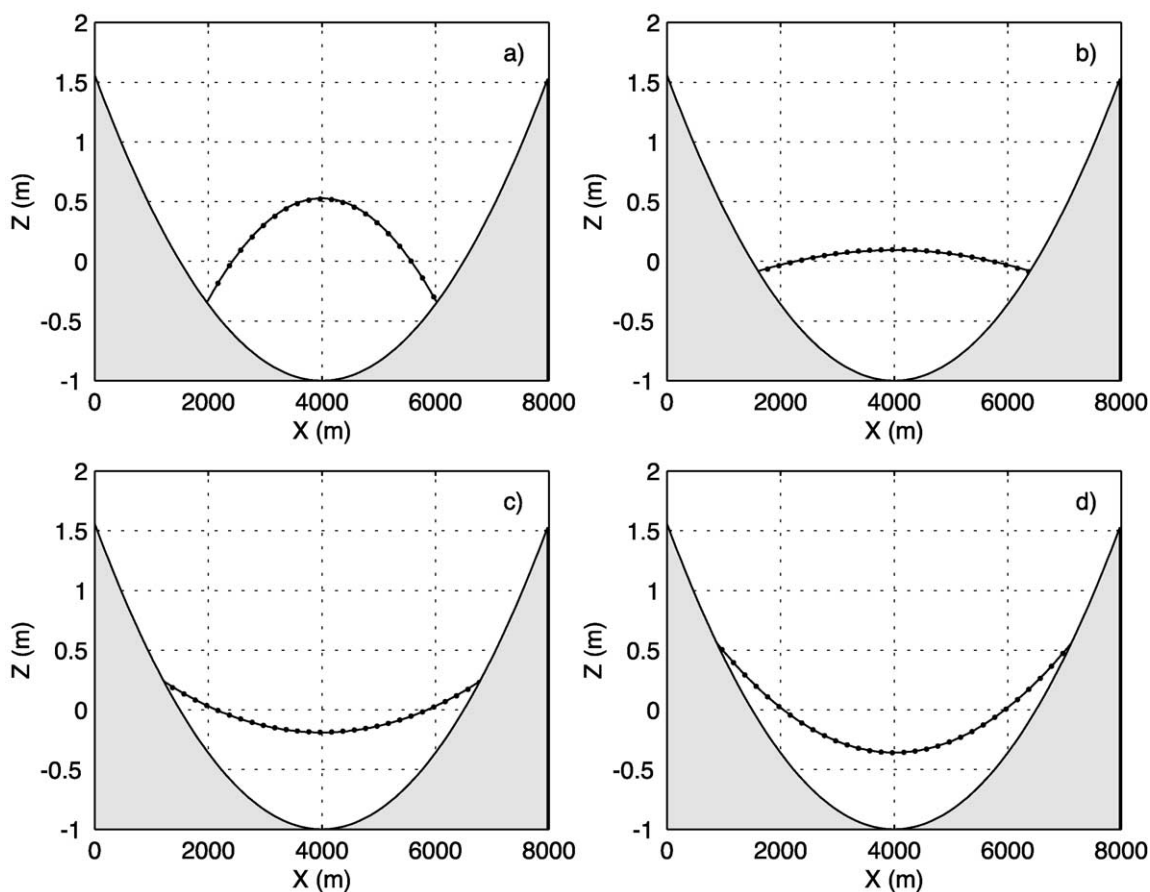


Fig. 8. Centerline free surface profiles for numerical (—) and analytical (· · ·) bowl oscillation solutions at $t = (a) 5T$, (b) $5/6T$, (c) $5/3T$, (d) $5/2T$, where T is the oscillation period.

where

$$w = \frac{1}{a} (8gh_0)^{1/2} \quad (25)$$

and g is gravity. Cho (1995) also used this solution as a test for his NLSW moving boundary model. Cho's model, an explicit leap-frog finite-difference scheme which includes numerical frequency dispersion, reproduced the analytical solution very well for roughly one-half of an oscillation, but began to deviate soon after. A simulation using the extrapolation boundary technique presented in this paper was undertaken, truncating the dispersive terms in Eqs. (2) and (3) to be consistent with the NLSW solution, and using $\Delta x = 28$ m and $\Delta t = 0.9$ s. Bottom friction is not included and the wave does not break. The comparison between the numerical and analytic results is shown in Fig. 8. The numerical free surfaces shown in Fig. 8a–d are from the fifth oscillation, and show excellent agreement with the analytic solution. Additionally, a test using the full equations (Eqs. (2) and (3)), with dispersive terms, was performed. Interestingly, the wave field in this situation becomes chaotic after the first oscillation, and shows a similar pattern of divergence from the analytical solution as Cho's results. Therefore, this parabolic basin comparison would appear to be an ideal test for NLSW models, as the effects of numerical dispersion or dissipation become evident rapidly.

5.2. Runup on a conical island

Briggs et al. (1994) presented a set of experimental data for solitary wave interaction around a conical island. The slope of the island is 1:4 and the water depth is 0.32 m. Three cases were simulated, corresponding to solitary wave heights of 0.013 m ($\varepsilon = 0.04$), 0.028 m ($\varepsilon = 0.09$), and 0.058 m ($\varepsilon = 0.18$). In addition to recording free surface elevation at a half dozen locations, maximum wave runup around the entire island was measured. This data set has been used by several researchers to validate numerical runup models (e.g., Liu et al., 1995; Titov and Synolakis, 1998; Chen et al., 2000). In this paper, free surface elevation is compared at the locations shown in Fig. 9. Gages #6 and #9 are located near the front face of the island, with #9 situated very near the initial shoreline position. Gages

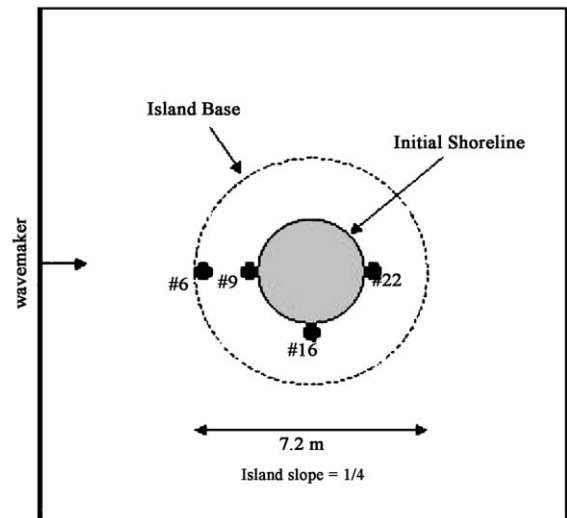


Fig. 9. Conical island setup. The gage locations are shown by the dots, and the wave approaches the island from the left.

#16 and #22 are also located at the initial shoreline, where #16 is on the side of the island and #22 on the back face.

Simulations were performed using $\Delta x = 0.15$ m and $\Delta t = 0.02$ s; bottom friction is neglected for these numerical tests. A soliton is placed in the numerical domain, as an initial condition. Numerical–experimental time series comparisons are shown in Fig. 10. Fig. 10a–d is for Case A ($\varepsilon = 0.04$), Fig. 10e–h is for Case B ($\varepsilon = 0.09$), and Fig. 10i–l is for Case C ($\varepsilon = 0.18$). The gage number is shown in the upper left of each subplot. For all comparisons, the lead wave height and shape is predicted very well with the current model. Also, for all comparisons, the secondary depression wave is not predicted well. The numerical results show less of a depression following the main wave than in the experiments. This deviation is consistent with other runup model tests (e.g., Liu et al., 1995; Chen et al., 2000). The agreement of Case C is especially notable, as the soliton breaks along the backside of the island as the trapped waves intersect. This breaking occurs both experimentally, as discussed in Liu et al. (1995), and numerically.

As mentioned, maximum runup was also experimentally recorded. The vertical runup heights are converted to horizontal runups, scaled by the initial shoreline radius, and plotted in Fig. 11. The crosshairs represents the experimental data, where Fig. 11a is for

Case A, Fig. 11b is for Case B, and Fig. 11c is for Case C. The numerical maximum inundation is also plotted, given by the solid line. The agreement for all cases is very good.

5.3. Soliton evolution in a trapezoidal channel

Peregrine (1969) presented laboratory experiments wherein solitary waves propagated through a trapezoidal channel. To experimentally create the solitons,

a piston wavemaker was cut to fit the channel and could slide horizontally along the trapezoidal channel. In the numerical simulations, as it is difficult to implement a piston wavemaker in a trapezoidal channel, the solutions of solitary waves in rectangular channels are used as an initial condition everywhere in the channel.

Once a solitary wave enters a trapezoidal channel, it deforms. Eventually, in certain channels, the leading wave will reach a quasi steady state, and the wave-

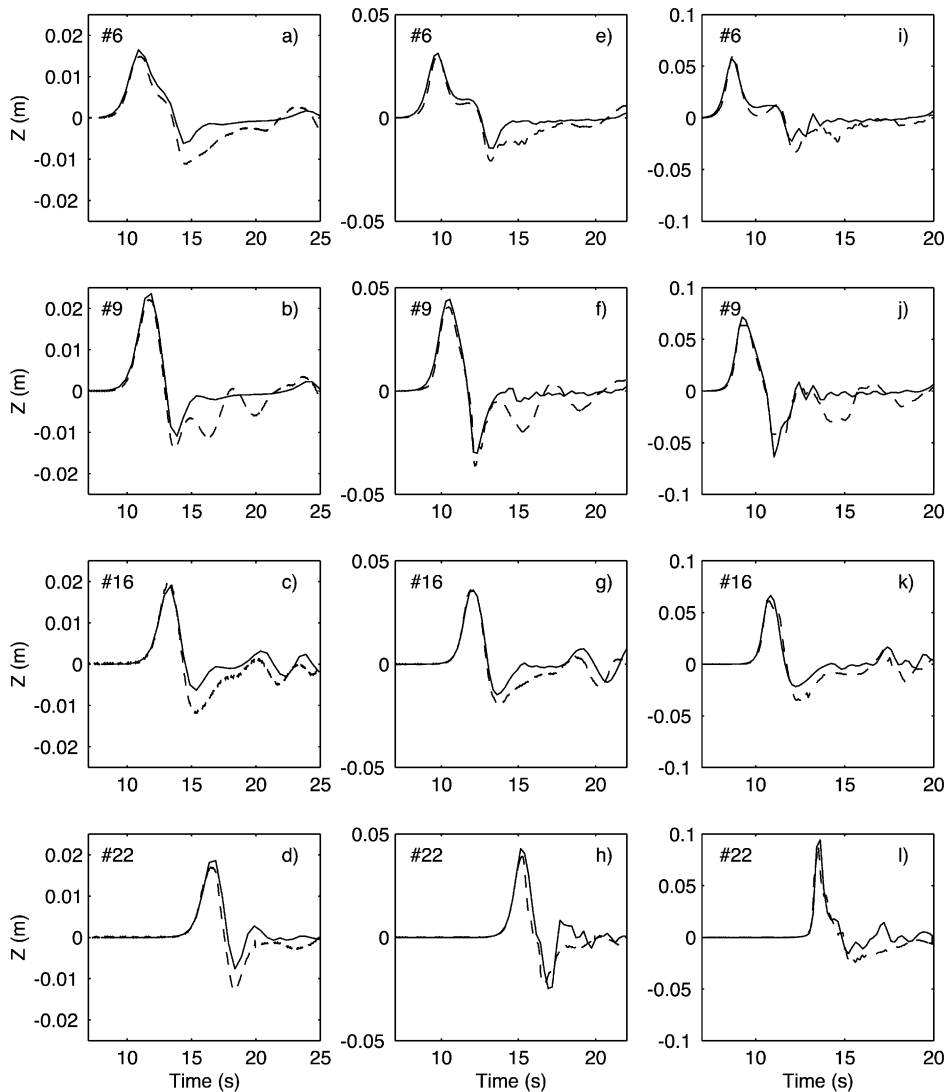


Fig. 10. Experimental (– –) and numerical (–) time series for solitary wave interaction with a conical island. (a–d) are for Case A, (e–h) are for Case B, and (i–l) are for Case C. The gage number is shown in the upper left.

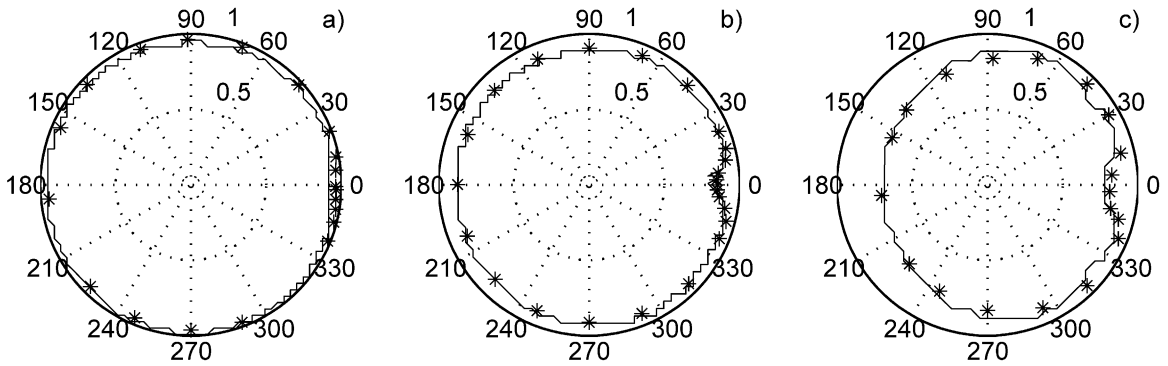


Fig. 11. Maximum horizontal runup, scaled by the initial shoreline radius, for Case A (a), Case B (b), and Case C (c). Experimental values are shown by the stars and the numerical results are shown by the solid line.

form will not change in time. After reaching this quasi steady state, numerical results of the lead wave height are compared with Peregrine’s experimental results. The comparisons are shown in Fig. 12. For this comparison, a trapezoidal half-channel (one vertical wall and one sloping side wall) with constant depth width of $1.5h_0$, where h_0 is the depth at the non-sloping part the channel, and a side-wall slope of 1:1

is employed. Three different amplitude solitary waves ($a = 0.08h_0$, $a = 0.12h_0$, and $a = 0.18h_0$) are simulated and compared with experimental results. The numerical results show reasonably good agreement with laboratory data, although there is a clear trend of under prediction of wave height near the shoreline.

An interesting property of wave evolution in certain trapezoidal channels is the successive regenera-

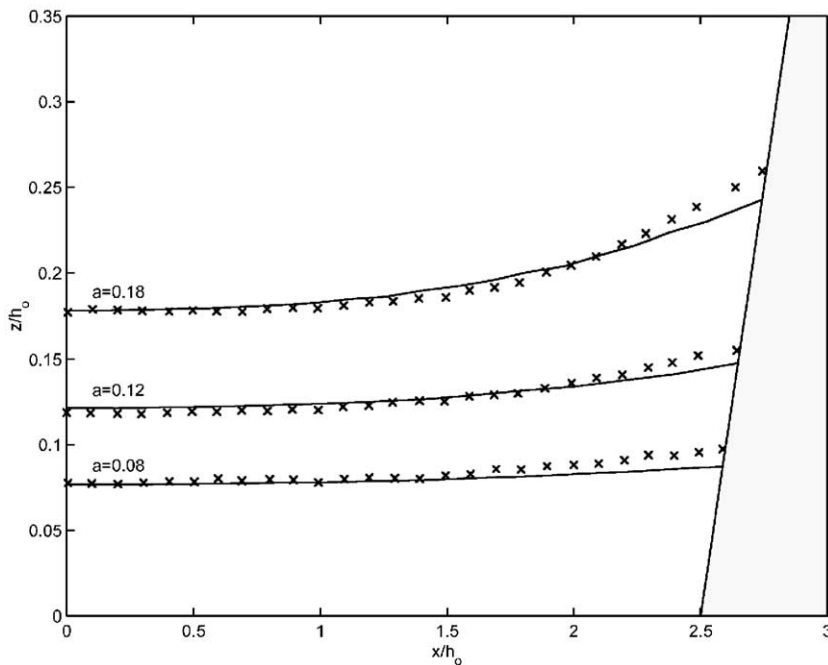


Fig. 12. The transverse profile of a solitary wave in a trapezoidal channel. The continuous line shows the numerical result; the crosses indicate the measured profile digitized from Peregrine’s (1969) paper.

tion of the wave front. When the channel is wide enough, with respect to the wavelength, and the side-wall slope is gradual enough, the wave energy that is reflected off the side walls does not resituate in the original wave. This occurs in the Peregrine (1969) experiments discussed above, but forms a distinct wave behind the original wave front. Wave energy is continually transferred from the original wave front into the new wave behind, until the original wave front virtually disappears. The new front has a smaller height, and a slightly longer wavelength than the original.

One example of the phenomenon is discussed in this section. A half channel is created (one vertical wall at $y=0$, one sloping side wall), with a constant water depth width of $9h_0$ and a length of $250h_0$, where h_0 is the constant water depth along the center of the

channel. The side wall is sloped at 1:5. A solitary wave, with wave height $0.1h_0$ is placed in the channel as an initial condition. The wave does not break, and bottom friction is not included. For this simulation, $\Delta x/h_0=0.14$ and $\Delta t\sqrt{g/h_0}=0.05$ are used.

Fig. 13 shows four snapshots, in plan view, of the wave propagating through the part of channel. The dashed line plotted across the channel is the $x - ct=0$ line, where c is the linear long wave speed, $\sqrt{gh_0}$. Seafloor elevation contours are also shown on each plot. Fig. 13a shows the wave soon after the simulation has begun, and the front is beginning to arc, due to slower movement in the shallower water. By the time shown in Fig. 13b, wave energy has reflected off the slope, and has formed a second, trailing, wave crest behind the original wave. As this slope-reflected wave crest interacts with the vertical wall (or center-

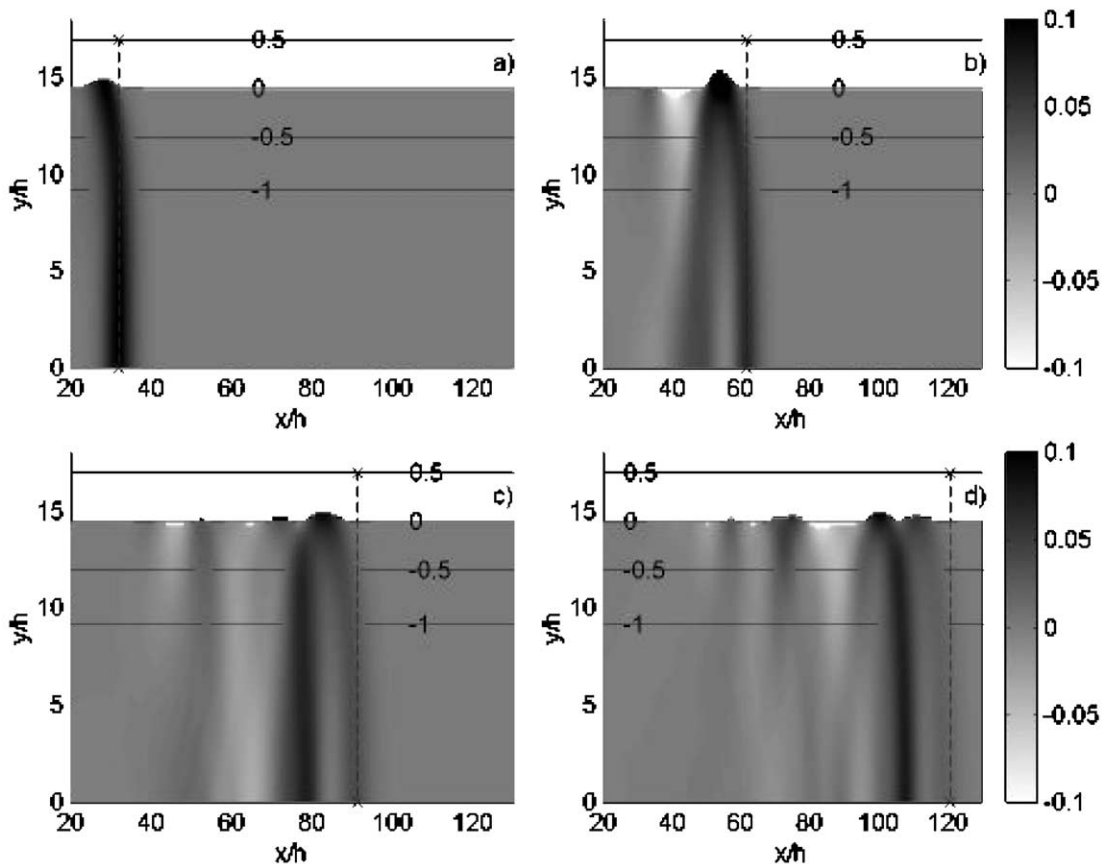


Fig. 13. Evolution of a solitary wave in a trapezoidal channel (half channel shown), at $t(g/h)^{1/2}=(a) 7.5, (b) 35, (c) 65, (d) 93$. Seafloor elevation contours are shown at increments of $0.5h_0$, by the solid lines. The line of $x - ct=0$ is shown by the dashed line.

line of channel), a Mach stem forms at the vertical wall, and virtually no wave energy is reflected off the vertical wall. Also at this time, an oscillatory train, trailing the leading wave, forms along the slope. At time = 65, shown in Fig. 13c, most of the wave energy has transferred from the original wave front, to the secondary crest. In the last plot, Fig. 13d, the process has started to repeat itself, evidenced by the lobe growing behind the second front, near a depth of 0.9.

This process can be examined from a different perspective with Fig. 14. This figure shows numerous time series, taken along the centerline of the channel ($y=0$). Also shown are three characteristic lines. Following the first characteristic, we can see that the lead wave as nearly disappeared by $x=140h_0$, whereas the secondary wave is clearly defined by this point. The process repeats; at $x=230h_0$, the secondary wave is vanishing, and a third wave front is beginning to take shape. The phenomenon shown in Figs. 13 and 14 is an interesting one, although not wholly unexpected, and is a demonstration of the interaction between nonlinearity and refraction.

6. Conclusion

A moving boundary algorithm is developed for use with depth integrated equations. Used here in conjunction with a fixed grid finite difference model, the moving boundary algorithm could also be employed by a finite element scheme. Founded around the restrictions of the high-order numerical wave propagation model, the moving boundary scheme employs linear extrapolation of free surface and velocity through the wet–dry boundary, into the dry region. The linear extrapolation is simple to implement and can be straightforwardly incorporated into a numerical model. The technique is numerically stable, does not require any sort of additional dissipative mechanisms or filtering, and conserves mass.

The moving boundary is tested for accuracy using one- and two-dimensional analytical solutions and experimental data sets. Nonbreaking and breaking solitary wave runup is accurately predicted, yielding a validation of both the eddy viscosity breaking parameterization and the runup model. For strongly

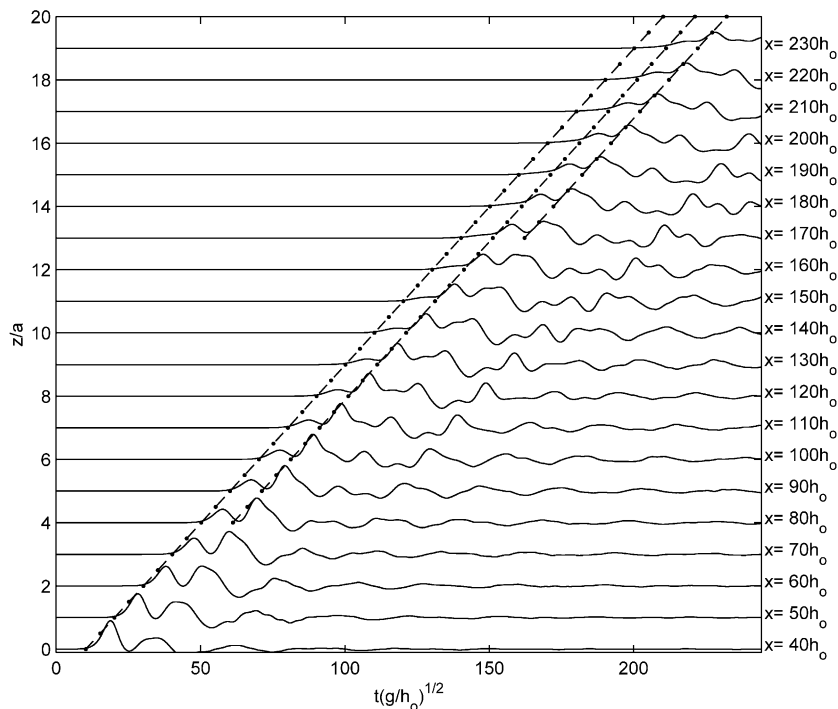


Fig. 14. Time series along the centerline of the channel ($y=0$); location of each time series is note along the right border of the figure. Characteristics are shown by the dashed–dotted lines.

breaking waves, the proper numerical estimation of bottom friction is shown to be important. Two-dimensional wave runup in a parabolic basin and around a conical island is investigated, and comparisons with published data show excellent agreement. Also, solitary wave evolution in a trapezoidal channel is simulated, and an interesting phenomenon is examined.

Acknowledgements

This research has been supported in part by National Science Foundation grants to Cornell University (CMS-9908392, CTS-9808542).

References

- Balzano, A., 1998. Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. *Coast. Eng.* 34, 83–107.
- Briggs, M.J., Synolakis, C.E., Harkins, G.S., 1994. Tsunami runup on a conical island. *Proc. Waves—Physical and Numerical Modeling*. University of British Columbia, Vancouver, Canada, pp. 446–455.
- Carrier, G.F., Greenspan, H.P., 1958. Water waves of finite amplitude on a sloping beach. *J. Fluid Mech.* 4, 97–109.
- Chen, Q., Kirby, J.T., Dalrymple, R.A., Kennedy, A.B., Chawla, A., 2000. Boussinesq modeling of wave transformation, breaking, and runup: Part I. 2D. *J. Waterw. Port Coast. Ocean Eng.* 126 (1), 57–62.
- Cho, Y.-S., 1995. Numerical simulations of tsunami propagation and run-up. PhD Thesis, Cornell University, 264 pp.
- Gopalakrishnan, T.C., 1989. A moving boundary circulation model for regions with large tidal flats. *Int. J. Numer. Methods Eng.* 28, 245–260.
- Hsiao, S.-C., Liu, P.L.-F., 2002. Permeable effects on nonlinear water waves. *Proc. R. Soc. Lond.* in press.
- Kennedy, A.B., Chen, Q., Kirby, J.T., Dalrymple, R.A., 2000. Boussinesq modeling of wave transformation, breaking, and runup. Part I: 1D. *J. Waterw. Port Coast. Ocean Eng.* 126 (1), 39–47.
- Kirby, J.T., Wei, G., Chen, Q., Kennedy, A.B., Dalrymple, R.A., 1998. “FUNWAVE 1.0. Fully nonlinear Boussinesq wave model. Documentation and user’s manual.” Report CACR-98-06, Center for Applied Coastal Research, Department of Civil and Environment Engineering, University of Delaware.
- Kowalik, Z., Bang, I., 1987. Numerical computation of tsunami runup by the upstream derivative method. *Sci. Tsunami Hazards* 5 (2), 77–84.
- Leendertse, J.J., 1987. “Aspects of SIMSYS2D, a system for two-dimensional flow computation.” R-3752-USGS. Rand, Santa Monica, CA, 80.
- Lin, P., Chang, K.-A., Liu, P.L.-F., 1999. Runup and rundown of solitary waves on sloping beaches. *J. Waterw. Port Coast. Ocean Eng.* 125 (5), 247–255.
- Liu, P.L.-F., 1994. Model equations for wave propagation from deep to shallow water. In: Liu, P.L.-F. (Ed.), *Advances in Coastal Engineering*, vol. 1, pp. 125–157.
- Liu, P.L.-F., Cho, Y.-S., Briggs, M.J., Kanoglu, U., Synolakis, C.E., 1995. Runup of solitary waves on a circular island. *J. Fluid Mech.* 302, 259–285.
- Lynett, P., Liu, P.L.-F., 2002a. A numerical study of submarine landslide generated waves and runup. *Proc. R. Soc. Lond.* in press.
- Lynett, P., Liu, P.L.-F., 2002b. A two-dimensional, depth-integrated model for internal wave propagation. *Wave Motion* in press.
- Lynett, P., Liu, P.L.-F., Losada, I., Vidal, C., 2000. Solitary wave interaction with porous breakwaters. *J. Waterw. Port Coast. Ocean Eng.* 126 (6), 314–322.
- Madsen, P.A., Sorensen, O.R., 1992. A new form of the Boussinesq equations with improved linear dispersion characteristics: Part II. A slowly varying bathymetry. *Coast. Eng.* 18, 183–204.
- Madsen, P.A., Sorensen, O.R., Schaffer, H.A., 1997. Surf zone dynamics simulated by a Boussinesq-type model: Part I. Model description and cross-shore motion of regular waves. *Coast. Eng.* 32, 255–287.
- Nwogu, O., 1993. Alternative form of Boussinesq equations for nearshore wave propagation. *J. Waterw. Port Coast. Ocean Eng.* 119 (6), 618–638.
- Peregrine, D.H., 1967. Long waves on a beach. *J. Fluid Mech.* 27, 815–827.
- Peregrine, D.H., 1969. Solitary waves in trapezoidal channels. *J. Fluid Mech.* 35, 1–6.
- Petera, J., Nassechi, V., 1996. A new two-dimensional finite element model for the shallow water equations using a Lagrangian framework constructed along fluid particle trajectories. *Int. J. Numer. Methods Eng.* 39, 4159–4182.
- Sielecki, A., Wurtele, M.G., 1970. The numerical integration of the nonlinear shallow-water equations with sloping boundaries. *J. Comput. Phys.* 6, 219–236.
- Synolakis, C.E., 1986. The runup of long waves. PhD thesis, California Institute of Technology, Pasadena, CA.
- Synolakis, C.E., 1987. The runup of solitary waves. *J. Fluid Mech.* 185, 523–545.
- Tao, J., 1983. Computation of wave runup and wave breaking. Internal Report, Danish Hydraulics Institute, Denmark.
- Tao, J., 1984. Numerical modeling of wave runup and breaking on the beach. *Acta Oceanol. Sin.* 6 (5), 692–700 (in Chinese).
- Thacker, W.C., 1981. Some exact solutions to the nonlinear shallow water wave equations. *J. Fluid Mech.* 107, 499–508.
- Titov, V.V., Synolakis, C.E., 1995. Modeling of breaking and non-breaking long wave evolution and runup using VTCS-2. *J. Waterw. Port Coast. Ocean Eng.* 121, 308–316.
- Titov, V.V., Synolakis, C.E., 1998. Numerical modeling of tidal wave runup. *J. Waterw. Port Coast. Ocean Eng.* 124 (4), 157–171.
- Wei, G., Kirby, J.T., 1995. A time-dependent numerical code for extended Boussinesq equations. *J. Waterw. Port Coast. Ocean Eng.* 120, 251–261.
- Wei, G., Kirby, J.T., Grilli, S.T., Subramanya, R., 1995. A fully nonlinear Boussinesq model for surface waves: Part I. Highly nonlinear unsteady waves. *J. Fluid Mech.* 294, 71–92.
- Zelt, J.A., 1991. The runup of nonbreaking and breaking solitary waves. *Coast. Eng.* 15, 205–246.