INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:237–261
Published online 19 January 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1172

A particle finite element method applied to long wave run-up

J. Birknes[1,2,‡] and G. Pedersen[1,*,†]

[1] *Mechanics Division, Department of Mathematics, University of Oslo, Oslo, Norway*
[2] *Det Norske Veritas, Høvik, Norway*

SUMMARY

This paper presents a Lagrangian–Eulerian finite element formulation for solving fluid dynamics problems with moving boundaries and employs the method to long wave run-up. The method is based on a set of Lagrangian particles which serve as moving nodes for the finite element mesh. Nodes at the moving shoreline are identified by the alpha shape concept which utilizes the distance from neighbouring nodes in different directions. An efficient triangulation technique is then used for the mesh generation at each time step.

In order to validate the numerical method the code has been compared with analytical solutions and a preexisting finite difference model.

The main focus of our investigation is to assess the numerical method through simulations of three-dimensional dam break and long wave run-up on curved beaches. Particularly the method is put to test for cases where different shoreline segments connect and produce a computational domain surrounding dry regions. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: moving boundaries; particle methods; finite element method; alpha shapes; triangulation; run-up; shallow water equations; dam break

1. INTRODUCTION

One major challenge when solving fluid mechanics problems numerically is the presence of boundaries with motions which are unknown *a priori*. Important examples are surface waves and fluid–structure interactions, where the fluid motion may be described by the Euler or the Navier–Stokes equations. Another example is inundation of land by tsunamis and storm surges, described by depth-integrated long wave equations. While the surface conditions of the fluid is incorporated in the coefficients in the depth integrated PDEs, the shoreline motion

*Correspondence to: G. Pedersen, Department of Mathematics, University of Oslo, Box 1053, Blindern, NO-0316 Oslo, Norway.
†E-mail: geirkp@math.uio.no
‡E-mail: jornb@math.uio.no

Contract/grant sponsor: Det Norske Veritas

Received 17 December 2004
Revised 12 September 2005
Accepted 20 November 2005

still has to be found. The moving shoreline is in some respects similar to a free water surface for the Navier–Stokes equations, since the boundary excursions correspond to motion of material particles with the gravity as restoring force. In both cases, numerical models may be based on dynamic grids adapting to the boundaries or on fixed grids where the boundaries are implemented by particular extrapolation techniques. However, there are differences as well. In run-up, there is a singularity in the governing equations at the shoreline, though not in the physically relevant solutions [1, 2]. In addition, the run-up dynamics are influenced by the beach bathymetry, while a free surface is autonomous. In spite of the similarities of the problems there have been little interaction between the fields of long wave run-up and modelling of general free surface flow. Herein we apply a recent method for the Navier–Stokes equations with free surfaces, namely the particle finite element (PFEM) technique, to long wave run-up. We have a double motivation for this work. Firstly, we wish to exploit the greater transparency of the shallow water equations which manifest themselves through the availability of a series of analytical solutions, in order to gain general insight into the performance of PFEM. Secondly, the long wave run-up is relevant for tsunami and other topics of coastal engineering. Hence, the problems and methods studied here are also important in their own right.

The first general method for inclusion of free surfaces in models of the primitive (Navier–Stokes or Euler) equations was the marker and cell method (MAC) [3]. It was developed in the sixties and employs finite differences, a staggered grid and a set of markers. The first problems addressed with this method were the broken dam problem and flow through an open sluice gate.

Today the Navier–Stokes models with the volume of fluid (VOF) technique for free surfaces are widely used. In the VOF technique, which was first presented in 1981 [4], the fractional volume of fluid in each cell is defined by a function F. This is updated according to particular schemes based on the volume fluxes between the cells. Surface cells are then recognized by $F < 1$. One problem with this method is that the numerical representation of the free surface becomes discontinuous. The method has, among other problems, been applied to breaking waves in the surf zone [5, 6]. In the late eighties, the level set method was presented [7]. Instead of following the interface itself, the method projects the interface onto a surface, known as the level set function.

A different approach for solving complex problems is the smooth particle hydrodynamics (SPH) method [8, 9]. Originally, the method was used to solve problems in astrophysics. The method has later been applied to fluid dynamics problems like run-up, breaking waves and multi-phase flows. This method is very well suited for problems with open boundaries, like free surfaces, while it is more cumbersome to impose boundary conditions on rigid or flexible bodies.

A new and promising method for solving the Navier–Stokes equations with complex free surface dynamics, like wave breaking, sloshing, and fluid–structure interactions, is the particle finite element method (PFEM), which is also referred to as the meshless finite element method (MFEM) [10–12]. This method is based on a mixture of the Lagrangian and the Eulerian formulation, and boundary nodes are recognized by using the alpha shape concept [13]. Major advantages of the PFEM are that it retains continuous and well defined moving boundaries and it is, in principle, easy to impose boundary conditions.

Long wave run-up on sloping beaches has been an active branch of hydrodynamic research for more than half a century. Analytical solutions are available for particular cases. Such solutions for plane non-breaking waves on an inclined plane are found in Reference [14] and

have later been extended by Synolakis [15] and others. Run-up of plane and uniform bores are treated by Keller *et al.* [16] and Shen and Meyer [17], while very simple expressions for eigen oscillations in parabolic basins are summarized by Thacker [18]. The analytic solutions are crucial for our present understanding of run-up on beaches and as test cases for numerical models.

The most vital line of research on long wave run-up is concerned with Eulerian methods which involve tracking of the shoreline and extrapolation of velocity fields onto dry land, in different varieties and combinations. Shallow water models of this type have been reported by, among others, References [19–28], while References [29–31] solve different varieties of high accuracy Boussinesq equations. All of the quoted Eulerian techniques work well for given applications. Moreover, some of the methods are also sufficiently general and robust for use in operational tsunami models. Still, in a fixed grid the shoreline representation is generally coarse, and more or less ingenious extrapolation schemes must be invoked. As a consequence the accuracy at the shoreline is reduced and the noise produced there must be checked by filters or dissipation inherent in the numerical models.

Another type of run-up models is based on automatic tracing of the instantaneous shoreline by application of Lagrangian coordinates, as in References [32–34]. The last reference is particularly relevant in the present context, since it employs a traditional finite element approach to a standard set of Boussinesq equations. A related approach is the use of the arbitrary Lagrangian Eulerian (ALE) methods where the computational domain is mapped onto a fixed coordinate region, generally consisting of one or more rectangles, but where the nodes not necessarily correspond to particles. Examples of solutions of the shallow water equations by ALE techniques are found in References [28, 35–38]. The last of these is a brief paper where a meshless technique, utilizing smooth shape functions and collocation is applied to plane waves. Very accurate descriptions of the moving shoreline is reported in some of the references on Lagrangian or ALE methods for idealized run-up problems. Then the main reason for the preference of Eulerian models is that Lagrangian (and ALE) grids often become severely distorted, with respect both to aspect ratios and densities, and thus need repeated re-meshing. In addition bathymetric data, for instance, must be re-interpolated at each time step. Hence, even though Lagrangian models may give very accurate solutions for the simpler problems they seem somewhat unpromising as general solution strategies. The remedy may then be to abandon the grid altogether, as in the SPH method, or employ a grid with a dynamic connectivity as we do here.

Following References [10–12], we convey the basic ideas of PFEM, as well as most of the particulars, to the nonlinear shallow water equations. We are employing relatively simple, but illustrative, test problems as eigen oscillations in basins, dam break and run-up, and compare the PFEM solutions with analytical and numerical solutions obtained by a finite difference model [39]. Dam break of a cone of water and long wave run-up on two different three-dimensional beaches have been carried out to demonstrate the capability of the method. The first beach is a headland while the latter is a bay with a hill on the beach. The last case involves both creation and destruction of shoreline segments.

The numerical model is described in Section 2, together with an analysis of the numerical scheme. The problems of eigen oscillations in elliptic basins, a broken dam and long wave run-up have been chosen to verify the numerical model, and are presented in Section 3. Section 4 presents the simulations of three-dimensional dam break and long wave run-up on three-dimensional beaches. Finally, the conclusions are summarized in Section 5.

2. THE LAGRANGIAN–EULERIAN MODEL APPLIED TO THE SHALLOW WATER EQUATIONS

We mark dimensional quantities by a superscript asterisk and introduce two length scales L and h_0 and one velocity scale c_0. L is a typical wavelength, h_0 is the maximum depth, and $c_0 = \sqrt{gh_0}$ is the shallow water wave speed assuming linear theory. This leads to the following definition of non-dimensional variables:

$$\begin{aligned} x^* &= Lx, \quad y^* = Ly, \quad z^* = h_0 z, \quad t^* = Lc_0^{-1}t \\ \eta^* &= h_0\eta, \quad \mathbf{v}^* = c_0\mathbf{v}, \quad h^* = h_0 h \end{aligned} \tag{1}$$

The z-axis points vertically upward and the xy-plane defines the still water level. Assuming the shallow water theory which is valid under the assumption that $\varepsilon = (h_0/L)^2$ is small, we arrive at the NLSW (see for example Reference [40]) equations

$$\frac{\mathrm{D}\mathbf{v}}{\mathrm{d}t} = -\nabla\eta + D_{\mathrm{a}}(u;\kappa_u)\mathbf{i} + D_{\mathrm{a}}(v;\kappa_v)\mathbf{j} \tag{2}$$

$$\frac{\mathrm{D}\eta}{\mathrm{d}t} = -\nabla\cdot[(h+\eta)\mathbf{v}] + \mathbf{v}\cdot\nabla\eta + D_{\mathrm{a}}(\eta;\kappa_\eta) \tag{3}$$

where $\mathbf{v} = (u, v)$ is the horizontal velocity and η is the wave elevation. The operator $\mathrm{D}/\mathrm{d}t \equiv \partial/\partial t + \mathbf{v}\cdot\nabla$ is the shallow water version of material differentiation and the function D_{a} is artificial diffusion introduced to avoid unphysical oscillations and is given by

$$D_{\mathrm{a}}(u,\kappa) = \nabla\cdot(\kappa(\mathbf{x})\nabla u) \tag{4}$$

For our numerical implementation of (2)–(3) we have assumed that κ is a constant and preferably as small as possible. Other possible choices would be the streamline diffusion method or application of a κ that depends on the surface gradient [30].

The boundary condition at a moving shoreline is $\eta + h = 0$ corresponding to a vanishing fluid depth. Since our set of test cases includes numerical wave tanks with a paddle, vertical side walls and a beach, we will also need boundary conditions for moving and fixed vertical boundaries. The general wall boundary condition is $\mathbf{v}\cdot\mathbf{n} = v_{\mathrm{w}}$ where $\mathbf{n}$ is the normal vector of the wall and v_{w} is the velocity of the wall.

2.1. Numerical method

Our strategy for solving the shallow water equations numerically is to mix the Lagrangian and Eulerian formulations. To this end, Equations (2)–(3) have been written with Lagrangian accelerations as left-hand sides. We emphasize that the Lagrangian points serve as nodes in the element mesh which has Eulerian trial functions. This leads to a multi-step algorithm where one of the crucial steps is the computation of new particle positions, which also serve as nodes in a FEM discretization according to

$$\frac{\mathrm{D}\mathbf{x}}{\mathrm{d}t} = \mathbf{v} \tag{5}$$

Following Reference [10] on the Navier–Stokes models, we employ first-order temporal discretization. The time advance from t^n to $t^{n+1} = t^n + \Delta t$ may then be outlined as follows:

1. Velocities at t^{n+1} are computed from Equation (2) by inserting old values from time t^n on the right-hand side.

2. New node positions are found from (5) by a backward temporal difference, inserting velocities at t^{n+1} at the right-hand side.
3. A new connectivity of the grid is computed at t^{n+1}, boundary nodes are recognized by the alpha-shape concept and nodes that are clustered too densely are merged (refer to Section 2.2).
4. The continuity Equation (3) is then solved on the new mesh. On the right-hand side velocities and surface elevations are inserted from t^{n+1} and t^n, respectively. In the linear case this corresponds to a backward temporal differentiation.

The velocity component u at time t^n can be approximated by $u^n = \sum_{j=1}^{M} u_j^n N_j(\mathbf{x})$, where u_j^n are the unknown coefficients and N_j are piecewise linear trial functions which equal unity at node j and which are zero at the remaining nodes. Similar expressions apply to v and η. Employing the Galerkin's method to Equation (2) we arrive at the following equations for u_j^{n+1} and v_j^{n+1}:

$$\sum_{j=1}^{M} A_{i,j} u_j^{n+1} = \int_\Omega \left(u^n N_i + \Delta t \eta^n \frac{\partial N_i}{\partial x} \right) \mathrm{d}\Omega - \Delta t \int_{\partial\Omega} \eta^n N_i n_1 \, \mathrm{d}\Gamma \tag{6}$$

$$\sum_{j=1}^{M} A_{i,j} v_j^{n+1} = \int_\Omega \left(v^n N_i + \Delta t \eta^n \frac{\partial N_i}{\partial y} \right) \mathrm{d}\Omega - \Delta t \int_{\partial\Omega} \eta^n N_i n_2 \, \mathrm{d}\Gamma \tag{7}$$

where $A_{i,j} = \int_\Omega N_i N_j \, \mathrm{d}\Omega$ and $\mathbf{n} = (n_1, n_2)$ is the normal vector at $\partial\Omega$ which is the boundary of the fluid domain Ω. Equations (6) and (7) are employed for i values corresponding to nodes in the interior or at the moving shoreline. For nodes on a vertical wall or a moving wave paddle only the tangential component is invoked (see the list of essential boundary conditions below). Node positions are being updated according to

$$x_j^{n+1} = x_j^n + \Delta t u_j^{n+1} \quad \text{and} \quad y_j^{n+1} = y_j^n + \Delta t v_j^{n+1} \tag{8}$$

in the whole domain, including boundaries. Similarly, the weak formulation of Equation (3) is

$$\begin{aligned} \sum_{j=1}^{M} A_{i,j} \eta_j^{n+1} = & \int_\Omega [(\eta^n + \Delta t \mathbf{v}^{n+1} \cdot \nabla \eta^n) N_i + \Delta t (h + \eta^n) \mathbf{v}^{n+1} \cdot \nabla N_i] \, \mathrm{d}\Omega \\ & - \Delta t \int_{\partial\Omega} (h + \eta^n) \mathbf{v}^{n+1} \cdot \mathbf{n} N_i \, \mathrm{d}\Gamma \end{aligned} \tag{9}$$

This equation is employed for all nodes except those at the moving shoreline and we observe that the line integral only gives contributions at a moving wave paddle. For readability, the diffusion terms are omitted in Equations (6), (7) and (9).

The following essential boundary conditions are imposed:

1. Zero normal velocity at fixed walls, $\mathbf{v} \cdot \mathbf{n} = 0$.
2. At moving walls (piston type paddles), the normal component of the fluid velocity equals that of the wall; $\mathbf{v} \cdot \mathbf{n} = v_\mathrm{w}$, where v_w is the velocity of the wall.

3. Vanishing fluid depth at the beach, $\eta + h = 0$, is used to find surface elevations at the shore instead of (9). Preliminary tests with a natural boundary condition at the shoreline did not give promising results.

The present algorithm is implemented in the Diffpack software environment [41]. The decoupled equation sets for u, v and η are solved by the Conjugate Gradient method with RILU preconditioning [42].

2.2. *Computational domain*

A triangular finite element mesh is produced from the positions of the nodes at each time step. The generation of the mesh is based on Guibas and Stolfi's algorithm [43]. Given a point set of n nodes, this algorithm produces a mesh at a time cost of order $n \log n$, and the algorithm has been implemented in the mesh generator Triangle [44, 45]. This generator produces a convex mesh, meaning that all holes and concave parts of the domain are patched with elements. Next, there is a need to identify boundary nodes and boundary surfaces and remove the false elements introduced by the mesh algorithm. To determine the boundary nodes the alpha shape concept [13] has been applied. Node i will be classified as a boundary node if there exists a disc, which contains no other nodes than node i, with a given radius r_i and node i at the circumference (refer to Figure 1(a)). In general, the radius of the disc for the node i is given by $r_i = \alpha d_i$, where d_i is the minimum distance between the node considered and the neighbouring nodes and α is a constant. This introduces an error for the boundary surface which is of order d_i, as noted in Reference [10]. Determining the α-coefficient is not trivial. A value which is too small, may introduce incorrect boundaries and unphysical holes in the fluid domain. On the other hand, real holes in the fluid domain may not be recognized if the α-coefficient is too large. For our simulations the value of α is of order 2.

When the boundary nodes are detected, all elements outside the fluid domain must be removed before solving the problem with Galerkin's method. The elements outside the fluid domain are identified as elements which only have boundary nodes. This strategy may remove elements at corners which should be part of the fluid domain, thus extra treatment of elements defining corners is needed. For the present simulations, extra care must be taken

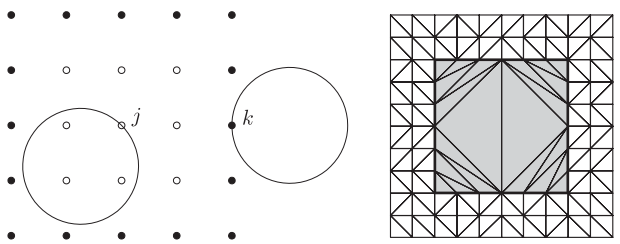

Figure 1. In (a) the filled bullets indicate the boundary nodes and the empty circles are interior nodes. Node j is classified as an interior node, since all possible discs contain neighbouring nodes. For node k, it is possible to place the disc in such a way that it does not contain any other nodes than itself. Therefore node k is a boundary node. (b) shows the mesh of a square domain with a hole indicated by the bold line and the grey background.

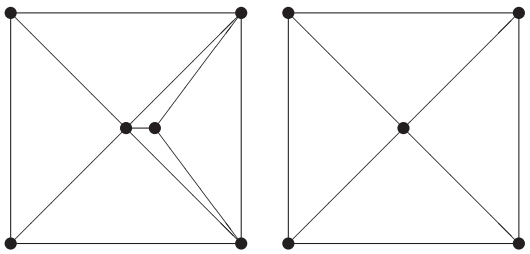

Figure 2. An example of merging nodes in the mesh. Left: The two nodes in the middle are identified as 'too close' and will be merged. Right: The result after merging.

when considering the shoreline segments at the walls of the numerical wave tank. As an example, a mesh of a square domain with a hole is shown in Figure 1(b). As seen, the original mesh has elements covering the hole marked with a grey background. All the elements inside the hole have their nodes on the boundary of the hole and must be removed in order to obtain a mesh with a hole.

Another aspect with the mesh generation is that the nodes are moved at each time step according to (8). As a consequence, some of the nodes may come very close to each other. In most cases the triangular mesh generated from this node distribution contains elements with bad aspect ratios. One way of overcoming this problem is to proceed as Reference [10], and introduce non-Sibsonian interpolants and polyhedral elements. It is worth noting that the non-Sibsonian interpolants are reduced to linear interpolants for three node elements and bi-linear interpolants for four node elements. However, another strategy is feasible. Instead of using polyhedral elements it is possible to merge nodes which are getting too close to each other, and obtain a mesh where all the triangular elements have acceptable aspect ratios. According to the Lagrangian–Eulerian formulation of the problem it is valid to merge nodes in the computational domain when considering the equations as Eulerian. The new field values in the merged nodes are taken as the average field value of the nodes which have been merged. Following these arguments it is also possible to insert nodes in the fluid domain where the node distribution is coarse which would be advantageous in many cases. This has not been implemented here.

Our criterion for detecting and merging nodes is given by

$$\frac{d(i,j)}{d_0} < \delta \tag{10}$$

where $d(i,j)$ is the distance between node i and j, and d_0 is the minimum distance between two nodes in the initial mesh. The constant δ is a number between zero and one. An example of merging two nodes is shown in Figure 2 where the mesh to the left has two bad shaped elements. After the merging the mesh has only elements with good aspect ratios as seen in the right mesh.

2.3. *Linear analysis of the numerical scheme*

In order to carry out a stability analysis of the scheme (6)–(9) we simplify the problem. We are assuming constant depth $h \equiv 1$, one-dimensional motion, and we are linearizing the

shallow water equations (2) and (3). Solving these equations by Galerkin's method with linear elements, constant spacing between the nodes and exact evaluation of all integrals, we obtain the following system for the nodal values u_j^n and η_j^n:

$$\frac{h}{6}(u_{i-1}^{n+1} + 4u_i^{n+1} + u_{i+1}^{n+1}) = \frac{h}{6}(u_{i-1}^n + 4u_i^n + u_{i+1}^n) - \frac{\Delta t}{2}(\eta_{i+1}^n - \eta_{i-1}^n) \tag{11}$$

$$\frac{h}{6}(\eta_{i-1}^{n+1} + 4\eta_i^{n+1} + \eta_{i+1}^{n+1}) = \frac{h}{6}(\eta_{i-1}^n + 4\eta_i^n + \eta_{i+1}^n) - \frac{\Delta t}{2}(u_{i+1}^{n+1} - u_{i-1}^{n+1}) \tag{12}$$

Insertion of periodic solutions $u_j^n = \hat{u}\exp(\mathrm{i}(kx_j - \omega t^n))$ and $\eta_j^n = \hat{\eta}\exp(\mathrm{i}(kx_j - \omega t^n))$ into the scheme above result in two equations for the amplitudes $\hat{u}$ and $\hat{\eta}$. The numerical dispersion relation is obtained from these equations and reads

$$(2 + \cos k\Delta x)\sin\frac{\omega}{2}\Delta t = \frac{3}{2}\frac{\Delta t}{\Delta x}\sin k\Delta x \tag{13}$$

A Taylor expansion of the numerical dispersion relation is straightforward and reads

$$\omega = k + \tfrac{1}{24}k^3\Delta t^2 + \mathcal{O}(\Delta t^4, \Delta t^2\Delta x^2, \Delta x^4)$$

which somewhat surprisingly shows that the accuracy of the dispersion relation is of fourth-order in space and second-order in time. In the limit $\Delta t, \Delta x \to 0$ we have $\omega = k$ which is identical to the analytical dispersion relation $\omega = k$. The unexpectedly high accuracy for this particular case, with plane waves, constant depth, linear equations, and uniform resolution, may be readily explained. First, the temporal alignment in (11) and (12) corresponds to what is often denoted as a forward/backward scheme. The point is that we may re-interpret the η nodes, for instance, as values taken at semi-integral temporal locations $(n + \frac{1}{2})\Delta t$. Apart from boundary conditions, it appears that we in fact have an alternating midpoint discretization in time which is of second-order accuracy. Still, this will not carry over to the general case with nonlinear terms and where both the momentum and continuity equations are not discretized on the same instant of the time dependent grid (refer to Section 3.1). The enhanced spatial accuracy is due to a cancellation effect between the errors of consistent mass representation of the time derivatives and the leap-frog formula for the x derivative. We may demonstrate this by inserting an exact solution of the PDEs into (11) and (12), and evaluate the residual R. Starting with a slight reorganization of (12) we find

$$\begin{aligned} R &= \frac{1}{6\Delta t}\left(\eta_{i-1}^{n+1/2} + 4\eta_i^{n+1/2} + \eta_{i+1}^{n+1/2} - \eta_{i-1}^{n-1/2} - 4\eta_i^{n-1/2} - \eta_{i+1}^{n-1/2}\right) + \frac{1}{2h}(u_{i+1}^n - u_{i-1}^n) - \frac{\partial\eta}{\partial t} - \frac{\partial u}{\partial x} \\ &= \frac{h^2}{6}\frac{\partial^3\eta}{\partial t\partial x^2} + \frac{\Delta t^2}{24}\frac{\partial^3\eta}{\partial t^3} + \frac{h^2}{6}\frac{\partial^3 u}{\partial x^3} + \mathcal{O}(\Delta t^4, \Delta t^2 h^2, h^4) \\ &= \mathcal{O}(\Delta t^2, \Delta t^2 h^2, h^4) \end{aligned}$$

where the $\mathcal{O}(h^2)$ terms in the mid-line cancel out because u and η fulfill the continuity equation $\partial\eta/\partial t + \partial u/\partial x = 0$. The same analysis applies to the momentum equation (11). However, in general we can expect only second-order accuracy in h.

The stability criterion is also found from (13), and it follows that

$$\left| \frac{3C \sin k\Delta x}{4 + 2\cos k\Delta x} \right| \leqslant 1$$

where $C = \Delta t/\Delta x$ is the Courant number. Hence the stability criterion becomes $C \leqslant 2\sqrt{3}/3 \approx 1.15$.

Unfortunately, the very high accuracy for the dispersion relation found above do not apply to the full nonlinear scheme. The temporal discretization of Equation (8) is of first-order. In addition, taking the nonlinearities and the varying depth into account suggest that the method generally is of first-order accuracy in time and of second-order in space. In particular cases with complex shoreline dynamics, the application of the alpha shape concept may even reduce the spatial accuracy below second-order. Confirmation of the first-order accuracy in the time increment is demonstrated by the example of eigen oscillations in elliptic basins (refer to Section 3.1).

3. VERIFICATION

3.1. Eigen oscillations in an elliptic basin

For eigen oscillations in parabolic basins in particular, simple analytical solutions are available from the literature [18]. In relation to the scaling (1) we assume that both horizontal half axes equal L and that the maximum depth is h_0. The non-dimensional depth becomes $h = 1 - x^2 - y^2$. For the lowest mode with motions only along the x-axis the solution is given by

$$u = -a\omega \sin \omega t, \quad v = 0 \tag{14}$$

$$\eta = 2a \cos \omega t \left(x - \tfrac{1}{2} a \cos \omega t \right), \quad \omega^2 = 2 \tag{15}$$

where a is the non-dimensional amplitude. The instantaneous shoreline remains a circle in the horizontal plane with radius 1 and centre at $(a \cos \omega t, 0)$. As seen from (15), the wave elevation is a planar surface, and the fluid sloshes back and forth in the basin. The initial shoreline and the wave elevation at some selected times are plotted in Figure 3 for $a = 0.5$.

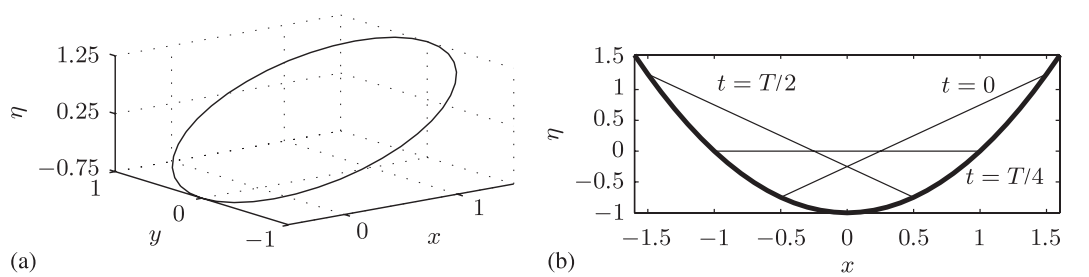

Figure 3. Eigen oscillations in elliptic basins. The bold line in (b) is the bottom: (a) the shoreline for $t = 0$; and (b) analytical solution for the wave elevation at different times (T is the period).

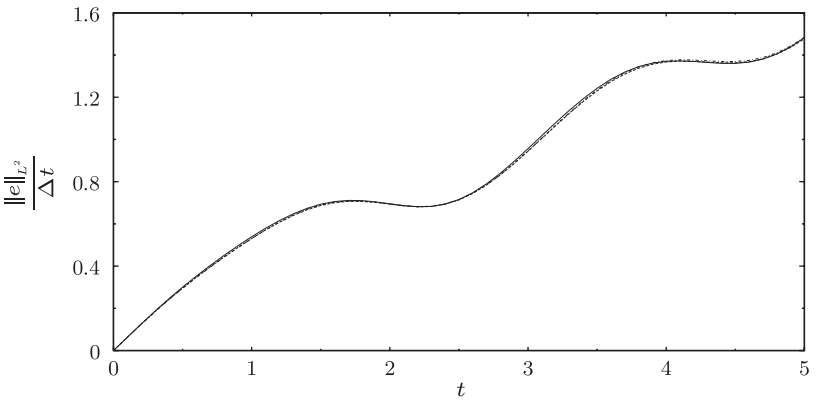

Figure 4. Eigen oscillations in elliptic basins. The error for the wave elevation measured in the L^2-norm, normalized by the time increment, is shown for $\Delta t = 0.02$ (dash-dot line), $\Delta t = 0.01$ (dashed line) and $\Delta t = 0.005$ (solid line). The number of nodes is 547 for all cases.

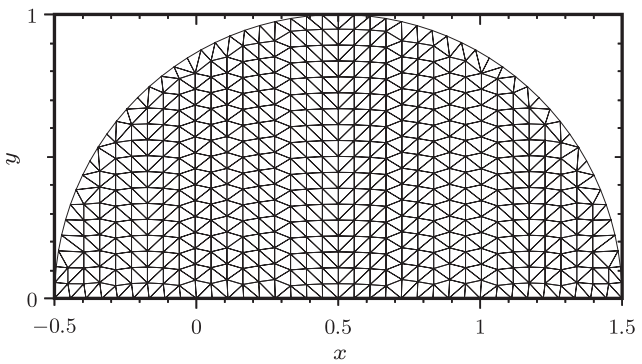

Figure 5. The initial grid (547 nodes) for the problem of eigen oscillations in elliptic basins.

Some methods [33] might capture the simple spatial structure of solutions like (15) exactly. Nevertheless this is no guarantee for excellent performance in relation to other applications. The present model does not reproduce the linear surface profile exactly due to the shoreline boundary. The deviation is mainly dependent on the time increment and its L^2-norm is displayed in Figure 4. We define the L^2-norm according to $\|e\|_{L^2(\Omega)} = (\int_\Omega e^2 \, \mathrm{d}\Omega)^{1/2}$ where e is the difference between the exact and the numerical solution. The first-order temporal accuracy is demonstrated in Figure 4 for $a = 0.5$. Figure 5 shows the initial mesh where only half of the fluid domain is discretized due to the symmetry of the problem.

This problem is almost unaffected by the artificial diffusion introduced in (2)–(3) since the instantaneous analytical solution for the velocity is constant and the wave elevation is linear. For the result in Figure 4 we have used $\kappa = 1.0 \times 10^{-3}$. A reduction of κ to 1.0×10^{-4} changed the results for the L^2-error for the wave elevation by less than 6 per thousand.

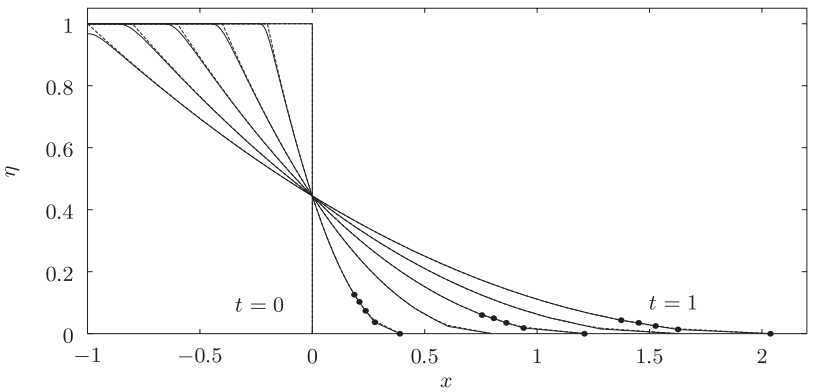

Figure 6. Analytical (dashed line) and numerical wave elevation (solid line) for dam break shown at $t = \{0, 0.2, 0.4, \ldots, 1\}$. The bullets show the node positions near the water front.

3.2. The broken dam problem

The breaking of a dam has a simple closed form solution when assuming hydrostatic pressure. For details we refer to Reference [46]. Applying the scaling introduced in (1), the solution for $t > 0$ and $-t \leqslant x \leqslant 2t$ is given by

$$\eta = \frac{1}{9}\left(\frac{x}{t} - 2\right)^2 \quad \text{and} \quad u = \frac{2}{3}\left(1 + \frac{x}{t}\right)$$

For $x < -t$ and $x > 2t$ the solution is $\eta = 1$ and $\eta = 0$, respectively, and the corresponding velocity u is zero. A wall condition is imposed at $x = -1$ for the numerical simulations to avoid an infinite length of the domain. For this reason the comparison between the numerical and analytical solution is only valid for $t \leqslant 1$.

Figure 6 shows a comparison between the exact and the numerical solution. Small differences are observed at $x = -t$ where the analytical solution has a break, and at the water front $x = 2t$. Reduced artificial diffusion and finer grids would decrease this error. For this case the number of nodes is 400 in the x-direction, corresponding to element lengths of 2.5×10^{-3}, and the time step is 6.2×10^{-4}. Initially the nodes are evenly distributed. The bullets in Figure 6 show the nodes at the water front for the numerical simulation. We see that the distance between the nodes increases while the water front moves in the x-direction, meaning that the elements in the front are stretched.

3.3. Long wave run-up

For run-up on a beach we compare the PFEM with the finite difference method (FDM) by Pedersen and Gjevik [32]. The geometry is shown in Figure 7. For this example the water is initially at rest and the wave is generated by a wave piston located at the left of the wave tank. The velocity of the wave piston is defined by

$$v_P = \begin{cases} \dfrac{V_P}{2}(1 + \cos(2\pi t/T - \pi)) & \text{if } t \leqslant T \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

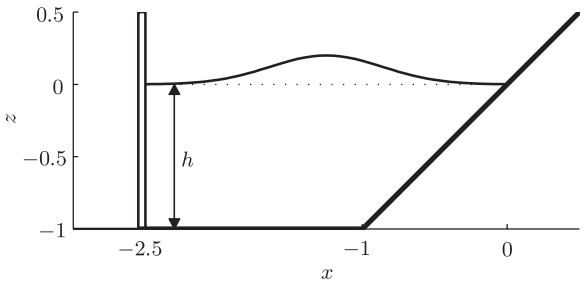

Figure 7. The numerical wave tank which has been used for comparing the present code against a finite difference model.

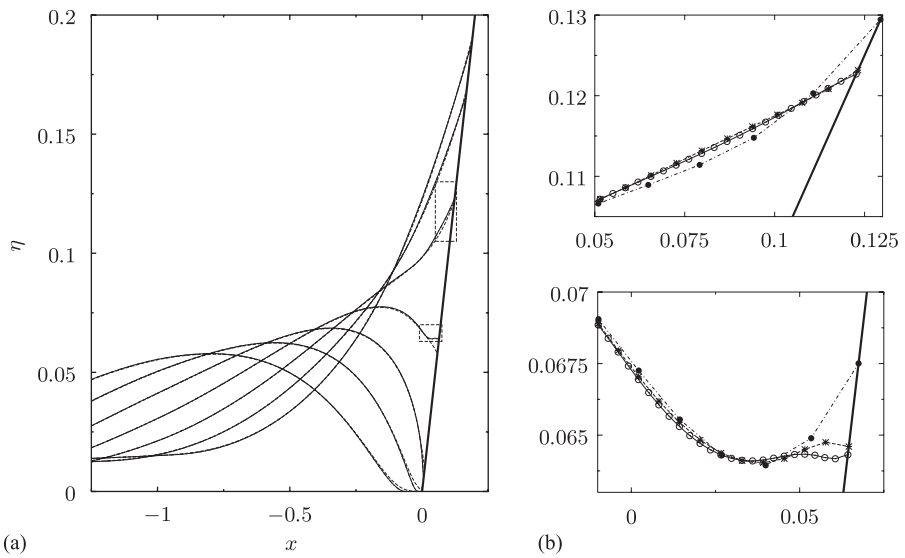

Figure 8. Run-up on a linear beach: (a) a comparison between the finite difference model [32] and the present work. Results are shown for $t = \{3.0, 3.25, \ldots, 4.5\}$ and the smallest and largest run-up are for $t = 3.0$ and $t = 4.5$, respectively. Solid and dash-dot lines: The finite difference model and the present work (800 nodes). On this scale it is impossible to distinguish these curves from each other. Dashed line: simulation with 190 nodes and $\kappa = 1.0 \times 10^{-3}$; and (b) two close-ups of the wave elevation near the beach, as indicated by the small boxes in (a), for $t = 3.75$ and 4. The results are shown for three different grid resolutions: 200 (dash-dot line), 400 (dashed line) and 800 (solid line). The symbols $(\bullet, *, \circ)$ show the node positions.

where V_P is the maximum velocity of the wave piston and T is the period. We have employed the scaling (1) with h_0 as the maximum equilibrium depth and $L = h_0 \cot\theta$, where θ is the beach inclination. Non-dimensionally, we then obtain a 1 in 1 slope and the slope ends at an off-shore depth equal to 1. Moreover, the length of the horizontal bottom is set to 1.5, $V_P = 0.05$ and $T = 3$.

The results are shown in Figure 8. The panel 8(a) shows the comparison between the FDM model and the present model with no artificial diffusion. Instead a 5-point smoothing formula

has been applied at time intervals of 0.05. For both simulations the mesh has 800 nodes corresponding to element lengths of order 3.1×10^{-3} and the time step for the FDM model is 2.08×10^{-3}, while it is 7.81×10^{-4} for the present model. For the reference frame chosen in Figure 8(a), it is impossible to distinguish between the two methods. The effect of the artificial diffusion is illustrated with the dashed curve in Figure 8(a) where $\kappa = 1.0 \times 10^{-3}$. Both the diffusion coefficient and element lengths (of order 0.013), as well as the time step (6.25×10^{-3}), correspond to those of the three-dimensional run-up cases in Section 4. The effect of diffusion is generally very negligible, even though small differences are evident close to the beach and the maximum run-up is reduced from 0.192 to 0.190.

Two close-ups of the long wave run-up near the beach are shown in Figure 8(b) for $t = 3.75$ and 4, displaying the mesh dependence. Three meshes with 200, 400 and 800 nodes have been used. For $t = 3.75$ some differences are observed in the vicinity of the beach, but otherwise the three curves more or less coincide. Also observe that the run-up results from the dense mesh has some very short scale features near the beach. A similar feature was reported by Jensen *et al.* [39], who applied a Boussinesq model for simulating run-up on a linear beach. However, at a later stage, $t = 4$, the differences between the results from the two finest meshes have vanished, while the results from the coarsest mesh still differs from the other two. For these simulations the time step is 3.125×10^{-3} for the coarse mesh, and for the medium and fine mesh the time step is reduced to one half and one quarter, respectively. The symbols in Figure 8(b) show the node positions near the beach for the three grids.

4. SIMULATIONS

4.1. Dam break of a cone

The classical solution for the breaking of a dam is physically two-dimensional. To include three-dimensional effects one may study the dam break of a vertical cone of water with a circular base. Initially the radius and the height of the cone are both set to 1 (non-dimensional unit). Due to the symmetry of the problem, the computational domain is limited to one quarter of a circular disc. We have employed three different grids with 80, 1099 and 2226 nodes, respectively. This corresponds to typical element lengths of 0.11, 0.03 and 0.02. Time steps are chosen as 0.033, 9.0×10^{-3} and 7.0×10^{-3}, respectively, and the initial grid, with 1099 nodes, is shown in Figure 9.

Surface profiles, along the line $y = x$, are displayed in Figure 10. It is almost impossible to distinguish between the results from the two finest grids, while some discrepancies are observed for the coarsest discretization. Due to the symmetry of the problem, the water front should be circular for all times, and this is confirmed by Figure 11 which shows the water front given by $\eta + h = 0$.

4.2. Run-up on three-dimensional beaches

Long wave run-up on three-dimensional beaches will be simulated in a numerical wave tank as illustrated in Figure 7. Two different geometries, with and without an idealized hill on the shore, are considered. Both can be expressed as

$$h(x, y) = q(x - s(y)) + \gamma(x, y), \quad s(y) = B \exp\left[-\left(\frac{y}{r}\right)^2\right] \tag{17}$$

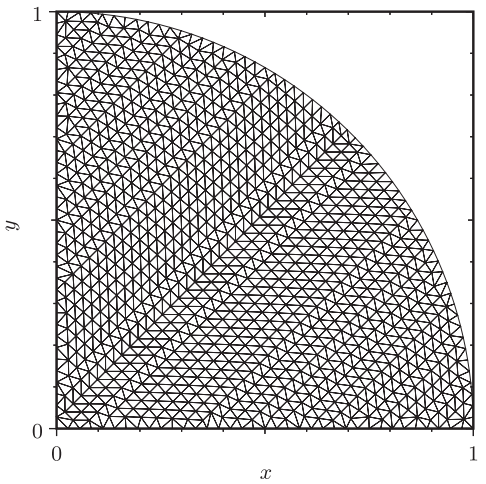

Figure 9. The initial grid (1099 nodes) for the problem of dam break of a cone.

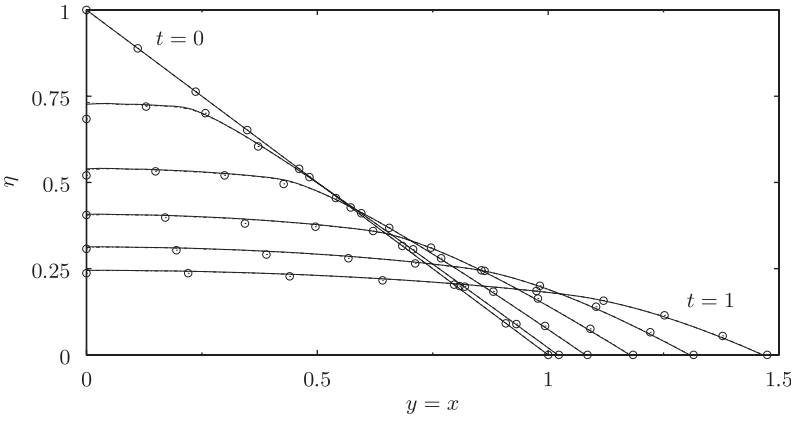

Figure 10. Dam break of a vertical cone. The wave elevation is shown for $y = x$ and $t = \{0, 0.2, \ldots, 1\}$ for three different grid resolutions: 80 nodes ($\odot$), 1099 nodes (dashed line) and 2226 nodes (solid line). Note that it is almost impossible to distinguish between the results from the two finest grids.

where s defines a gentle bay or headland, γ may represent a hill (sand-dune), and

$$q(\xi) = \begin{cases} (\tan\theta)\xi & \text{if } \xi < \cot\theta - l \\ 1 - \dfrac{\tan\theta}{4l}(\xi - \cot\theta - l)^2 & \text{if } -l < \xi - \cot\theta < l \\ 1 & \text{if } \xi > \cot\theta + l \end{cases} \tag{18}$$

In the expressions above, the protrusion of the headland/bay is B, its width is of order $2r$, and the beach inclination is θ. Note that the beach is a headland if B is positive and a bay

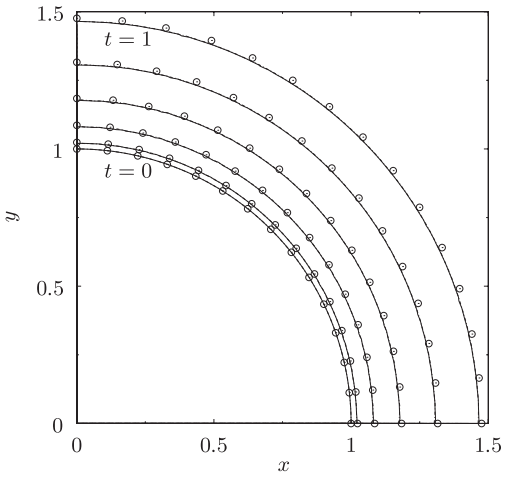

Figure 11. Dam break of a vertical cone. The curves show the water front in the xy-plane for $t = \{0, 0.2, \ldots, 1\}$. The corresponding wave elevation is shown in Figure 10. Grid resolutions: 80 nodes ($\odot$), 1099 nodes (dashed line) and 2226 nodes (solid line).

if B is negative. There is a transition zone between the horizontal bottom and the beach, and the half width of the transition interval is l, which is a small quantity. In our simulations we have chosen $l = 0.01 \cot\theta$. The hill is of Gaussian shape with height a, radius r_0 and centre at (x_0, y_0)

$$\gamma(x, y) = a \exp\left[-\frac{(x - x_0)^2 + (y - y_0)^2}{r_0^2/4}\right] \tag{19}$$

The bathymetries of the headland ($a = 0$, $B > 0$) and the bay with a hill ($a > 0$, $B < 0$) are shown in Figure 12. All input parameters and grid characteristics for the simulations are listed in Table I. For both cases, grid refinement tests have been performed, with approximately twice as many nodes in the fine grid, as compared to the coarse one. Figure 13 shows the initial coarse grids for both depth bathymetries.

The case with the hill geometry becomes severely infested with noise, unless smoothing is applied. Therefore, we employ the artificial diffusion with $\kappa = 1.0 \times 10^{-3}$. Both wavelengths and depth gradients are similar to those in the plane case of long wave run-up (refer to Section 3.3), and an indication of the influence of the diffusion may then be inferred from Figure 8(a).

In principle, the presence of sidewalls extend the computational domain to an infinite region with a periodic bathimetry and repeated lines of symmetry. Furthermore, the grid generator traverses the nodes in a given sequence and both the triangulation and the solutions depend slightly on this. Therefore, we may observe deviations from perfect symmetry around $y = 0$, in particular for the coarser grids.

The geometry consisting of a hill on the beach offers some resemblance with a standard problem in the run-up literature, namely that of run-up of solitary waves on a conical island. Experiments were reported by Briggs *et al.* [47] and the first theoretical approach by

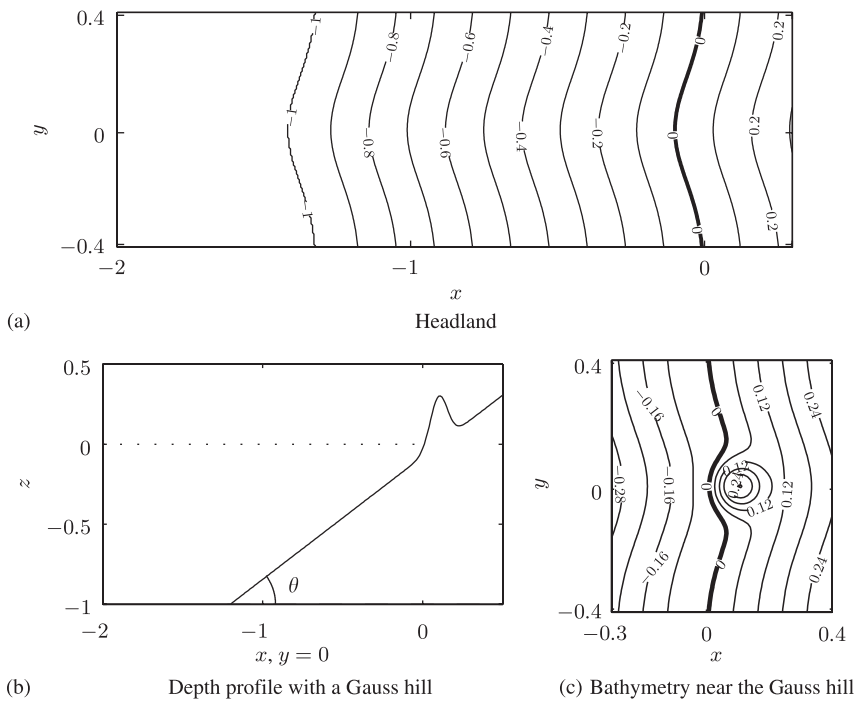

Figure 12. Two different geometries: (a) is the headland; (b) and (c) present the bay with a Gauss hill. The dotted and the bold lines indicate the still water level.

Table I. Parameters for the wave piston, the headland/bay bathymetry, and the Gauss hill, as used in the run-up simulations.

Parameter	Case 1: Headland	Case 2: Bay with Gauss hill
T	3	3
V_P	0.05	0.05
B	0.1	−0.1
r	0.25	0.25
θ	37.5	37.5
a	0	0.3
x_0, y_0	—	0.1, 0
r_0	—	0.15
α	3.0	2.0 (coarse grid) 2.5 (fine grid)
δ	0.5	0.5
κ	0.001	0.001
Coarse grid	4732 nodes, $\Delta x \approx 0.021$, $\Delta t = 7.1 \times 10^{-3}$	
Fine grid	10 336 nodes, $\Delta x \approx 0.013$, $\Delta t = 2.6 \times 10^{-3}$	

Note: All parameters are non-dimensional, except for the angle θ which is given in degrees. Δx is a typical element length. The δ parameter refers to Equation (10).

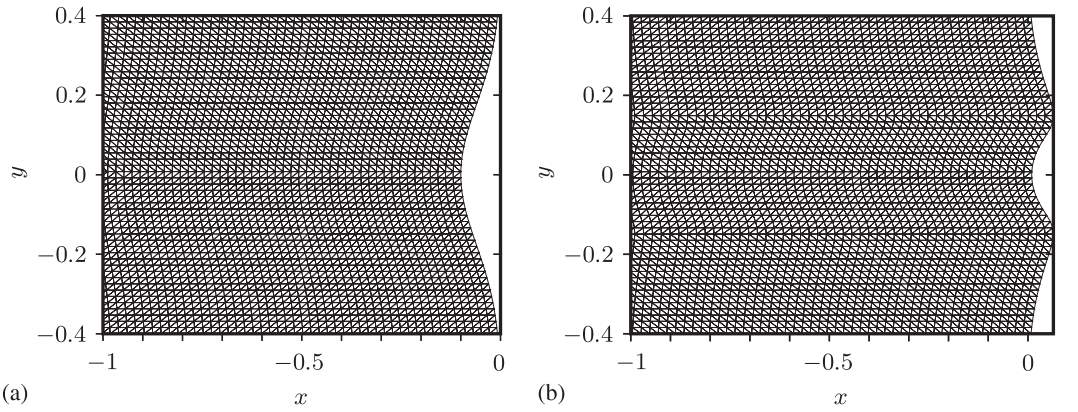

Figure 13. The initial coarse grids (4732 nodes) for the run-up simulations shown for $x \geqslant -1$: (a) headland; and (b) bay with a Gauss hill.

Liu *et al.* [22]. This case has served as a kind of benchmark problem for recent numerical models [24, 30, 31, 48]. Even though the problem has been addressed with hydrostatic models, accurate predictions require a Boussinesq type model for propagation of the incident wave. On the other hand, the run-up may be fairly well described by linear theory, save the interference phenomenon on the rear side of island. Adding that this case does not involve merging of shorelines, we have regarded it as a less appropriate test in the present context.

The wave run-up on the headland is shown in Figure 14 at four selected times, $t = \{3.3, 4.2, 4.6, 5.9\}$. Plots for the fluid velocity at $t = 3.3$ shows the wave focusing on the headland. At a later stage, $t = 4.2$, the maximum wave elevation on the headland occurs and the maximum wave elevation is four times the undisturbed, non-dimensional wave amplitude of 0.05. Since the headland width is rather narrow compared to the length of the incident wave, the focusing is not very strong. Instead the wave will spread around the headland, and continue to inundate the beach near both walls reaching its maximum wave elevation of 0.233 at $t = 4.6$ which is 4.66 times higher than the undisturbed wave amplitude. After reaching this maximum, the wave will withdraw from the beach and there will be a secondary run-up at a later stage. This is shown at $t = 5.9$, where we observe a second run-up on the headland while the fluid at both walls are retreating down from the beach. A comparison between the coarse and the fine grid is shown for the boundary nodes in the left column and only small differences are seen. Furthermore, the maximum run-up for both grids are shown in Figure 16 where only small differences of approximately one percent are seen. The maximum run-up in the centre of the tank is 0.205, while it is 0.230 at the walls. The same figure also shows the maximum run-up for the bay with a Gauss hill for both grid resolutions. Differences of up to five percent are observed for this bathymetry with maximum run-up of 0.228 at $y = \pm 0.2$.

Figure 15 displays the wave run-up in a bay with a Gauss hill on the beach for the times $t = \{3.6, 4.2, 4.8, 5.1\}$. The figure illustrates the process of generating a hole in the computational domain as well as the wave run-up. At $t = 3.6$ the wave is approaching the hill, and at $t = 4.2$ the wave has almost surrounded it. Afterwards, the shorelines from each side will merge behind the hill. The water will continue to flow around the hill building up a mound of

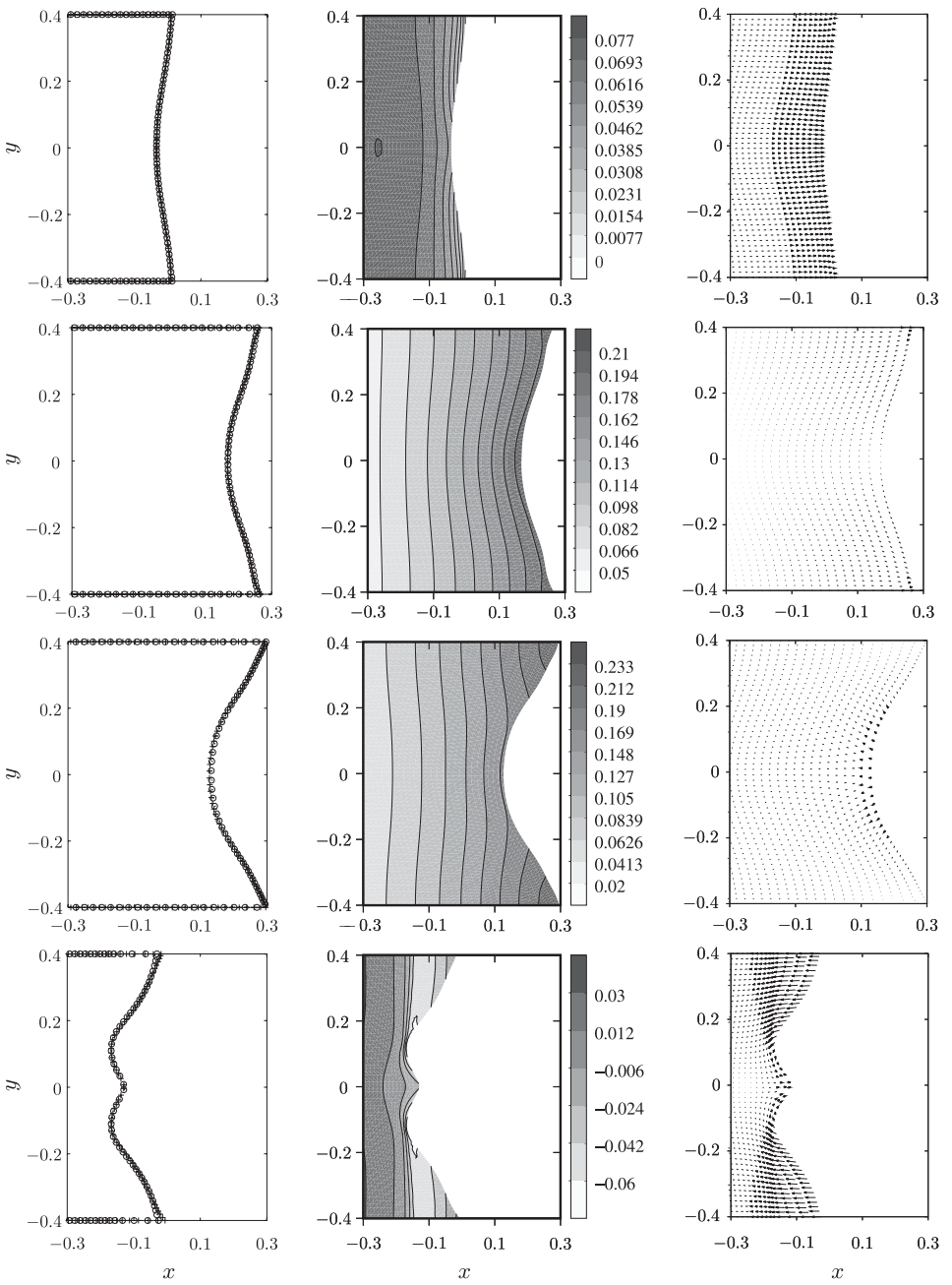

Figure 14. Run-up on a beach with headland shown for $t = 3.3$ (upper row), $t = 4.2$, $t = 4.6$ and $t = 5.9$ (bottom row). The left column shows the boundary nodes for two grids: + (10 336 nodes); $\odot$ (4732 nodes), the middle column shows the wave elevation, and the right column shows the vector plots for the fluid velocity.

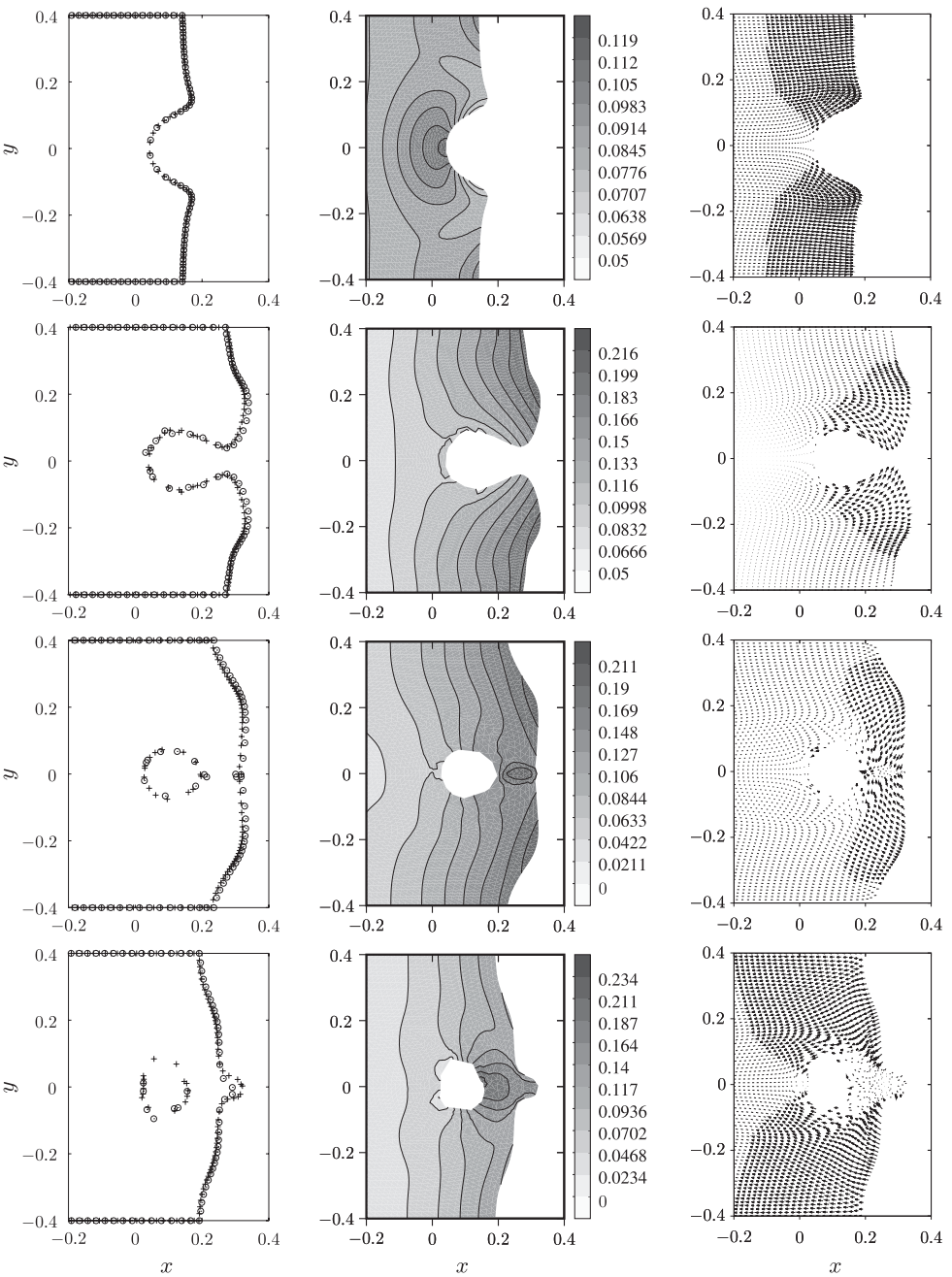

Figure 15. Run-up in a bay with a Gauss hill shown for $t = 3.6$ (upper row), $t = 4.2$, $t = 4.8$ and $t = 5.1$ (bottom row). The left column shows the boundary nodes for two grids: + (10 336 nodes); $\odot$ (4732 nodes), the middle column shows the wave elevation and the right column shows the vector plots for the fluid velocity.

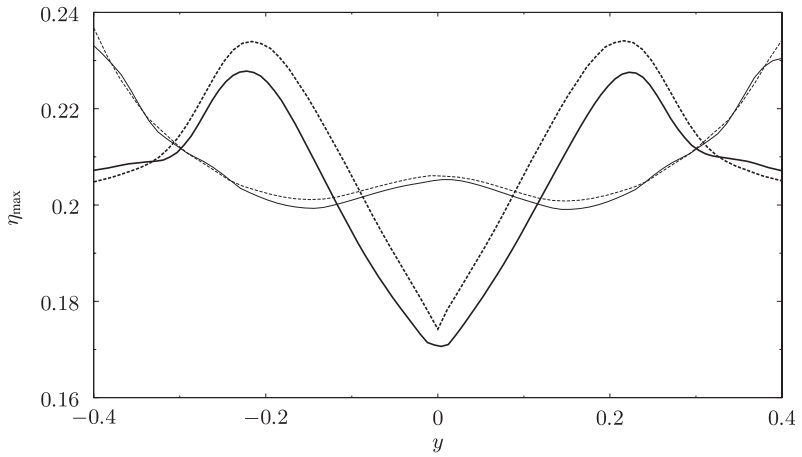

Figure 16. Maximum run-up, $\eta+h=0$, on the headland profile (fine grid: solid line; coarse grid: dashed line) and the bay with a Gauss hill (fine grid: bold solid line; coarse grid: bold dashed line).

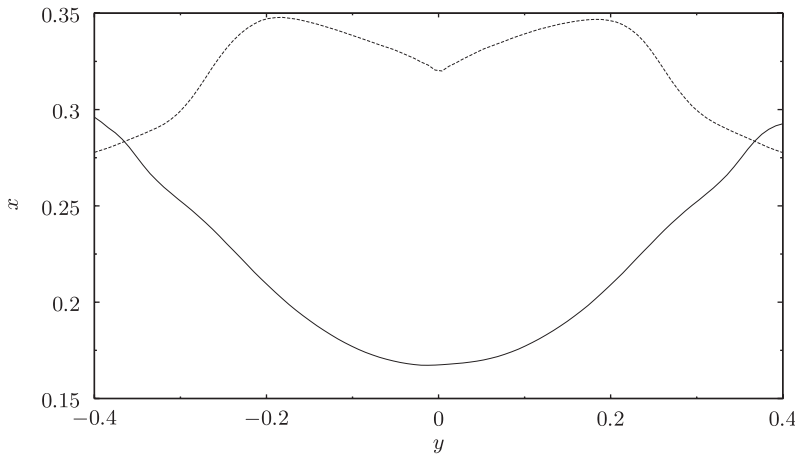

Figure 17. Maximum inundation on the headland profile (solid line) and the bay with a Gauss hill (dashed line).

water, which is seen for $t=4.8$. While the fluid withdraws elsewhere, this pile will produce a secondary run-up, as seen for $t=5.1$. Physically this is similar to a local dam break. Like the headland case, a grid refinement test has been accomplished. The boundary nodes at the beach are shown in the left column of Figure 15 for both the coarse and the fine grid. Again, good general agreement is observed. Still, we notice the local deviations in the secondary run-up and the poor representation of the dry region caused by the hill for the coarse grid, refer to Figure 18. One cure would be to insert new nodes in regions where a fine resolution is needed, as discussed in Section 2.2. This would have to be done in a systematic manner,

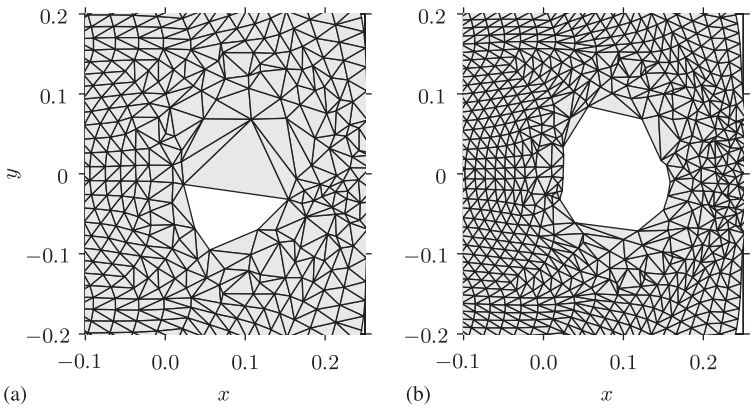

Figure 18. Two grids at $t = 5.1$ for run-up in a bay with a Gauss hill: (a) coarse grid; and (b) fine grid.

based on local grid parameters. We believe that this is feasible, but far from straightforward. An alternative would be to perform a total re-meshing at times, but this is in not in line with the general idea of the PFEM method. A third option would be to map regions with sparsity of nodes back on the initial grid, and then refine these regions from the beginning in a new simulation. This is the simplest procedure, but is probably useful only when the regions that require higher resolutions are few and easily detected, which points to simple bathymetries and short simulation times. None of the options have been systematically investigated in the present work.

Following the shoreline, $\eta + h = 0$, it is possible to extract the maximum run-up and inundation on the beach. The maximum run-up across the wave tank is shown in Figure 16 for both geometries. For the headland case, the maximum run-up occur at the centre of the tank and at the walls. Not surprisingly, the situation is different for the bay with a Gauss hill, which causes the wave to travel around it, and the maximum run-up occurs for approximately $y = \pm 0.2$. The corresponding maximum inundation is shown in Figure 17 for both beaches. For the headland case the maximum inundations occur at the walls. It is worth noticing that the local maximum run-up in the centre of the tank, refer to Figure 16, does not correspond to a maximum for the inundation due to the beach slope steepness. For the bay with a hill, the maximum inundation takes place at the same locations as the maximum run-up.

The PFEM does not conserve volume exactly. We define the normalized volume deficit as $\tilde{M}(t) = (M(t) - M(0))/M_{\mathrm{R}}(t)$, where $M(t)$ is the total volume at time t, and $M_{\mathrm{R}}(t)$ is a reference volume chosen as the volume displaced by the wave piston. M_{R} equals the horizontal displacement of the paddle, given as the time integral of (16) from 0 to t, muliplied with the paddle width and the equilibrium depth. Naturally, M_{R} corresponds to the net elevated volume under the incident wave. The quantity $\tilde{M}(t)$ is displayed in Figure 19 for both depth bathymetries and grid resolutions. For the case of the headland bathymetry, Figure 19(a), the volume is almost unaffected by the grid resolution. From the figure it is observed that the volume is relatively constant for $t < 3$. After $t = 3$ the wave starts running up the beach

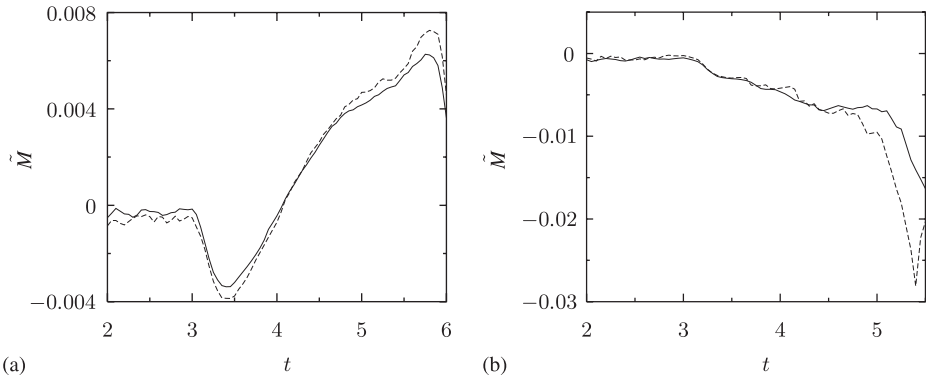

Figure 19. Mass conservation $\tilde{M}(t) = (M(t) - M(0))/M_R(t)$ for long wave run-up on two different beach profiles. The dashed line is for the coarse discretization and the solid line is for the fine discretization, refer to Table I: (a) headland; and (b) bay with a Gauss hill.

and the volume decreases first before it starts increasing at about $t = 3.5$. At the end of the simulation, $\tilde{M}$ is increased to roughly 0.005.

The situation changes slightly for the bay with a hill, refer to Figure 19(b). For $t < 4.5$ the volume is unaffected by the grid resolution. However, after this time the two curves differ and the coarse grid results in a greater loss of volume than the fine grid. Before $t = 5.5$ the volume, $\tilde{M}(t)$, is reduced by 3.0% at most for the coarse grid and 1.6% for the fine grid. Physically, the fluid has just enclosed the hill at $t = 4.5$ and afterwards the fluid builds up a pile of water behind the hill, refer to Figure 15. This suggests that the node distribution in the coarse grid is too coarse to capture the correct shoreline, resulting in a loss of volume.

5. CONCLUSIONS

In short, the numerical model is composed of four parts: the finite element method, boundary capturing by using alpha shapes, a time marching scheme, and dynamic meshing.

As demonstrated in the comparisons with other models and analytic solutions, the method computes simpler shoreline motions with good accuracy. Moreover, in the more demanding example with a hill on the beach, it demonstrates the ability to reproduce more complicated dynamics as merging shorelines and the presence of dry patches of higher land surrounded by fluid. Such features are not likely to be reproduced by the traditional Lagrangian, or ALE, techniques referenced in the introduction.

However, the application of the alpha shape concept is not without pitfalls. It is not trivial to identify the correct boundary nodes. A rather fine grid is required to yield the correct shoreline and run-up dynamics. Coarse grids easily lead to wrong connectivity of the fluid boundary, and even spurious 'holes' within the fluid in regions with large dilution or stretching of the particle distribution. These problems will also be present in other applications of the method, such as Navier–Stokes modelling. One remedy might be to combine the application

of the alpha shape concept with physical characteristics of the boundary, which, in our case, is a vanishing fluid depth.

As formulated here, the method does not conserve mass in an exact manner. Yet we have no indications of this being a serious drawback for the present kind of applications. The mass deficits are small and mainly linked to the shoreline motion. This is a feature which the present method has in common with several other new techniques for long wave run-up, as for instance References [24, 30, 31]. One should also bear in mind that in a real case of tsunami run-up, the flooded beach generally must be regarded as porous and a significant loss of volume in the swash may follow.

Formally, the PFEM formulation is of first-order in the time increment and second-order in space, save for effects due to application of the alpha shape concept. The former implies employment of very small time increments, and should be improved. Application of higher order elements on the other hand, is far from straightforward within the frame of PFEM. Improved spatial accuracy should instead be sought by adaptive refinement and through introduction of new nodes, particularly near the shore. We have so far not exploited this possibility, but the PFEM method should be better suited for such approaches than a traditional FEM method, let alone the FDM techniques.

One interesting aspect for future development is to include wave dispersion in a Boussinesq type model. This should be fairly straightforward. Furthermore, it is also desirable to implement a robust technique for handling of wave breaking. An attractive choice would then be the method of a wave-steepness dependent diffusion which has recently been employed with promising results [30, 31].

In its present form, the PFEM method has shortcomings, particularly concerning the boundary detection. Still, it yields promising results for cases where traditional methods (e.g. standard Lagrangian methods) easily fail. Thus it seems worthwhile to explore the method further. Two lines of improvements immediately present themselves, namely adaptive refinement and the combination of the alpha shape concept with physical properties of the boundary.

ACKNOWLEDGEMENTS

The authors thank Det Norske Veritas (DNV) for support through their educational funds. They also acknowledge the use of J.R. Shewchuk's Triangle program for mesh generation, which is freely available from Netlib, `www.netlib.org`.

REFERENCES

1. Meyer RE. On the shore singularity of water wave theory. Part 1. The local model. *Physics of Fluids* 1986; **19**:3142–3163.
2. Meyer RE. On the shore singularity of water wave theory. Part 2. Small waves do not break on gentle beaches. *Physics of Fluids* 1986; **19**:3164–3171.
3. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 1965; **8**(12):2182–2189.
4. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
5. Lin P, Liu PL-F. A numerical study of breaking waves in the surf zone. *Journal of Fluid Mechanics* 1998; **359**:239–264.
6. Lin P, Liu PL-F. Turbulence transport, vorticity dynamics, and solute mixing under plunging breaking waves in surf zone. *Journal of Geophysical Research* 1998; **103**(C8):15677–15694.
7. Osher S, Sethian J. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.

8. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375–389.
9. Lucy LB. A numerical approach to testing the fission hypothesis. *Astronomical Journal* 1977; **82**(12): 1013–1924.
10. Idelsohn SR, Oñate E, Calvo N, Del Pin F. The meshless finite element method. *International Journal for Numerical Methods in Engineering* 2003; **58**(6):893–912.
11. Idelsohn SR, Oñate E, Del Pin F. A Lagrangian meshless finite element method applied to fluid–structure interaction problems. *Computers and Structures* 2003; **81**:655–671.
12. Idelsohn SR, Oñate E, Del Pin F, Calvo N. Lagrangian formulations: The only way to solve some free-surface fluid mechanics problems. In *Fifth World Congress on Computational Mechanics*, Mang HA, Rammerstorfer FG, Eberhardsteiner J (eds). Vienna, Austria, 2002.
13. Edelsbrunner H, Mücke EP. Three-dimensional alpha shapes. *ACM Transactions on Graphics* 1994; **13**(1): 43–72.
14. Carrier GF, Greenspan HP. Water waves of finite amplitude on a sloping beach. *Journal of Fluid Mechanics* 1958; **4**:97–109.
15. Synolakis CE. The run-up of solitary waves. *Journal of Fluid Mechanics* 1987; **185**:523–545.
16. Keller HB, Levine AD, Whitham GB. Motion of a bore over a sloping beach. *Journal of Fluid Mechanics* 1960; **7**:302–316.
17. Shen MC, Meyer RE. Climb of a bore on a beach, 3: run-up. *Journal of Fluid Mechanics* 1963; **16**:113–125.
18. Thacker WC. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics* 1981; **107**:499–508.
19. Sielecki A, Wurtele MG. The numerical integration of the nonlinear shallow-water equations with sloping boundaries. *Journal of Computational Physics* 1970; **6**:219–236.
20. Hibberd S, Peregrine DH. Surf and run-up on a beach: a uniform bore. *Journal of Fluid Mechanics* 1979; **95**:323–345.
21. Kowalik Z, Murty TS. Numerical simulation of two-dimensional tsunami run-up. *Marine Geodesy* 1993; **16**: 87–100.
22. Liu PL-F, Cho Y-S, Briggs MJ, Kânoğlu U, Synolakis CE. Runup of solitary waves on a circular island. *Journal of Fluid Mechanics* 1995; **302**:259–285.
23. Sleigh PA, Gaskell PH, Berszins M, Wright NG. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. *Computers & Fluids* 1998; **27**(4):479–508.
24. Titov VV, Synolakis CE. Numerical modeling of tidal wave runup. *Journal of the Waterway*, *Port*, *Coastal*, *and Ocean Engineering* 1998; **124**(4):157–171.
25. Hu K, Mingham CG, Causon DM. Numerical simulation of wave overtopping of coastal structures using the non-linear shallow water equations. *Coastal Engineering* 2000; **41**:433–465.
26. Brocchini M, Bernetti R, Mancinelli A, Albertini G. An efficient solver for nearshore flows based on the WAF method. *Coastal Engineering* 2001; **43**:105–129.
27. Hubbard ME, Dodd N. A 2D numerical model of wave run-up and overtopping. *Coastal Engineering* 2002; **47**:1–26.
28. Prasad RS, Svendsen IA. Moving shoreline boundary condition for nearshore models. *Coastal Engineering* 2003; **49**(4):239–261.
29. Madsen PA, Sörensen OR, Schäffer HA. Surf zone dynamics simulated by a Boussinesq-type model: Part I. Model description and cross-shore motion of regular waves. *Coastal Engineering* 1997; **32**:255–287.
30. Kennedy AB, Chen Q, Kirby JT, Dalrymple RA. Boussinesq modeling of wave transformation, breaking, and run-up. Part I: 1D. *Journal of the Waterway*, *Port*, *Coastal*, *and Ocean Engineering* 2000; **126**(1):39–47.
31. Lynett PJ, Wu T-R, Liu PL-F. Modeling wave runup with depth-integrated equations. *Coastal Engineering* 2002; **46**:89–107.
32. Pedersen G, Gjevik B. Run-up of solitary waves. *Journal of Fluid Mechanics* 1983; **135**:283–299.
33. Johnsgard H, Pedersen G. A numerical model for three-dimensional run-up. *International Journal for Numerical Methods in Engineering* 1997; **24**:913–931.
34. Zelt JA, Raichlen F. A Lagrangian model for wave-induced harbour oscillations. *Journal of Fluid Mechanics* 1990; **213**:203–225.
35. Johns B. Numerical integration of the shallow water equations over a sloping shelf. *International Journal for Numerical Methods in Engineering* 1982; **2**:253–261.
36. Shi F, Sun W. A variable boundary model of storm surge flooding in generalized curvilinear grids. *International Journal for Numerical Methods in Engineering* 1995; **21**:641–651.
37. Özkan-Haller HT, Kirby JT. A Fourier–Chebyshev collocation method for the shallow water equations including shoreline run-up. *Applied Ocean Research* 1997; **19**:21–34.
38. Zhou X, Hon YC, Cheung KF. A grid-free, nonlinear shallow-water model with moving boundary. *Engineering Analysis with Boundary Elements* 2004; **28**(8):967–973.

39. Jensen A, Pedersen G, Wood DJ. An experimental study of wave run-up at a steep beach. *Journal of Fluid Mechanics* 2003; **486**:161–188.
40. Mei CC. *The applied dynamics of ocean surface waves*. Advanced Series on Ocean Engineering—vol. 1. World Scientific: Singapore, 1989.
41. Diffpack. Object oriented numerical framework for the solution of PDEs. http://www.diffpack.com
42. Langtangen HP. *Computational Partial Differential Equations* (2nd edn). Springer: Berlin, 2003.
43. Guibas L, Stolfi J. Primitives for the manipulation of general subdivisions and the computation of Voronoï diagrams. *ACM Transactions on Graphics* 1985; **4**(2):74–123.
44. Shewchuk JR. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry*, Lin MC, Manocha D (eds). Lecture Notes in Computer Science, vol. 1148. Springer: Berlin, 1996; 203–222.
45. Shewchuk JR. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications* 2002; **22**(1–3):21–74.
46. Stoker JJ. *Water waves*. Interscience, 1957.
47. Briggs MJ, Synolakis CE, Harkins GS, Green DR. Laboratory experiments of tsunami runup on circular island. *Pure and Applied Geophysics* 1995; **144**(3/4):569–593.
48. Yeh H, Synolakis CE, Liu PL-F. *Long-wave runup models*. World Scientific Publishing Co: River Edge, NJ, Singapore, London, Hong Kong, 1996.

