INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **63**:69–95
Published online 21 May 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2079

Finite volume simulation of viscous free surface waves using the Cartesian cut cell approach

W. Bai*,†, C. G. Mingham, D. M. Causon and L. Qian

Department of Computing and Mathematics, Centre for Mathematical Modelling and Flow Analysis, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, U.K.

SUMMARY

The application of the Cartesian cut cell approach in the numerical simulation of two-dimensional viscous free surface flows is described. The Arbitrary Lagrangian–Eulerian method is adopted to update the moving free water surface in a semi-Lagrangian scheme, in which a finite volume method of second-order accuracy in space is used for solving the flow field based on an Eulerian description at each time step. The cut cell approach is employed to track the free surface and solid boundaries across a stationary background Cartesian grid covering the whole fluid, air and solid regions. In this approach, the cells full of air and solid are not calculated explicitly, and apart from the fluid cells, cut cells and merged cells are treated separately in terms of corresponding boundary conditions. In order to validate the present numerical method, current flow past a circular cylinder at various low Reynolds numbers and wave sloshing in a rectangular container are tested first. Further numerical results are obtained for the propagation of regular waves and a wave passing over a submerged dike. The model is also applied to the simulation of radiation waves induced by a forced oscillating submerged circular cylinder. The results indicate that the present numerical model using the Cartesian cut cell approach is highly efficient for solving the wave fields, and fully automatic for generating boundary fitted meshes. These features are particularly useful for moving boundary problems in a larger computational domain and with a longer simulation time. Copyright © 2009 John Wiley & Sons, Ltd.

Received 31 October 2008; Revised 14 February 2009; Accepted 4 April 2009

KEY WORDS: Cartesian cut cell; arbitrary Lagrangian–Eulerian method; viscous wave; finite volume method; free surface; SIMPLE algorithm

*Correspondence to: W. Bai, Department of Civil Engineering, National University of Singapore, 1 Engineering Drive 2, Singapore 117576, Singapore.
†E-mail: cvebw@nus.edu.sg

Contract/grant sponsor: EPSRC; contract/grant number: EP/D034566/1

1. INTRODUCTION

Considerable research effort has been devoted to developing numerical models for free surface flows over the last three decades. Generally speaking, two kinds of numerical models have been proposed to investigate this problem. One is developed on the basis of the inviscid and irrotational flow assumption, and the resulting Laplace equation is solved by means of the well-established boundary or finite element methods. For example, Bai and Eatock Taylor [1, 2] simulated fully nonlinear waves radiated by a forced oscillating cylinder, wave propagation in a wave tank and wave diffraction around a cylinder using a higher-order boundary element method. The alternative finite element method was used by Turnbull *et al.* [3], Wu and Hu [4] for investigations of wave diffraction problems. In addition, Grilli *et al.* [5] and Xue *et al.* [6] calculated three-dimensional overturning waves in a numerical wave tank. It has been shown that the potential flow model is able to describe unsteady gravity waves of finite amplitude, which can be simulated accurately with all nonlinear effects and without any significant damping. However, this type of model cannot predict rotational velocity fields and the effects of viscosity or turbulence due to its inherent assumptions.

In parallel with the rapid growth of computer power, the other model employing the Navier–Stokes or Euler equations without further simplification has been applied more and more widely. It seems that solutions to the unsteady Navier–Stokes or Euler equations, including a fully nonlinear free surface description, have usually suffered from relatively large numerical errors which do not allow accurate long-term simulations of water waves. At the same time, the accuracy of the simulation depends on calculating the correct position of the air/water interface throughout the wave motion, and this becomes especially difficult when the wave overturns and merges with the water surface or when the interface breaks up into spray. Therefore, the development of increasingly accurate and efficient numerical models for viscous free surface waves remains a challenge.

Various techniques by which to predict the position of a moving free water surface can be divided roughly in two distinct categories: interface capturing and interface tracking. Interface capturing is based on the solution of the equations in a fixed-in-time domain, and usually the interface is achieved either by following massless particles introduced into the liquid phase near the free surface initially or by solving a transport equation for the void fraction of the liquid phase. The volume of fluid method [7–9], the MAC method [10, 11] and the level set method [12] are the most generally used methods of interface capturing. Moreover, Qian *et al.* [13] and Gao *et al.* [14] developed a conservative, fully coupled numerical algorithm to simulate free surface flows based on the solution of the density distribution in the whole computational domain directly. Interface capturing can easily handle much distorted free surface profiles including jets and spray at solid boundaries and breaking waves. The major drawbacks of this method are its tendency to smear the interface and the high CPU cost due to the need for fine grids and small time steps.

On the other hand, interface tracking is based on a moving mesh approach. The arbitrary Lagrangian–Eulerian (ALE) method is an example of interface tracking [15, 16] in which the grid points are moved independently of the fluid motion to obtain elements of proper shape to avoid overly distorted meshes. It gives qualitatively accurate free surface shapes and it is easy to implement the free surface boundary conditions. However, the mesh usually deforms severely as the free surface moves, making remeshing necessary at almost all time steps, and a robust remeshing algorithm is therefore essential for the success of the ALE method. Determination of the mesh velocity is another major problem with the ALE method. Different techniques have been developed for updating the mesh, depending on the fluid domain. For problems involving simple domains, the mesh velocity can be deduced through a uniform or non-uniform distribution of

DOI: 10.1002/fld

nodes along straight lines ending at the moving boundaries [17, 18]. For general computational domains, the mesh velocity can be computed through partial differential equations with appropriate boundary conditions [19, 20].

From the above discussion, it may be observed that a fixed Cartesian grid is usually used in interface capturing; hence, problems associated with grid generation are avoided. In addition, this method can exploit the simplicity and availability of well-established flow solvers developed for fixed structured grids. On the other hand, a boundary-conforming grid arrangement adopted in interface tracking is convenient and accurate; the free surface boundary conditions are applied at the exact locations without any smearing or redistribution. These attributes provide the motivation for the present work: to achieve a combination of the best features of the interface capturing and interface tracking methods. In this paper, the flow field is solved on a fixed structured Cartesian grid, and the free water surface overlying the fixed grid is explicitly tracked using the semi-Lagrangian method and a Cartesian cut cell grid.

The Cartesian cut cell approach is an effective alternative to the traditional structured and unstructured grids. In this approach, solid regions are simply cut out of a stationary background Cartesian mesh, and their boundaries are represented by different types of cut cells. As a result, the union of all solid cells, fluid cells and partially cut cells comprises a single Cartesian mesh. The cut cell mesh generation is relatively straightforward through calculations for the boundary segment intersections with the background Cartesian mesh. Furthermore, moving boundaries can be easily accommodated by recomputing cell-boundary intersections, rather than remeshing the whole flow domain or large portions of it. This method has recently been applied successfully to the shallow water equations [21–23], to the Boussinesq equations [24], to the Euler equations [25, 26] and extended to deal with incompressible flows [27, 28]. However, all these previous works generated cut cell elements only around solid bodies. In this study, the developed cut cell solver has the capacity to tackle both fixed/moving solid bodies and free water surfaces, which means that the air region above the free surface is treated as another type of solid body but with free surface boundary conditions.

The present paper develops the numerical solution to the Navier–Stokes equations by means of a cell-centered finite volume method based on a structured Cartesian grid with a colocated variables arrangement. A second-order scheme with deferred correction is used to discretize the convective fluxes in conjunction with a second-order central difference scheme (CDS) adopted for the diffusive fluxes. The pressure–velocity coupling is evaluated by using the SIMPLE algorithm. Owing to the use of the Cartesian cut cell approach, the algorithm performed at cut cells and merged cells is adjusted accordingly to account for the influence of boundary conditions. A time-stepping scheme is employed in the semi-Lagrangian frame to update the position of the free surface at the next time step. Numerical results are first obtained for current flow past a circular cylinder. Next, sloshing waves, waves generated by a piston-like wave maker and travelling waves over a submerged bar in a two-dimensional wave tank, are simulated. Finally, new results are given for the waves induced by a forced oscillating submerged circular cylinder. Comprehensive comparisons indicate that the present numerical results are in satisfactory agreement with analytical or experimental data.

2. MATHEMATICAL FORMULATION

In a right-handed Cartesian coordinate system Oxy having y-axis pointing vertically upwards, two-dimensional incompressible viscous flow is governed by the continuity equation and the

DOI: 10.1002/fld

Navier–Stokes equations,

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{1}$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)-g \tag{3}$$

where $\mathbf{u}$ is the fluid velocity vector whose Cartesian components in the x and y directions are u and v, respectively, ρ and ν are the density and kinematic viscosity of the fluid, p denotes the total pressure and g is the acceleration due to gravity. It should be noted that the definition of the origin O is relatively arbitrary, depending on different problems considered. For free surface problems, it is always helpful to locate the origin O on the mean water surface, in order to measure the actual wave oscillations. One can notice that the total pressure p can be further divided into two components: $p=p_{\mathrm{H}}+p_{\mathrm{D}}$, where p_{H} is the hydrostatic pressure defined as $p_H=-\rho \mathrm{g}y$ and p_{D} is the dynamic pressure. By this division, the last term in Equation (3) can be eliminated, such that the dynamic pressure becomes the unknown instead of the total pressure. For the purposes of convenience, the dynamic pressure p_{D} remains expressed as p in the following parts of the paper.

The flow is also subject to various boundary conditions on all surfaces of the fluid domain. On the instantaneous free water surface, the physical condition that there will be no mass flux across the free surface requires the surface to move and deform as a material surface, which yields the kinematic condition in the Lagrangian form,

$$\frac{\mathrm{D}\mathbf{X}}{\mathrm{D}t}=\mathbf{u} \tag{4}$$

where $\mathrm{D}/\mathrm{D}t$ is the usual material derivative and $\mathbf{X}$ denotes the position of points on the free surface. In the ALE method, as the nodes on the free surface move along with the prescribed velocity $\mathbf{U}$ rather than the fluid velocity $\mathbf{u}$, the Lagrangian form of the kinematic condition can be rewritten as

$$\frac{\delta\mathbf{X}}{\delta t}=\mathbf{u}-(\mathbf{u}-\mathbf{U})\cdot\nabla\mathbf{X} \tag{5}$$

where $\delta/\delta t$ is the total derivative following the moving node; this is also referred here as the semi-Lagrangian description. In this paper, the prescribed velocity is restricted to the horizontal direction, and nodes are only allowed to move vertically. In addition, a single-valued height function $\eta(x,t)$ is defined to evaluate the free surface position, which has been proven to be well suited for small or moderate non-breaking waves investigated here. By doing so, the resulting expression for the kinematic condition is

$$\frac{\delta\eta}{\delta t}=v-u\frac{\partial\eta}{\partial x} \tag{6}$$

Another condition applied on the free surface is the dynamic one, which requires that the stresses on the free surface are balanced in both normal and tangential directions. By ignoring the shear

DOI: 10.1002/fld

stress and simplifying the normal shear, as is done in many applications, the dynamic condition used in this paper can be taken as $p=\rho g\eta$. We recall that p here is the dynamic pressure.

On the surfaces of rigid side walls and solid bodies slip and non-slip boundary conditions are applied, respectively, and the fluid velocity is specified at the inlet boundary. At the outlet boundary, the set condition should make the boundary 'transparent', i.e. the numerical solutions in the inner computational region would not be affected by the outlet boundary. Many forms of far field condition have been developed by other investigators. Here a periodic condition is introduced at the far field boundary for current flow problems. In simulations of right travelling water waves, an artificial damping layer on the free surface is adopted to absorb the wave energy near the right-hand end of the wave tank. At this numerical beach, the kinematic condition is modified by a damping term over a finite length of the free surface (Bai and Eatock Taylor [1]),

$$\frac{\delta\eta}{\delta t}=v-u\frac{\partial\eta}{\partial x}-\gamma(x)(\eta-\eta_0) \tag{7}$$

where $\gamma(x)$ is the damping coefficient and η_0 is a reference value specifying the at-rest position of a fluid particle. In practice, the damping coefficient is chosen to be continuous and continuously differentiable. Here we have used:

$$\gamma(x)=\begin{cases}\omega\left(\dfrac{x-x_0}{\lambda}\right)^2, & x\geqslant x_0=L_D-\lambda\\ 0, & x<x_0\end{cases} \tag{8}$$

In this definition, λ and ω are the representative wave wavelength and wave frequency (rad/s), respectively, and L_D is the position of the far field boundary. Lastly, a Neumann-type boundary condition needs to be introduced for the pressure on all the boundary surfaces, except at the free surface.

3. NUMERICAL METHOD

One important advantage of the present numerical model using the Cartesian cut cell approach is that it is possible to adopt well-tested flow solvers developed for structured Cartesian grids with only slight modifications for the cut cells and merged cells, respectively. In this section, we only briefly describe the cell-centered finite volume method, one well-established algorithm, for solving the Navier–Stokes or Euler equations in a rectangular fluid domain; more details can be found in many textbooks, such as Ferziger and Perić [29]. The cut cells and the merged cells issues will be dealt with in the next section.

The momentum equations are discretized first on the structured fluid cells. In each rectangular control volume (CV) P surrounding by the neighboring cells $P_j\,(j=1,\ldots,4)$, as shown in Figure 1, the integral form of the momentum equations is obtained through Gauss's theorem,

$$\int_{V_P}\frac{\partial\phi}{\partial t}\,\mathrm{d}v+\int_{S_P}\phi\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}s=-\frac{1}{\rho}\int_{S_P}p\mathbf{n}\,\mathrm{d}s+\int_{S_P}v\nabla\phi\cdot\mathbf{n}\,\mathrm{d}s \tag{9}$$

where ϕ is the general variable denoting u or v, V_P and S_P are the volume and surface of CV P respectively, and $\mathbf{n}$ is the normal unit vector on cell surfaces positive pointing out of CV P.

DOI: 10.1002/fld

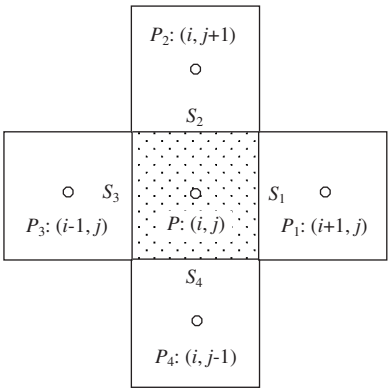

Figure 1. Assignment of finite volume cells and variables.

For the unsteady term in Equation (9), an implicit three-level scheme of second-order accuracy is adopted. A first-order upwind difference scheme (UDS) with deferred correction is simply used to approximate the convective fluxes. Using this method, the contribution of the first-order scheme will vanish, and second-order accuracy is achieved when the calculation is convergent. We evaluate the diffusive fluxes using the second-order CDS. By substituting these discrete representations into Equation (9), the final discretized momentum equations can be expressed as

$$a_0\phi_P + \sum_{j=1}^{4} a_j\phi_{P_j} = b_0 \tag{10}$$

where

$$a_j = \left[\min(\mathbf{u}_j\cdot\mathbf{n}_j, 0) - \frac{v}{(\mathbf{X}_{P_j}-\mathbf{X}_P)\cdot\mathbf{n}_j}\right] S_j \tag{11}$$

$$a_0 = -\sum_{j=1}^{4} a_j + \frac{V_P}{\Delta t}(1+0.5\lambda) \tag{12}$$

$$b_0 = \sum_{j=1}^{4}\left[-\frac{1}{\rho}p_j S_j\mathbf{n}_j - (\mathbf{u}_j\cdot\mathbf{n}_j)(\phi_j^{\mathrm{CDS}} - \phi_j^{\mathrm{UDS}})^{n-1}\right] + \frac{V_P}{\Delta t}[(1+\lambda)\phi_P^{n-1} - 0.5\lambda\phi_P^{n-2}] \tag{13}$$

Here, Δt is the time interval, $\mathbf{X}$ is the position of centre nodes, p_j and $\mathbf{u}_j$ present the pressure and velocity at the jth cell surface obtained by linear interpolation between the two adjacent nodes, and S_j denotes the area of the jth cell surface (see Figure 1). Superscript UDS and CDS stand for the variables approximated by the first-order UDS and second-order scheme, respectively, and superscript $n-1$ and $n-2$ denotes that the terms are evaluated using values obtained from the previous iteration and the iteration before the previous one, respectively. In addition, $\lambda=1$ indicates the implicit three-level scheme, and $\lambda=0$ corresponds to the implicit Euler scheme.

The velocity obtained from the solution to the momentum equations cannot be guaranteed to satisfy the continuity equation, which needs to be corrected subsequently by an appropriate algorithm. Here the SIMPLE algorithm is adopted, in which the cell surface velocity correction is

DOI: 10.1002/fld

defined based on the pressure correction. Substituting this definition into the continuity equation leads to the discretized pressure correction equation for p'_P,

$$a_0^p p'_P + \sum_{j=1}^{4} a_j^p p'_{P_j} = b_0^p \tag{14}$$

where

$$a_j^p = -\frac{(S_j)^2}{(a_0^u)_j}, \quad a_0^p = -\sum_{j=1}^{4} a_j^p, \quad b_0^p = -\sum_{j=1}^{4} (\mathbf{u}_j \cdot \mathbf{n}_j) S_j \tag{15}$$

In these equations, superscripts p and u indicate that the corresponding coefficients are of the pressure correction equation and the momentum equations, respectively. It should be noticed that, as a colocated variable arrangement is adopted to store all the variables at the same set of grid points, the velocity on the jth cell surfaces $\mathbf{u}_j$ in Equation (15) must be determined by momentum interpolation in order to avoid numerical instability,

$$\mathbf{u}_j = \overline{\mathbf{u}_j} - \overline{\left(\frac{1}{\rho a_0^u}\right)}_j \cdot [(p_{P_j} - p_P)\mathbf{n}_j - \overline{(\nabla p)_j}(\mathbf{X}_{P_j} - \mathbf{X}_P) \cdot \mathbf{n}_j] S_j \tag{16}$$

where the overbar means that the variables are obtained by linear interpolation based on the values at two adjacent nodes.

After obtaining the pressure correction p'_P, the pressure and velocities can be corrected by

$$p_P = p_P^* + p'_P, \quad \mathbf{u}_P = \mathbf{u}_P^* - \frac{V_P}{\rho a_0^u} \nabla p'_P \tag{17}$$

where the pressure p_P^* comes from the previous iteration, and the velocity $\mathbf{u}_P^*$ is the solution to the discretized momentum equations (Equation (10)). Under-relaxation is always used in the numerical simulation to avoid divergence of the iterative procedure. For the momentum equations, under-relaxation is used directly in calculating the coefficient matrix, while for the pressure under- relaxation is involved in the correction procedure. In this paper, the under-relaxation factors for the pressure and velocities are chosen to be 0.4 and 0.7, respectively. The procedure discussed above including the prediction and correction can be performed repeatedly until the numerical error is negligibly small, which will result in converged numerical solutions to the Navier–Stokes or Euler equations at each time step.

4. CARTESIAN CUT CELL APPROACH

A Cartesian cut cell mesh can be generated by cutting solid regions out of a background grid, and therefore three main types of cell are formed in the computational domain, namely the cut cell, fluid cell and solid cell, as shown in Figure 2. The algorithm discussed in the previous section needs to be modified, in order to be suitable for a cut cell mesh. In addition, some specific aspects related to free surface flows have to be considered. These are discussed next.

DOI: 10.1002/fld

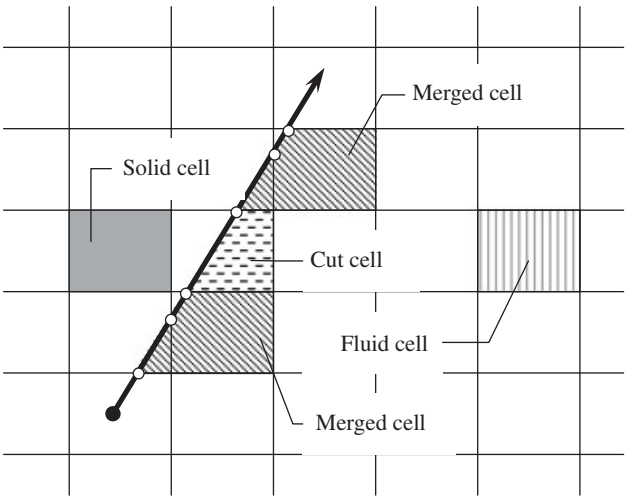

Figure 2. Sketch of definition for the cut cell mesh.

4.1. Mesh generation

To generate a cut cell mesh, the background Cartesian grid is constructed first, which could be either uniform or non-uniform. The geometry of the solid body is represented by a series of straight line segments defined in an anti-clockwise direction, and the next step is to find the intersection points between the background Cartesian grid lines and an individual line segment (see Figure 2). Once the grid cells containing the start and end points of the line segment have been identified and its slope computed, the required intersection points can be easily found; see Causon *et al.* [21] for more details. It should be noted that the air region above the free water surface is regarded as an individual solid body bounded by the side walls and the free surface represented by a series of straight line segments. The same algorithm used for a solid body can also be adopted to determine the intersection points between the free surface and the background grid lines. The cut cell method produces a boundary conforming mesh without the necessity to make the boundary a coordinate surface. In fact, there is no mesh generation in the conventional sense; all that is necessary is to calculate the intersections of a series of line segments with a background Cartesian grid.

After establishing all mesh intersections, the cells that intersect the boundary of a solid body are registered as cut cells. Sweeps across the background grid are then performed to identify which cells or rows of cells are bounded by cut cells; these are registered as solid cells. In the domains with free water surfaces, solid cells also include the upper part of cells above the corresponding cell cut by the free surface in each column. We assume here that each background cell is allowed to be cut only once, so that multiple cuts do not invalidate the data structure. When a sharp corner is situated inside a background cell, resulting in the case of a multiple cut, the solid region is re-defined by placing a straight line connecting the intersections with the background cell. By doing so, the sharp corner can be approximated and only one cut is required in the cell.

Since, in practice, a cut cell can be arbitrarily small, numerical stability may be compromised at these small cut cells. To overcome this problem cell merging is implemented. The basic idea is to combine several neighbouring cells together so that any interfaces between merged cells are ignored and computations can be performed at a newly combined larger cell without reducing the

DOI: 10.1002/fld

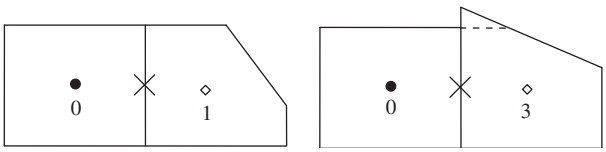

Figure 3. Illustration of east neighbour for a fluid cell (Type 0).

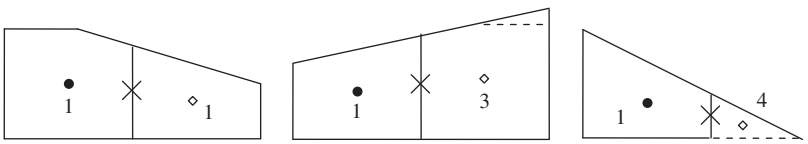

Figure 4. Illustration of east neighbour for a cut cell (Type 1).

global time step. To implement the cell merging technique, cells whose volumes are smaller than a minimum volume criterion $V_{\min}$, are merged with neighbouring cells. Suitable neighbouring cells should have the largest common interface with the merging cell. An example neighbouring cell and merging cell are shown in Figure 2. In our calculations, $V_{\min}$ is set to be one half of the fluid cell size. The total computational domain is composed of fluid cells (Type 0), cut cells (Type 1), solid cells (Type 2), merged cells (Type3) and small merging cells (Type 4).

4.2. Cut cells

Now, based on the definition of the five different types of cells, the algorithm developed in the previous section can be implemented without modification at cells marked Type 0 in the computational domain. Clearly, neighbours adjacent to a fluid cell could be a cut cell or a merged cell. For the purposes of illustration, a typical CV surface, the east surface will be considered in what follows; analogous situations may be derived straightforwardly for all other surfaces. Figure 3 shows these two possibilities, from which we can see that no additional treatments are necessary.

However, the cut cell and the merged cell need to be dealt with separately. For a cut cell Type 1, the discretized equation at the jth cell surface ($j=1,\ldots,4$) is based on the same algorithm developed for a fluid cell, with the free surface or non-slip boundary conditions at the cut surface corresponding to the free water surface or solid wall, respectively. We note that $S_j(j=1,\ldots,4)$ represents the area of the common interface and one or two could be zero depending on the position of the boundary, but this does not affect the equations given in the previous section. Figure 4 shows other possibilities for the east neighbour. From this figure, we can see that whether the east cell is a cut cell or a merged cell, the same discretized equations can be adopted without modification. However, at the small merging cell Type 4, because we do not wish to encounter a numerical instability issue mentioned above, a specific explicit treatment is used to avoid arranging unknowns on it. In this case, the coefficient for the small merging cell is moved to the right-hand side by multiplying the known value on this small cell from the previous iteration, instead of solving it implicitly.

4.3. Merged cells

For merged cell Type 3, the situation is more complex, as shown in Figure 5 that shows possibilities not included in Figures 3 and 4. In the first case of Figure 5, a merged cell is connected to both a

DOI: 10.1002/fld

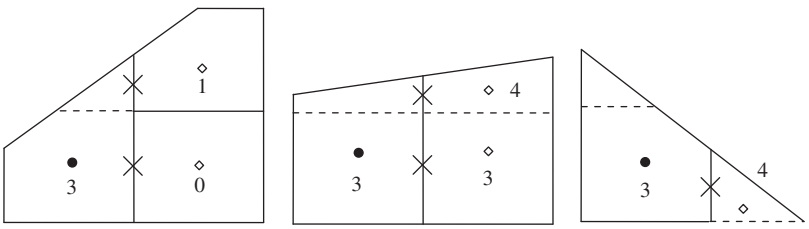

Figure 5. Illustration of east neighbour for a merged cell (Type 3).

fluid cell and a cut cell. Generally speaking, the flux balance at the common side should involve both the fluid cell and the cut cell; but this, of course, would destroy the structure of the resulting algebraic equation system. We wish to retain a well-organized 5-point diagonal coefficient matrix which can be solved very efficiently, while able to fully take advantage of the Cartesian cut cell approach. Therefore, we avoid elements not directly adjacent to a CV being involved in the discretized equation for the CV. The approach used here is to calculate the influence of the cut cell in the first case of Figure 5 explicitly, and only use the unknowns in the fluid cell. For the second case of Figure 5, a similar technique is used, in order to maintain the structure of the algorithm. In this case, we separate a merged cell into a fluid cell or a cut cell and a small merging cell, and calculate the small merging cell explicitly. The small merging cell in the last case of Figure 5 can also be treated explicitly, just like that discussed for the cut cell. The introduction of merged cells will result in some such explicit treatments, locally reducing the formal accuracy of the solver. However, we have found this compromise between formal accuracy and numerical efficiency to be acceptable in practice.

However, since moving boundary problems are considered here, merged cells must be used to ensure conservation of mass. When boundaries move relative to the background grid, cells near the moving boundaries undergo changes. If a solid cell changes to a cut cell, it is difficult to define where the fluid originates from without using merged cells. By introducing cell merging this difficulty is avoided. In such cases, V_P in the last terms of Equations (12) and (13) should be calculated accurately to keep the change of fluid mass continuous, based on the cell type at the current and previous time steps.

After calculating the discretized coefficients for each individual CV, we can finally set up a 5-point diagonal algebraic equation system for which many well-established solvers exist. As discussed above, solid cells and small merging cells are not computed explicitly. We can, therefore, set the coefficients for these cells as unity in the diagonal of the coefficient matrix and zero in the right-hand-side vector. The strongly implicit procedure is applied here (see Ferziger and Perić [29]), which has been proven to be very efficient for algebraic equation systems having a similar structure to that established in this paper. One remaining problem is: how to determine the values at the small merging cells. These can be simply set to those values at the corresponding merged cell to which the small merging cell belongs.

4.4. Free surface flows

Once the flow field is obtained, the free surface boundary condition can be used to update the position of the free surface at the next time step. The updating points are chosen to be at the centre of the top cell surface at the appropriate cut cell or merged cell, and their velocities are simply taken as those of the corresponding cut cell or merged cell. However, these velocities at the cut

DOI: 10.1002/fld

cell (or merged cell) are actually located at the cell centres, rather than at the cell surface. Various extrapolation techniques of first-order and second-order accuracy can be used to reconstruct the cell surface data, but our numerical calculations show that the practice does not improve the numerical results significantly. In our investigations, both the fourth-order Runge-Kutta scheme and the first-order Euler scheme are adopted for updating the free surface boundary condition in time.

Another important issue related to free surface flows is numerical instability. It is known that the use of merged cells will locally increase numerical errors in the calculation; this, in conjunction with other possible numerical errors, will be present in the local position of the free surface. The saw-tooth free surface may be induced due to the different types of adjacent cells on the free surface, which could affect the calculation of the flow field at the next time step. Numerical error will be magnified continuously, and sharp corners may appear on the free surface, finally leading to the computation ceasing prematurely. To avoid numerical instability, a 5-point smoothing technique is used to smooth the velocity and free surface every few time steps in this paper. The smoothed values at each node i on the free surface are calculated by the formula: $f_i = (-f_{i-2} + 4f_{i-1} + 10fi + 4f_{i+1} - f_{i+2})/16$, where subscripts $i-2, i-1, i+1, i+2$ denote the values at corresponding neighbouring cells.

5. NUMERICAL RESULTS

In this section, five different cases are considered to validate the present numerical model. The first case concerns current flow past a circular cylinder, and the second case is a sloshing wave in a wave tank. The third and fourth cases simulate wave propagation in a tank and extend the model to consider the deformation of waves travelling over a submerged bar, respectively. In order to illustrate the moving body capability, the present numerical model is also applied to radiation waves induced by a submerged circular cylinder undergoing forced oscillations. The numerical results are compared with experimental data and analytical solutions.

5.1. Current flow past a circular cylinder

We first calculate the current flow past a circular cylinder at various low Reynolds numbers (Re) which are defined on the basis of the diameter of the cylinder (D), and compare the results with previous experimental and numerical data. The simulations are performed in a rectangular domain $20D$ long and $10D$ wide. The centre of the cylinder is located at a distance $5D$ from the inlet boundary at the left hand end.

Grid convergence tests are carried out first for the case of Re=200, with three different non-uniform meshes. *Mesh 1* employs $92(x) \times 76(y)$ cells with a minimum grid size of $\Delta x = \Delta y = 0.03D$ around the cylinder, and the time interval is taken as $\Delta t = 0.1$. *Mesh 2* and *Mesh 3* have 138×113 cells and 180×146 cells, where the minimum grid sizes are $\Delta x = \Delta y = 0.02D$ and $\Delta x = \Delta y = 0.015D$, respectively. For the last two meshes, a smaller time interval of $\Delta t = 0.05$ is chosen to obtain stable computations. Table I shows the numerical results including the drag (C_D) and lift (C_L) coefficients, and the Strouhal number (St) that indicates the vortex shedding frequency, obtained with these three meshes. It can be seen that the computation converges fast, and the results obtained with *Mesh 2* are very close to those using *Mesh 3*, thus *Mesh 2* is adopted in the following calculations for this case. The vortex shedding frequency is seen to converge the

DOI: 10.1002/fld

Table I. Hydrodynamic parameters of current flow over a circular cylinder at $Re = 200$.

	C_D			C_L	
	Max.	Min.	Mean	Amp.	St
Lecointe and Piquet [30]	1.50	1.42	1.46	0.70	0.230
Franke *et al.* [31]	—	—	1.31	—	0.194
Chan and Anastasiou [32]	1.53	1.43	1.48	0.63	0.180
Chen *et al.* [33]	1.37	1.29	1.33	0.72	0.200
Farrant *et al.* [34]	—	—	1.37	—	0.196
Meneghini *et al.* [35]	—	—	1.30	—	0.196
Wu and Hu [36]	1.39	1.32	1.36	0.56	0.190
Present					
Mesh 1	1.53	1.42	1.47	0.78	0.195
Mesh 2	1.45	1.35	1.40	0.74	0.195
Mesh 3	1.43	1.33	1.39	0.76	0.195

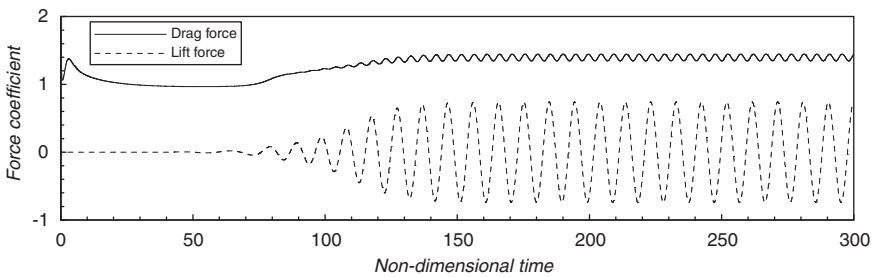

Figure 6. Time history of drag and lift coefficients on the cylinder at $Re = 200$.

fastest, and even the coarse mesh (*Mesh 1*) can predict very accurate *St* number. In addition, the present results fall within the range of other experimental and numerical results, except that the amplitude of the lift force is a little larger here.

The time history of the forces acting on the cylinder at $Re = 200$ is given in Figure 6. The calculations presented here are performed on a desktop PC with 2.8 GHz CPU and 2.0 GB RAM. The CPU time required for this case is around 12 h, which corresponds to 7.2 s for each time step with about 15 000 cells in the computational domain. Figure 7 shows the results for the drag coefficient at lower Reynolds numbers under 100, and comparison with the experimental data measured by Tritton [37]. The numerical results reported in Lei *et al.* [38] are also plotted in the same figure for comparison. It can be seen that the variation of the calculated drag coefficient with the Reynolds number agrees better with the measured data than that from Lei *et al*. [38] whose values are slightly higher than the experimental data.

5.2. Wave sloshing in a rectangular container

Wave sloshing in a container under the influence of gravity is a classical test case for free surface flow problems. In this case, the length of the container is 1 m, and the free water surface has an initial slope of 0.02 with the still water depth of 0.2 m. Once the fluid begins to move under gravity,

DOI: 10.1002/fld

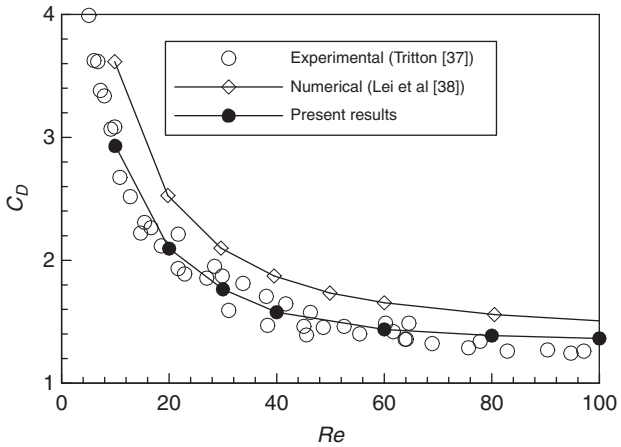

Figure 7. Variation with Reynolds number of drag coefficient on the cylinder.

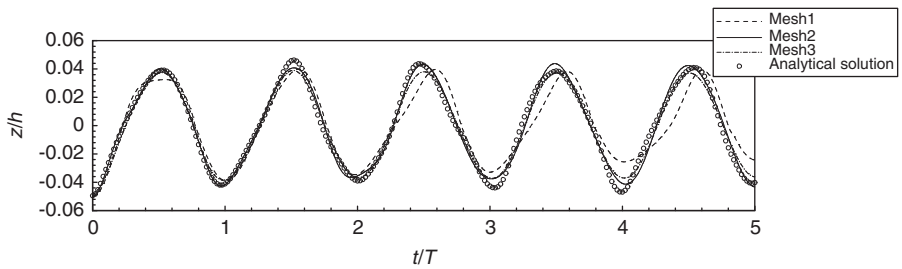

Figure 8. Convergence of wave elevation on the left side wall with different meshes.

there exists an infinite number of standing wave modes in the container and the analytical solution to this problem can be found based on linear wave theory (see Lin and Li [39] for details). In this and the following sections, viscous effects are ignored in all calculations, and the Euler scheme is chosen to approximate the unsteady term in the momentum equations ($\lambda=0$ in Equations (12) and (13)) due to the difficulty in determining the value of ϕ_P^{n-2} in Equation (13) for moving boundary problems. At the same time, the non-uniform meshes used in all calculations employ finer grids around the solid body surface in the horizontal direction and near the free water surface in the vertical direction.

To investigate grid convergence, three different meshes, denoted by *Mesh 1*, *Mesh 2* and *Mesh 3*, using 15×3, 30×6 and 60×12 cells, respectively, are adopted. In Figure 8, results for the position of the interface at the left boundary against time are compared with the analytical solution, from which we can observe that the numerical results with different meshes approach the analytical solution gradually and a nearly convergent solution can be obtained using *Mesh 2* with 180 cells in the computational domain. We also investigate the convergence of the calculation with three different time intervals, as shown in Figure 9. It can be found that the calculations seem to be insensitive to the time interval, and all results are almost identical. Therefore, the time interval

DOI: 10.1002/fld

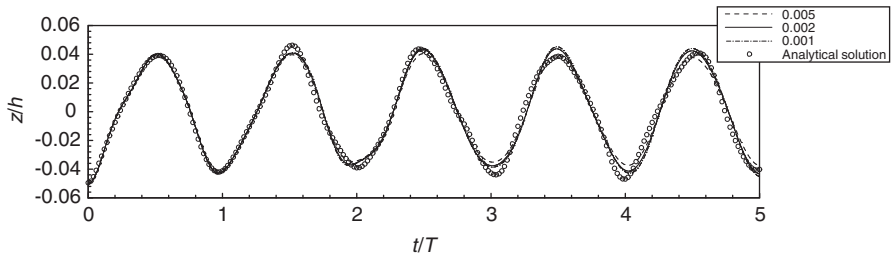

Figure 9. Convergence of wave elevation on the left side wall with different time steps.

used for all calculations in this and the following sections is taken as $0.002T$, where T is the wave period.

Figure 10 shows comparisons of the wave profiles at six time instants with the corresponding analytical solutions. The first four frames in Figure 10 correspond to the free surfaces during one wave period for the leading mode and the last two demonstrate how the wave behaves after a long time simulation. From the comparisons, we can see that the numerical results agree well with the analytical solution during the first wave period, but a little larger discrepancy appears at the last time instant. This might be caused by accumulated errors in the numerical model or the neglect of nonlinear effects in the analytical solutions. The overall comparisons, however, are fairly good, indicating that the model can predict the free surface location accurately. The model is further tested for the conservation of total mass. In Figure 11, the time history of mass normalized by the total fluid mass at the initial time step is shown. It can be seen that the total mass discrepancy is less than 0.15% even when the very coarse mesh is used, which can prove that the present numerical method is conservative.

5.3. Wave generation in a wave tank

To further validate the present method, the experiment conducted by Gao [40] to investigate wave generation and propagation in a tank is reproduced numerically. In this test, regular waves are generated in a 8.75 m long wave tank with a still water depth of 0.28 m, and the velocity of the wave paddle is prescribed to be the same as that used in the experiment. Three different meshes are tested, whose numbers of cells are $200\times7\,400\times13$ and 600×20, respectively.

During our initial numerical calculations, however, a saw-tooth instability arises leading to the computation diverging, as seen in Figure 12 for the wave profiles along the tank at two time instants obtained with *Mesh 1*. It can be seen that at $t=4T$ the numerical instability is hardly evident, whereas at $t=8T$ it appears clearly near the wave paddle. A smoothing technique is therefore applied. To check the effect of smoothing on the numerical results, two smoothing frequencies have been used: in one wave period the smoothing technique is performed once and 2.5 times, respectively. From the results shown in Figure 12, we find that both of these smoothing frequencies will work efficiently. The smoothing technique can remove the numerical instability where the saw-tooth instability appears, and it does not change the results much where the free surface is very smooth. In addition, different smoothing frequencies seem to provide very similar results, but frequent smoothing will of course introduce some numerical damping into the numerical results. The principle of choosing the smoothing frequency in the following calculations is to use as little as possible.

DOI: 10.1002/fld

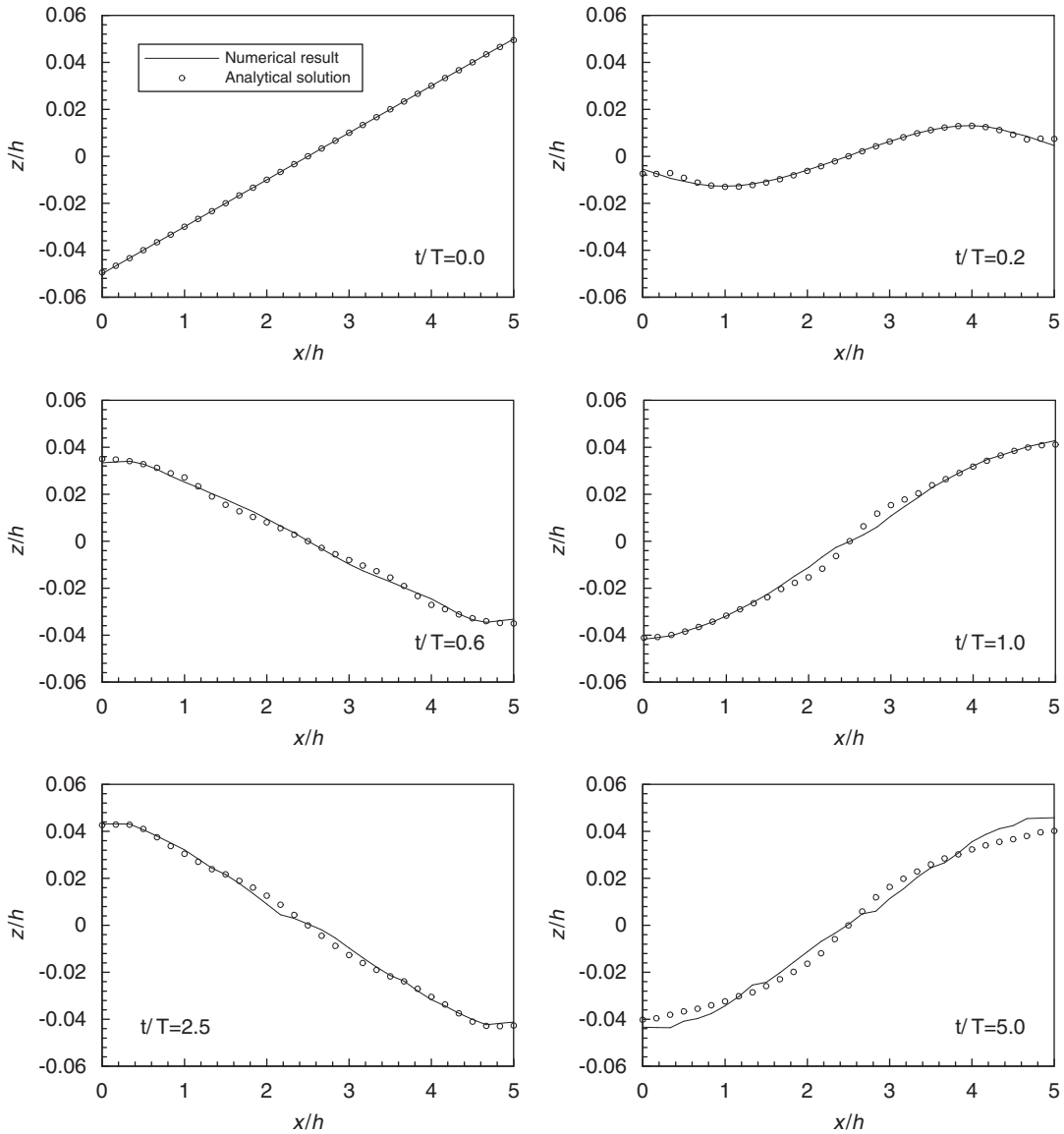

Figure 10. Comparison of wave profile with analytical solution.

In Figure 13, free surface elevations at three different stations ($x = 0.55, 3.75$ and 5.45 m from the position of the wave paddle) are compared with the corresponding experimental results. We can see that the results obtained with the coarse mesh are in good agreement with those using the other two finer meshes at stations close to the wave paddle ($x = 0.55$ m). However, far away from the wave paddle ($x = 3.75$m and 5.45 m) large differences appear in the results, where numerical damping caused by the accumulation of numerical errors can be observed for the coarse

DOI: 10.1002/fld

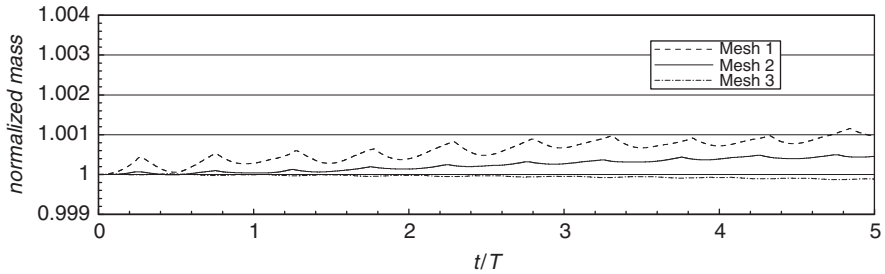

Figure 11. Time history of normalized mass calculated with different meshes.

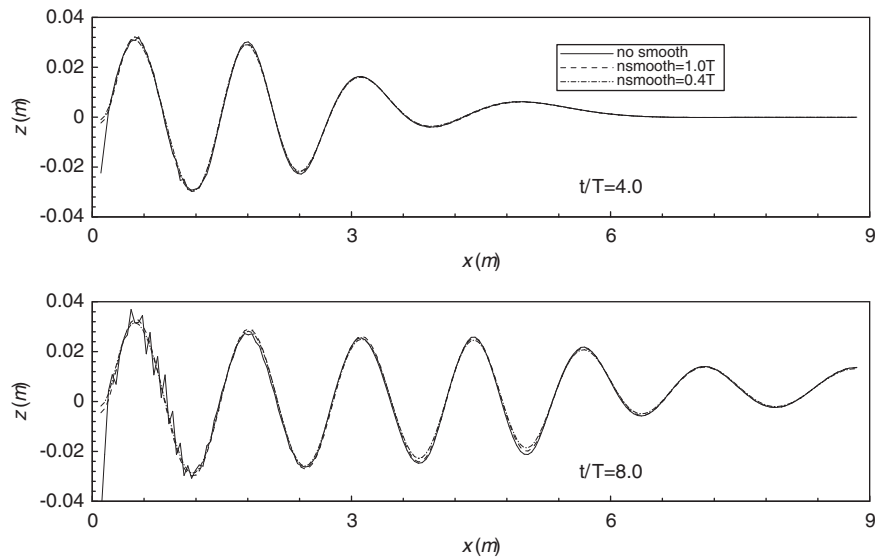

Figure 12. Effect of the smoothing on the wave profile.

mesh. The values predicted on the two finer meshes lie close to each other at all three stations; this indicates that a grid-independent solution is achieved by the present numerical method. It should be noted that a large discrepancy between the present numerical results and the experimental data is found at the station near the wave paddle, and this discrepancy becomes smaller at increasing the distance from the wave paddle. The reason for this phenomenon is that compared with the physical experiment, a different inlet boundary condition is used in the numerical simulation that uses the velocity at the inlet boundary with fixed wave paddle. However, in the experiment, the wave paddle is actually moving, which consequently affects the comparison of the results near the wave paddle.

In order to support our explanation for the discrepancy between the numerical and experimental results, the present numerical model is also compared with the analytical solution. Except that the velocity at the wave paddle is calculated based on the second-order Stokes wave theory, the geometry, the computational meshes and the generated wave condition are all the same as those

DOI: 10.1002/fld

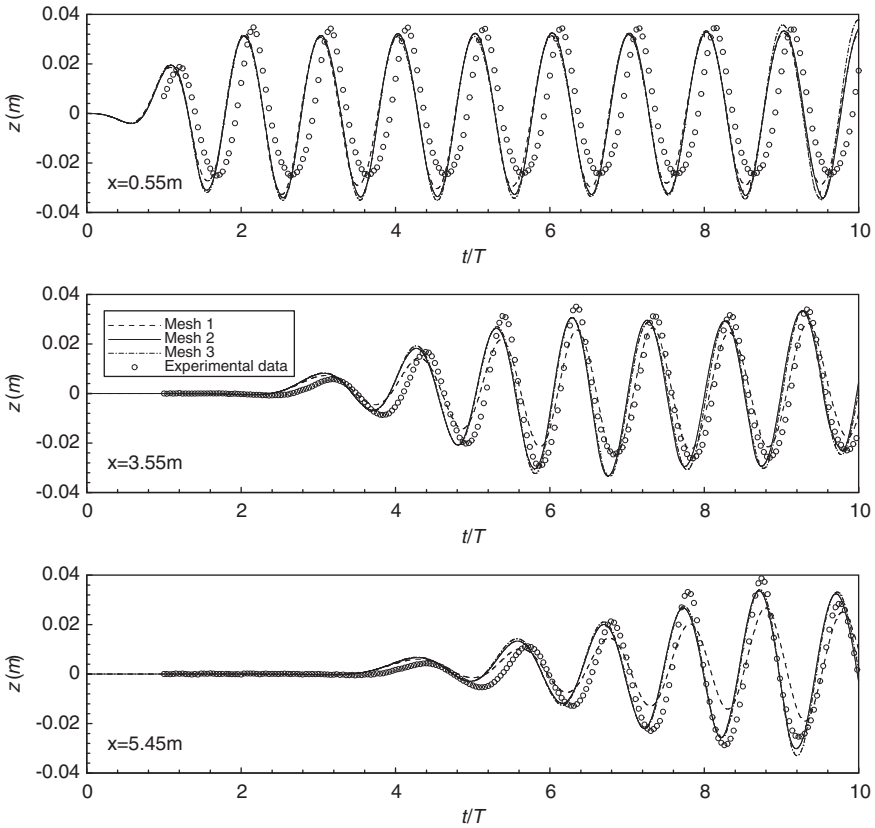

Figure 13. Comparison of wave elevation with experimental data.

in the physical experiment. Figure 14 compares the numerical results with the analytical solutions to the second-order Stokes wave at the same three stations. Now, we can find that they are in the satisfactory agreement, which indicates again that the present numerical model is accurate.

5.4. Wave travelling over a submerged bar

The present numerical model is also used for simulating the propagation of regular waves over a submerged bar on a horizontal bottom, which has been investigated numerically by numerous authors (Lin and Li [39], Huang and Dong [41], Shen and Chan [42]). In this case, the incoming wave on the upward slope is shoaling, with nonlinearity generating bounded higher harmonic waves, which travel phase-locked to the primary wave. On the downward slope these harmonic waves are released as free waves, resulting in an irregular wave pattern. In the computational domain, sketched in Figure 15, the velocity at the inlet boundary is prescribed in terms of linear wave theory, the incident wave height is 0.02 m and the wave period is 2 s. The set-up is the same as that used in the experiment reported by Beji and Battjes [43]. There are 16 cells in the vertical direction, and the numbers of cells in the horizontal direction are 500, 750 and 1000,

DOI: 10.1002/fld

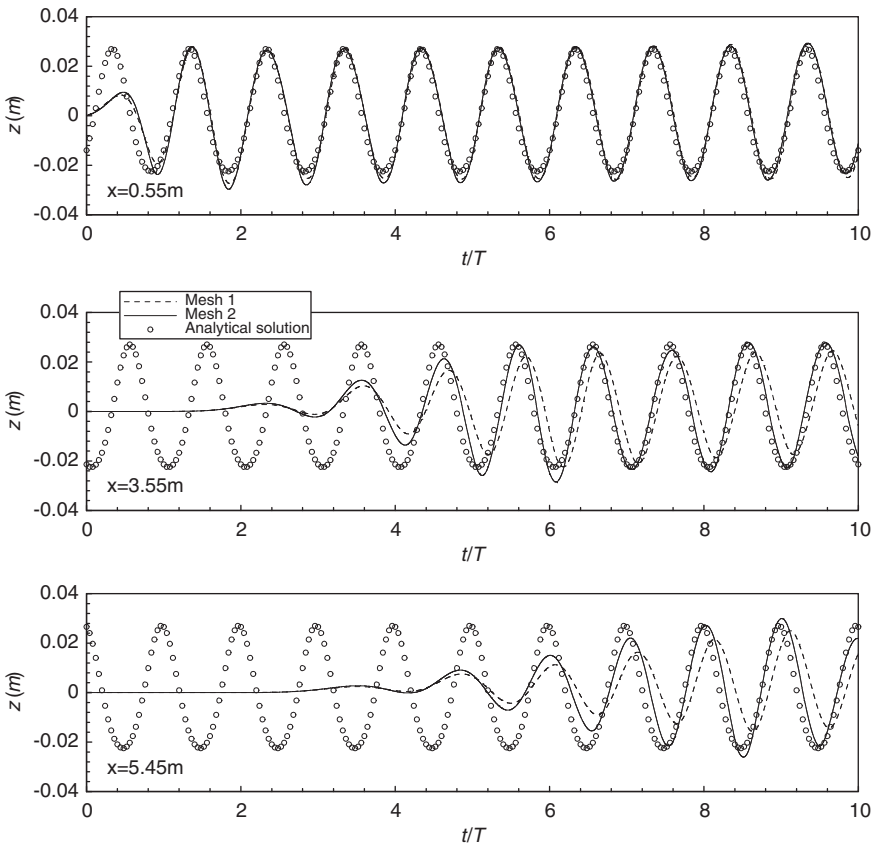

Figure 14. Comparison of wave elevation with analytical solution for the second-order Stokes wave.

corresponding to *Mesh 1*, *Mesh 2* and *Mesh 3*, respectively. For the cases in this and the next section, the Euler scheme is adopted for updating the free surface boundary condition, which can save computer effort compared with the Runge–Kutta scheme used before.

Figure 16 shows a comparison of free surface elevation with experimental data at six different stations. At the first two stations, the waves remain almost sinusoidal with good agreement between the numerical results and the experimental data, even for the coarsest mesh. At the second two stations, where the wave rides over the top of breakwater, we find that some details of wave signature do not completely agree with the experimental data. Behind the breakwater, the prediction of wave transformation in this region is most difficult because of the complicated flow separation and nonlinear wave energy transfer. Therefore, the discrepancies between the numerical results and the experimental data become larger at the last two stations but the numerical results tend to agree better with the experimental data when finer meshes are adopted. These discrepancies are probably due to the neglect of viscosity and the accumulation of numerical errors caused by the introduction of merged cells.

The numerical results for free surface profiles along the wave tank at an interval of 1s are shown in Figure 17. It can be seen that as waves propagate over the bar, the primary wave crests become

DOI: 10.1002/fld

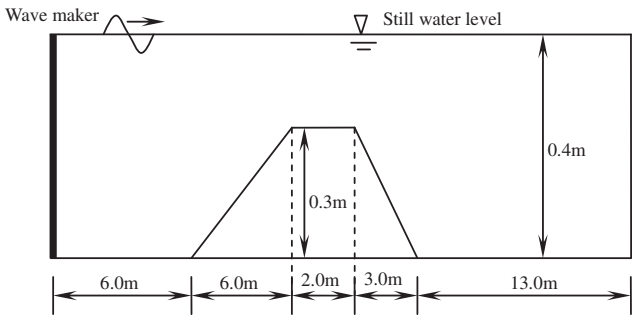

Figure 15. Experimental layout for wave travelling over a submerged bar.

steeper and a dispersive tail gradually develops. A small wave appears at the trailing edge of a primary wave. Because this small wave propagates more slowly than the main crest, it gradually detaches from the main crest and is overtaken by the next wave. Very careful examination of Figure 17 reveals some apparent very slight saw-tooth instabilities in the wave profiles, but these do not affect the final results of the calculations.

5.5. Wave generation by a submerged circular cylinder undergoing forced motions

Lastly, we investigate wave radiation by a submerged circular cylinder undergoing forced sinusoidal oscillations. This is a more difficult case, because the computational domain involves both a complicated free water surface as a domain boundary and a moving body. It may be noted that the mesh generation for this case cannot be easily realized by mesh-moving methods, whereas the present numerical model using the Cartesian cut cell approach can show its strong ability for dealing with complicated moving boundary problems. The size of the rectangular tank is shown in Figure 18 for the forced oscillation period T of 2 s, and 420×25 cells are distributed in the computational domain. The balanced position of the submerged circular cylinder is situated on the centre of the tank, and the radius of the circular cylinder is 0.1 m.

The vertical oscillating circular cylinder is considered first. Figure 19 shows the wave profiles excited by the submerged circular cylinder under different oscillating amplitudes a at $T = 2$ s. From this figure, we can clearly see the process of radiated wave generation and propagation in the tank. Because the specified excitation period is very small which corresponds to a low-frequency oscillation, the wave crest at $t = 1.0T$ has already reduced due to gravity, before the circular cylinder excites the next wave. Therefore, waves at higher frequency become apparent at this time instant, and these higher-frequency waves superpose on the primary waves, leading to an irregular wave pattern at $t = 3.0T$. To summarize, the numerical results are very stable during the whole simulated period, and they are symmetric about the expected symmetric plane. In addition, the influence of higher-frequency waves becomes more marked and the resulting wave is therefore more irregular, with the increase of the incident wave amplitude. This phenomenon can also be seen in Figure 20, in which the time history of wave elevation at two different stations is given. At $x = 7.0$ m, the higher-frequency waves are more visible for $a = 0.04$ m. However, at $x = 8.0$ m where the station is situated at the top of the balanced position of the submerged circular cylinder, the primary wave plays a dominant role.

DOI: 10.1002/fld

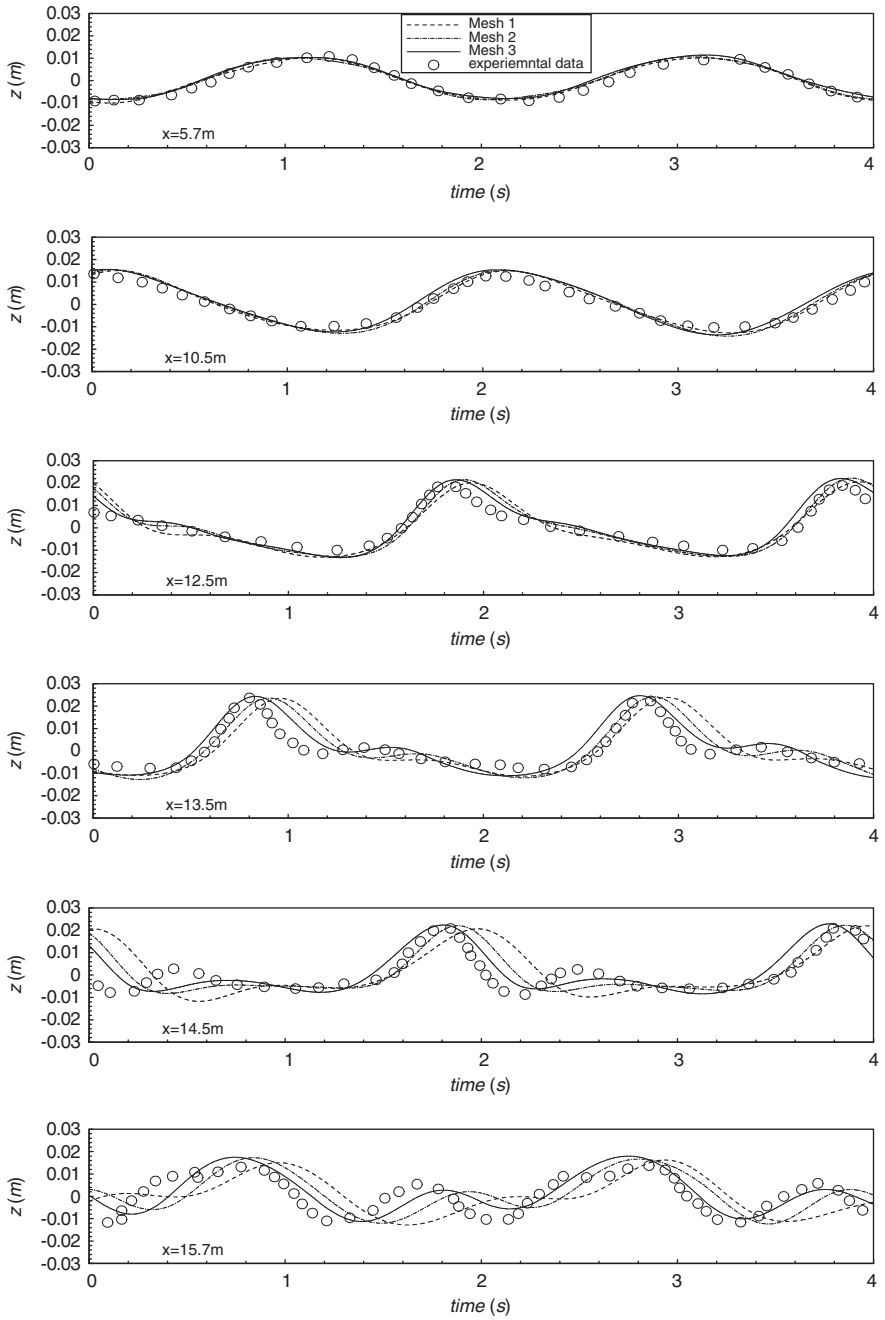

Figure 16. Time history of wave elevations at six different stations and the comparison with experimental data.

DOI: 10.1002/fld

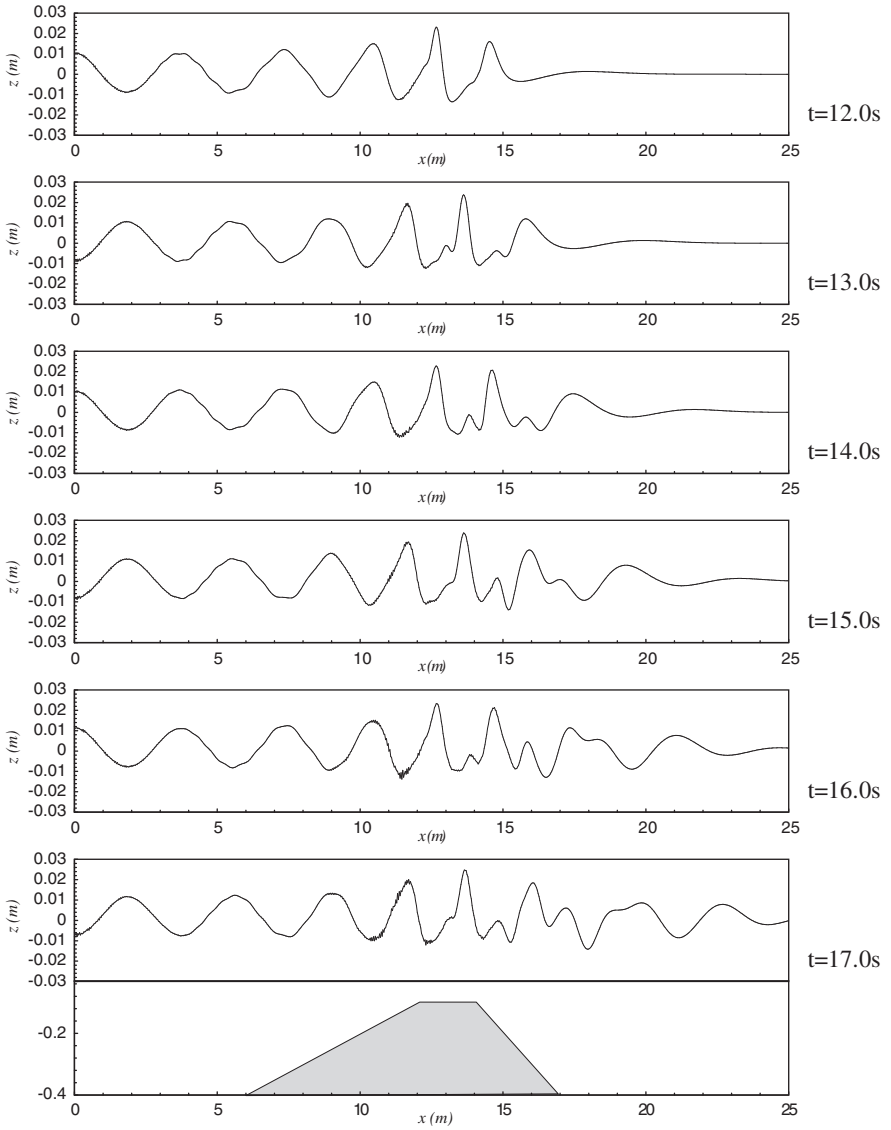

Figure 17. Wave profiles along the wave tank at several time instants.

Having considered vertical motion, we now simulate forced horizontal motion. Figure 21 shows the wave profiles for two motion amplitudes at $T = 2$ s. We can see from the figure that the characteristics of the wave system in this case are similar to those produced by the vertically moving circular cylinder: the superposition of higher-frequency waves and primary waves causes the resulting irregular wave. However, the distribution of the higher-frequency waves is smaller here, and the wave travels with the typical excitation wave length approximately. The time history of wave elevation at three different stations is also compared between these two motion amplitudes,

DOI: 10.1002/fld

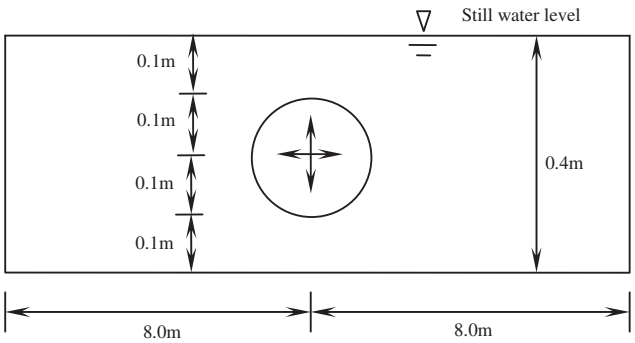

Figure 18. Sketch of the computational domain for the submerged circular cylinder undergoing force oscillations with the period $T = 2$ s.

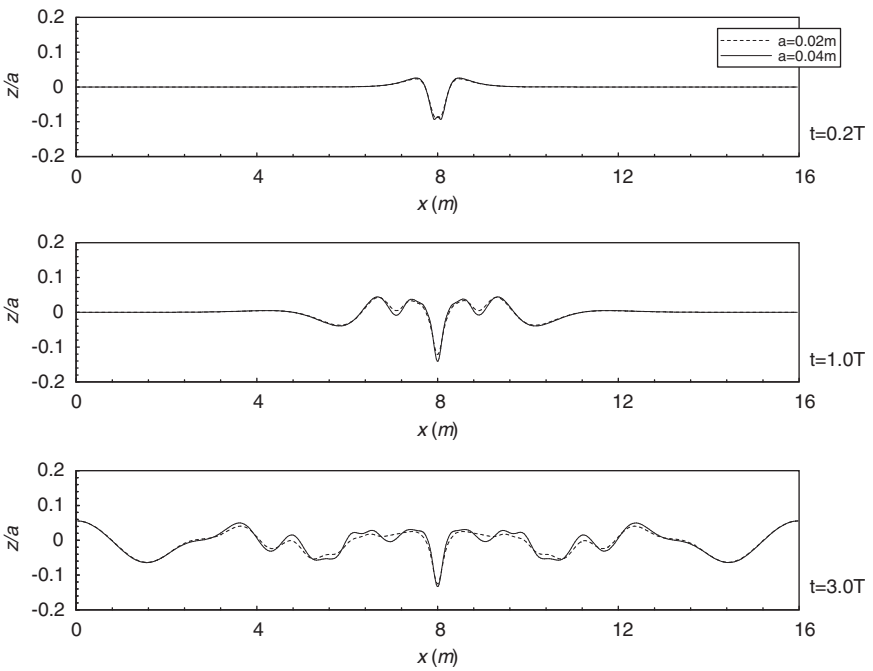

Figure 19. Wave profile for the submerged circular cylinder undergoing forced vertical oscillation at $T = 2$ s with two different amplitudes.

as shown in Figure 22. The phase shift between the results at $x = 7.0$ m and $x = 9.0$ m is clear from this figure. At $x = 8.0$ m, the free surface is nearly stationary for $a = 0.02$ m, whereas it oscillates at double the frequency of the body motion for $a = 0.05$ m due to the stronger nonlinearity.

In order to investigate the influence of excitation period on the results, we calculate the wave profiles induced by the circular cylinder having motion amplitude $a = 0.02$ m at three different periods: the results can be observed in Figure 23. It should be noted that the length of the tank is

DOI: 10.1002/fld

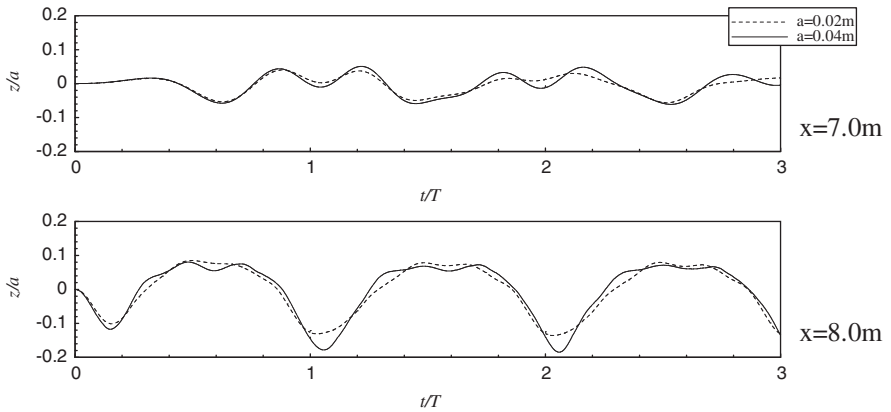

Figure 20. Time history of wave elevation for the submerged circular cylinder undergoing forced vertical oscillation at $T=2\,\mathrm{s}$ with two different amplitudes.

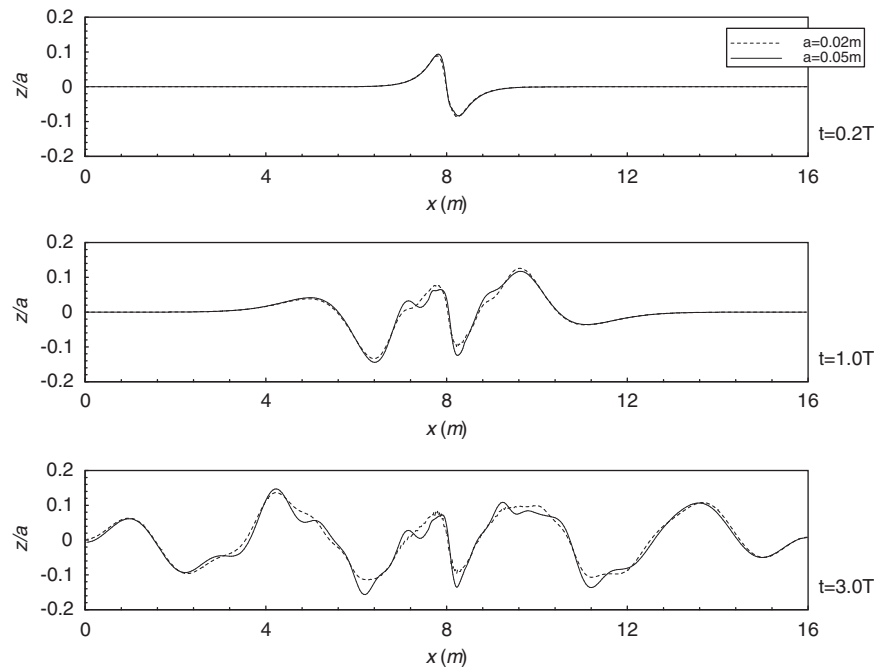

Figure 21. Wave profile for the submerged circular cylinder undergoing forced horizontal oscillation at $T=2\,\mathrm{s}$ with two different amplitudes.

chosen as 4λ approximately for different excitation periods, where λ is the wave length in terms of linear wave theory. We can see that the disturbance of the water surface is very small at such a low excitation frequency ($t=4.0\,\mathrm{s}$). However, the body oscillating at higher frequency ($t=1.0\,\mathrm{s}$)

DOI: 10.1002/fld

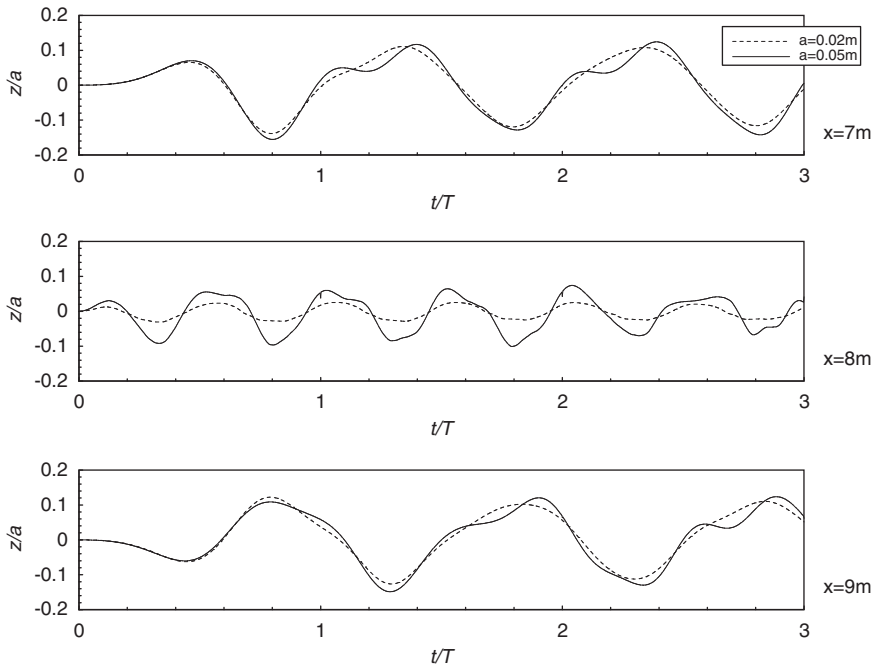

Figure 22. Time history of wave elevation for the submerged circular cylinder undergoing forced horizontal oscillation at $T=2$ s with two different amplitudes.

can induce a very large primary wave having the same frequency as the body oscillation. In this case, the higher-frequency waves mentioned before are suppressed.

6. CONCLUSIONS

The Navier–Stokes and Euler equations have been solved for simulating flows with and without free surfaces by using a finite volume method with the application of the Cartesian cut cell approach. In the cut cell approach, generally speaking, the choice of the background grid is arbitrary: any existing grid systems can be applied in generating a cut cell mesh, but the Cartesian grid adopted here is the simplest one among others. Therefore, in principle, any existing flow solvers developed for any grid systems can be extended conveniently to adopt the cut cell approach. Not much additional work is required, only the introduction of corresponding boundary conditions in the cut cells is necessary, which means that existing programs can be utilized to a maximum degree. The cut cell approach is particularly suitable for extending existing flow solvers to investigate moving boundary problems, such as free surface flows and fluid–body interactions. In this paper, a number of numerical results obtained using the present cut cell model have been shown. Comprehensive comparisons with experimental data and analytical solutions indicate that this numerical model is very powerful and efficient, especially for complicated moving boundary problems. However, it should be mentioned that due to the use of merged cells in the numerical model,

DOI: 10.1002/fld

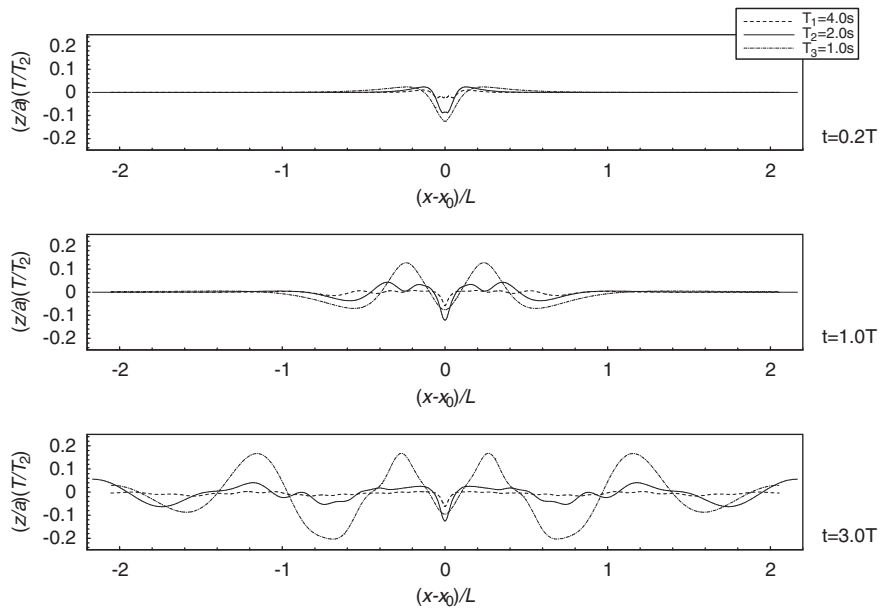

Figure 23. Wave profile for the submerged circular cylinder undergoing forced vertical oscillation with $a = 0.02$ m and three different oscillation periods.

sometimes relatively larger numerical errors become apparent after a long time accumulation: these possibly can be avoided by using a finer mesh in regions where a detailed velocity field needs to be described.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support provided by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/D034566/1.

REFERENCES

1. Bai W, Eatock Taylor R. Higher-order boundary element simulation of fully nonlinear wave radiation by oscillating vertical cylinders. *Applied Ocean Research* 2006; **28**:247–265.
2. Bai W, Eatock Taylor R. Numerical simulation of fully nonlinear regular and focused wave diffraction around a vertical cylinder using domain decomposition. *Applied Ocean Research* 2007; **29**:55–71.
3. Turnbull MS, Borthwick AGL, Eatock Taylor R. Numerical wave tank based on a σ–transformed finite element inviscid flow solver. *International Journal for Numerical Methods in Fluids* 2003; **42**:641–663.
4. Wu GX, Hu ZZ. Simulation of nonlinear interactions between waves and floating bodies through a finite-element-based numerical tank. *Proceedings of the Royal Society*, *Series A* 2004; **460**:2797–2817.
5. Grilli ST, Guyenne P, Dias F. A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom. *International Journal for Numerical Methods in Fluids* 2001; **35**:829–867.
6. Xue M, Xü H, Liu Y, Yue DKP. Computations of fully nonlinear three-dimensional wave–wave and wave–body interactions. Part 1. Dynamics of steep three-dimensional waves. *Journal of Fluid Mechanics* 2001; **438**:11–39.
7. Li TQ, Troch P, De Rouck J. Wave overtopping over a sea dike. *Journal of Computational Physics* 2004; **198**:686–726.

DOI: 10.1002/fld

8. Zhao Q, ARmfield S, Tanimoto K. Numerical simulation of breaking waves by a multi-scale turbulence model. *Coastal Engineering* 2004; **51**:53–80.
9. Lubin P, Vincent S, Abadie S, Caltagirone JP. Three-dimensional large eddy simulation of air entrainment under plunging breaking waves. *Coastal Engineering* 2006; **53**:631–655.
10. Christensen ED. Large eddy simulation of breaking waves. *Coastal Engineering* 2001; **42**:53–86.
11. Park JC, Kim MH, Miyata H, Chun HH. Fully nonlinear numerical wave tank (NWT) simulations and wave run-up prediction around 3-D structures. *Ocean Engineering* 2003; **30**(15):1969–1996.
12. Losasso F, Fedkiw R, Osher S. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids* 2005; **35**:995–1010.
13. Qian L, Causon DM, Mingham CG, Ingram DM. A free-surface capturing method for two fluid flows with moving bodies. *Proceedings of the Royal Society*, *Series A* 2006; **462**:21–42.
14. Gao F, Ingram DM, Causon DM, Mingham CG. The development of a Cartesian cut cell method for incompressible viscous flows. *International Journal for Numerical Methods in Fluids* 2007; **54**:1033–1053.
15. Sung J, Choi HG, Yoo JY. Time-accurate computation of unsteady free surface flows using an ALE-segregated equal-order FEM. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:1425–1440.
16. Apsley D, Hu W. CFD simulation of two- and three-dimensional free-surface flow. *International Journal for Numerical Methods in Fluids* 2003; **42**:465–491.
17. Zhu G, Borthwick AGL, Eatock Taylor R. A finite element model of interaction between viscous free surface waves and submerged cylinders. *Ocean Engineering* 2001; **28**:989–1008.
18. Lo DC, Young DL. Arbitrary Lagrangian–Eulerian finite element analysis of free surface flow using a velocity–vorticity formulation. *Journal of Computational Physics* 2004; **195**:175–201.
19. Souli M, Zolesio JP. Arbitrary Lagrangian–Eulerian and free surface methods in fluid mechanics. *Computer Methods in Applied Mechanics and Engineering* 2001; **191**:451–466.
20. Rabier S, Medale M. Computation of free surface flows with a projection FEM in a moving mesh framework. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:4703–4721.
21. Causon DM, Ingram DM, Mingham CG, Yang G, Pearson RV. Calculation of shallow water flows using a Cartesian cut cell approach. *Advanced in Water Resources* 2000; **23**:545–562.
22. Causon DM, Ingram DM, Mingham CG. A Cartesian cut cell method for shallow water flows with moving boundaries. *Advanced in Water Resources* 2001; **24**:899–911.
23. Liang Q, Zang J, Borthwick AGL, Taylor PH. Shallow flow simulation on dynamically adaptive cut cell quadtree grids. *International Journal for Numerical Methods in Fluids* 2007; **53**:1777–1799.
24. Ning DZ, Zang J, Liang Q, Taylor PH, Borthwick AGL. Boussinesq cut-cell model for non-linear wave interaction with coastal structures. *International Journal for Numerical Methods in Fluids* 2008; **57**:1459–1483.
25. Yang G, Causon DM, Ingram DM. Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut cell method. *International Journal for Numerical Methods in Fluids* 2000; **33**:1121–1151.
26. Cieslak S, Khelil SB, Choqiet I, Merlen A. Cut cell strategy for 3D blast wave numerical simulation. *Shock Waves* 2001; **10**(6):421–429.
27. Tucker PG, Pan Z. A Cartesian cut cell method for incompressible viscous flow. *Applied Mathematical Modelling* 2000; **24**:591–606.
28. Chung MH. Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape. *Computers and Fluids* 2006; **35**:607–623.
29. Ferziger JH, Perić M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, Heidelberg, 1999.
30. Lecointe Y, Piquet J. On the use of several compact methods for the study of unsteady incompressible viscous flow around a circular cylinder. *Computer and Fluids* 1984; **12**:255–280.
31. Franke R, Rodi W, SchÖnung B. Numerical calculation of laminar vortex shedding flow past cylinders. *Journal of Wind Engineering and Industrial Aerodynamics* 1990; **35**:237–257.
32. Chan CT, Anastasiou K. Solution of incompressible flows with or without a free surface using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1999; **29**:35–57.
33. Chen YH, Yang SC, Yang JY. Implicit weighted essentially non-oscillatory schemes for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1999; **31**:747–765.
34. Farrant T, Tan M, Price WG. A cell boundary element method applied to laminar vortex-shedding from arrays of cylinders in various arrangements. *Journal of Fluids and Structures* 2000; **14**:375–402.
35. Meneghini JR, Saltara F, Siqueira CLR. Numerical simulation of flow interference between two circular cylinders in tandom and side-by-side arrangements. *Journal of Fluids and Structures* 2001; **15**:327–350.

DOI: 10.1002/fld

36. Wu GX, Hu ZZ. Numerical simulation of viscous flow around unrestrained cylinders. *Journal of Fluids and Structures* 2006; **22**:371–390.
37. Tritton DJ. Experiments on the flow past a circular cylinder at low Reynolds number. *Journal of Fluid Mechanics* 1959; **6**:547–567.
38. Lei C, Cheng L, Kavanagh K. A finite difference solution of the shear flow over a circular cylinder. *Ocean Engineering* 2000; **27**:271–290.
39. Lin PZ, Li CW. A σ-coordinate three-dimensional numerical model for surface wave propagation. *International Journal for Numerical Methods in Fluids* 2002; **38**:1045–1068.
40. Gao F. An efficient finite element technique for free surface flow. *Ph.D. Thesis*, Brighton University, U.K., 2003.
41. Huang CJ, Dong CM. Wave deformation and vortex generation in water waves propagating over a submerged dike. *Coastal Engineering* 1999; **37**:123–148.
42. Shen L, Chan ES. Numerical simulation of fluid–structure interaction using a combined volume of fluid and immersed boundary method. *Ocean Engineering* 2008; **35**:939–952.
43. Beji S, Battjes JA. Numerical simulation of nonlinear wave propagation over a bar. *Coastal Engineering* 1994; **23**:1–16.

DOI: 10.1002/fld