Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 212 (2006) 52–72
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Quasi ALE finite element method for nonlinear water waves

Q.W. Ma *, S. Yan

School of Engineering and Mathematical Sciences, City University, Northampton Square, London EC1V 0HB, UK

Received 4 April 2005; received in revised form 31 May 2005; accepted 21 June 2005
Available online 3 August 2005

Abstract

This paper presents a newly developed quasi arbitrary Lagrangian–Eulerian finite element method (QALE-FEM) for simulating water waves based on fully nonlinear potential theory. The main difference of this method from the conventional finite element method developed by one of authors of this paper and others (see e.g. [Q.W. Ma, G.X. Wu, R. Eatock Taylor, Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 1: Methodology and numerical procedure and Part 2: Numerical results and validation, Int. J. Numer. Methods Fluids, 36 (2001) 265–308.] and [G.X. Wu, Z.Z. Hu, Simulation of nonlinear interactions between waves and floating bodies through a finite-element-based numerical tank, Proc. R. Soc. A 460 (2004) 2050, 3037–3058.]) is that the complex mesh is generated only once at the beginning and is moved at all other time steps in order to conform to the motion of the free surface and structures. This feature allows one to use an unstructured mesh with any degree of complexity without the need of regenerating it every time step, which is generally inevitable and very costly. Due to this feature, the QALE-FEM has high potential in enhancing computational efficiency when applied to problems associated with the complex interaction between large steep waves and structures since the use of an unstructured mesh in such a case is likely to be necessary. To achieve overall high efficiency, the numerical techniques involved in the QALE-FEM are developed, including the method to move interior nodes, technique to re-distribute the nodes on the free surface, scheme to calculate velocities and so on. The model is validated by water waves generated by a wavemaker in a tank and the interaction between water waves and periodic bars on the bed of tank. Satisfactory agreement is achieved with analytical solutions, experimental data and numerical results from other methods.

Keywords: QALE-FEM; Nonlinear water waves; Spring analogy; Periodic bars; Free surface

1. Introduction

With operations in the oil and gas industry moving to deeper waters, offshore structures are more likely to be exposed to very harsh environments and extremely steep waves and therefore undergo large motions.

* Corresponding author. Tel.: +44 20 7040 8159; fax: +44 20 7040 8566.
E-mail address: q.ma@city.ac.uk (Q.W. Ma).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.06.014

As a result, there is increasing interest in numerically simulating nonlinear water waves and their interaction with structures. There are two classes of theoretical models for cases with finite water depth in common use for numerical simulations. One is based on the general flow theory and the other is based on the potential theory. In the first class of models, the Navier–Stokes and continuity equations together with proper boundary conditions are solved, while in the second class, the Laplace equation with fully nonlinear boundary conditions is dealt with. For brevity, the first class of models will be called NS Model and the second called FNPT (representing fully nonlinear potential theory) Model in the paper.

In the community of researchers who use the NS Model, three formulations have been suggested: Eulerian, Lagrangian and arbitrary Lagrangian–Eulerian (ALE) formulations. In the Eulerian formulation, the computational mesh is fixed and the fluid moves relative to it (see for example [1–4]). Use of this formulation can handle, with relative ease, the large distortions of interfaces between two different materials by employing proper interface capturing techniques (such as volume of fluids, level set method and so forth) but possibly at the expense of solving a larger domain than necessary, solving an extra governing equation and smearing the interfaces and flow details near the interfaces. In addition, it seems to be difficult to handle cases with interfaces of three different materials, such as those with floating bodies in waves. In the Lagrangian formulation (see for instance [5]), all nodes follow their corresponding fluid particles. Due to this feature, the formulation allows sharp tracking of interfaces between different materials. However, if large distortions of the fluid domain occur, certain nodes may become too close to or too far from others and consequently lead to a breakdown of the computing process if remeshing is not performed. The arbitrary ALE formulation is a hybrid approach, in which the computational mesh does not need to adhere to fluid particles or to be fixed in space but can be moved arbitrarily. Therefore, the ALE formulation can make use of the merits of both the Lagrangian and Eulerian formulations and alleviate many of their drawbacks. Specifically, the interface can be precisely tracked without necessarily remeshing. Of course, the nodes have to be moved in order to conform to the deformation or distortion of interfaces or boundaries and the governing equations are made a bit more complex to account for the moving velocities of mesh. The ALE formulation has been discussed and used in many publications. Only a few are listed here as examples [6–8]. Various numerical methods, such as finite element, finite volume and finite different methods have been used to solve the Navier–Stokes and continuity equations together with one of three formulations to investigate the nonlinear water waves and their interaction with fixed structures. However, no matter which method is used, solving NS equations is always a time consuming task.

Due to this fact, the FNPT model has been adopted in many publications for problems associated with the nonlinear water waves and their interactions with structures. In this model, viscosity is ignored. The governing equations are dramatically simplified and so they need much less computational resource to be solved than in the NS Model. Comparison with experimental data [9–12] has shown that the results obtained by using this model are accurate enough if breaking waves do not occur and/or if structures involved are large, implying that to neglect the viscous effects is acceptable in these cases. Therefore, the FNPT Model instead of the NS Model should be employed if a case considered falls in this category, i.e., without wave breaking and/or with large structures. The problems formulated by this model are usually solved by a time marching procedure suggested by Longuet-Higgins and Cokelet [13]. At each time step, a boundary value problem based on the Laplace equation together with boundary conditions is solved. Then the variables on the free surface and on the surface of structures are updated using the kinematic/dynamic free surface conditions and dynamic equations of structures, respectively, which give the required boundary values for the solution at the next time step. The procedure can be repeated in principle for any desired period of time. In this procedure, the key task is to solve the boundary value problem by using an efficient numerical method. To do so, boundary element methods (BEMs) have been used in many publications, such as [10,14–17], and have produced many impressive and useful results. Finite element methods (FEMs) have also been developed and used for two and three dimensional problems, see for instance [11,12,18–24]. Both methods have been proved quite efficient but the FEMs need less memory and so are computationally more efficient, as

indicated by Wu and Eatock Taylor [19] and Ma et al. [11]. The reason is that although there are far fewer unknowns when using the BEMs than using the FEMs, nonzero elements in the matrix for the BEMs may be more than those in the matrix for the FEMs since any node in the BEMs is affected by all others, while in the FEMs only by those connected with the node. A drawback of the FEMs, however, is that an unstructured mesh is generally required for complex interaction between water waves and structures and may need to be remeshed at every time step to follow the motion of waves and/or structures. Repeatedly regenerating such a mesh can make the required CPU time prohibitive in a simulation of several thousands steps on a normal workstation. In order to reduce the time spent on the remeshing, simple structured mesh has been used in [11,12]. For the same purpose, Wu et al. [22] have recently employed a hybrid mesh. In their approach, a 2D mesh in a horizontal plane (say, the free surface at $t = 0$) is first generated and then vertical lines are drawn to construct a 3D mesh. The 2D mesh is formed by combining an unstructured mesh in a region near structures with a simple structured mesh (similar to [23]) in other regions. This is a sensible approach but restricted to cylindrical structures without roll and pitch motions.

In this paper, the FNPT model and the FEM are still employed. However, a new way is pioneered to alleviate the bottleneck caused by the mesh regeneration. The main idea is that the complex mesh is generated only once at the beginning and is moved at other time steps in order to conform to the motions of the free and structure surfaces. In this approach, the mesh can be generated by any generator and can have any complexity, any structure and any desired distribution. Because the mesh generator is used only once in a simulation of several thousands time steps, the CPU time spent on mesh generation is not an important matter since it may be only a small proportion of total computational time even it is quite long (say several minutes). In addition, the generator is not necessarily included in the main code. The idea of moving mesh is borrowed from the ALE formulation for the NS Model. However, the velocities of the moving mesh do not need to be considered in governing equations in our approach. That is why this approach is called as quasi arbitrary Lagrangian–Eulerian finite element method (QALE-FEM). It is obvious that the technique for moving mesh in this approach is vital in order to achieve a good quality mesh at all time steps and to avoid a large CPU requirement. A robust method will be developed herein for this purpose. As the velocity potential, instead of the fluid velocity, is solved as unknowns, care must be taken in computing the velocity based on the velocity potential due to the arbitrary and moving nature of the mesh. An efficient and accurate technique will also be suggested for the velocity computation in this paper.

Although the QALE-FEM based on the FNPT Model can be used to deal with any wave/structure interaction problem without wave overturning/breaking, this paper focuses on the description of the method and validates for cases without floating bodies. Applications to more general cases will be given in other papers.

2. Mathematical model and numerical method

Without loss of generality, the computational domain is chosen as a rectangular tank, and the nonlinear wave is generated in the tank by a piston-like wavemaker. The wavemaker is mounted at the left end (on the negative side of the x-axis) and a damping zone with a Sommerfeld condition (see [11,12] for details) is applied at the right end of the tank in order to suppress the reflection, as sketched in Fig. 1. Arbitrary forms of submerged bodies on the tank-bed may be included. A Cartesian coordinate system is used with the oxy plane on the mean free surface and with the z-axis being positive upwards. Unless mentioned otherwise, its origin will be located at the centre of the tank.

Similar to the usual formulation for the FNPT Model, the velocity potential (ϕ) satisfies Laplace's equation

$$\nabla^2\phi = 0 \tag{1}$$

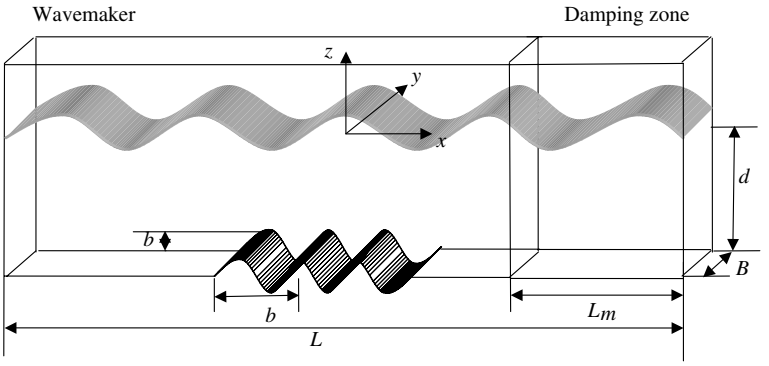

Fig. 1. Sketch of fluid domain.

in fluid domain. On the free surface $z = \varsigma(x, y, t)$, the velocity potential satisfies the kinematic and dynamic conditions:

$$\frac{\mathrm{D}x}{\mathrm{d}t} = \frac{\partial\phi}{\partial x}, \quad \frac{\mathrm{D}y}{\mathrm{d}t} = \frac{\partial\phi}{\partial y}, \quad \frac{\mathrm{D}z}{\mathrm{d}t} = \frac{\partial\phi}{\partial z}, \tag{2}$$

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = -gz + \frac{1}{2}|\nabla\phi|^2, \tag{3}$$

in which g is the gravitational acceleration. The atmospheric pressure has been taken as zero in Eq. (3). On all rigid boundaries, the velocity potential must satisfy

$$\frac{\partial\phi}{\partial n} = \vec{n} \cdot \vec{U}(t), \tag{4}$$

where $\vec{U}(t)$ and $\vec{n}$ is the velocity and the unit normal vector of the rigid boundaries, respectively. For the specific case illustrated in Fig. 1, $\vec{U}(t) = U_x(t)$ on the wavemaker and $\vec{U}(t) = 0$ on other rigid boundaries.

The problem described by Eqs. (1)–(4) will be solved by using a time step marching procedure as outlined in Section 1. At each time step, the shape of the free surface and the potential values on it as well as velocities on all rigid boundaries are known (given as initial conditions or calculated from the solution at previous time steps). Thus, at an instant, the boundary condition for the potential on the free surface can be replaced by a Dirichlet condition

$$\phi = f_{\mathrm{p}}, \tag{5}$$

where f_{p} is the potential values on the free surface at the instant. Therefore, the unknown velocity potential in the fluid domain can be found by solving a mixed boundary value problem which is defined by Eqs. (1), (4) and (5). After the solution is obtained, Eqs. (2) and (3) are then employed to update the position of and the potential values on the free surface using the same method as in [11], which gives the new information on the free surface for the solution at next time step. If floating bodies are included, the dynamic governing equations of the bodies should be solved at this point to update the velocity in Eq. (4). Similar to [11], the finite element method is used to solve the mixed boundary problems, in which the fluid domain is discretised into a set of small tetrahedral elements and the velocity potential is expressed in terms of a shape function, $N_J(x, y, z)$:

$$\phi = \sum_J \phi_J N_J(x, y, z), \tag{6}$$

where ϕ_J is the velocity potential at node J. Using the Galerkin method, Laplace's equation with corresponding boundary conditions can be discretised as

$$\int\int_{\forall}\int \nabla N_I \cdot \sum_{J\in S_P} \phi_J \nabla N_J \,\mathrm{d}\forall = \int_{S_n}\int N_I f_n \,\mathrm{d}S - \int\int_{\forall}\int \nabla N_I \cdot \sum_{J\in S_P} (f_p)_J \nabla N_J \,\mathrm{d}\forall \quad (I \notin S_P), \tag{7}$$

where S_P represents the Dirichlet boundary (such as the free surface), on which the velocity potential f_p is known and S_n represents the Neumann boundary (such as the wavemaker), on which the normal derivative of the velocity potential $f_n = \vec{n}\cdot\vec{U}(t)$ is given. It is noted that the term associated with the velocity potential on the free surface has appeared on the right-hand side of Eq. (7). Wu and Eatock Taylor [18] have found that this can ease the well-known singularity problem at the waterline between the free surface and rigid boundaries. Eq. (7) can further be written in the matrix form

$$[A]\{\phi\} = \{B\}, \tag{8}$$

where

$$\{\phi\} = [\phi_1, \phi_2, \phi_3, \ldots, \phi_I, \ldots]^{\mathrm{T}} \quad (I \notin S_P), \tag{9}$$

$$A_{IJ} = \int\int_{\forall}\int \nabla N_I \cdot \nabla N_J \,\mathrm{d}\forall \quad (I \notin S_P, J \notin S_P), \tag{10}$$

$$B_I = \int_{S_n}\int N_I f_n \,\mathrm{d}S - \int\int_{\forall}\int \nabla N_I \cdot \sum_{J\in S_P} (f_p)_J \nabla N_J \,\mathrm{d}\forall \quad (I \notin S_P). \tag{11}$$

The algebraic equation (8) is solved using a conjugate gradient iterative method with SSOR preconditioner and optimised parameters. Details about these have been discussed by Ma [24].

3. Mesh moving in QALE-FEM

As indicated in Section 1, the main task in the QALE-FEM is to move the mesh so as to accommodate large variations in the fluid domain. In order to achieve high efficiency and accuracy of the computation, the method to move the mesh should satisfy the following criteria:

- It must create satisfactory element shapes at all time steps.
- It must preserve reasonable refinement and distribution in regions of interest, such as those close to the free surface and structures.
- It must be computationally efficient.

Many methods to move the mesh in the ALE formulation have been suggested for the NS model. Often-used methods include the weighted average method [25], the transfinite mapping method [26], the method based on the solution of a linear elastic equation to define the new positions of nodes [8], the method based on the solution of Laplace's equation to find the velocity of mesh [27] and so forth. These methods either tend to make the mesh uniform, need a special mesh structure or require much computational time and so they are not perfect options satisfying the above criteria.

Another kind of method for moving the mesh, called the spring analogy method, has also been developed for the NS model, and mainly applied to aerodynamic problems without the free surface. The main idea behind the method is that nodes in a mesh are considered to be connected by springs. The whole mesh is then deformed like a spring system. The spring system may comprise linear springs along each element edge [28,29] or may consist of both linear and torsional springs [30–32], the latter applying a moment to

each node. The distinct advantages of the linear spring analogy system include requiring little CPU time and being very simple and easy to implement. The system has been successful in cases without extremely large distortion of elements and without nearly flat elements. Nevertheless, for the cases with very severely distorted and/or nearly flat elements, the system may produce negative volume elements. This drawback may be eliminated by adding torsional springs to the system. However, the CPU time required to deal with the torsional springs is significantly increased, particularly for three-dimensional cases.

In order to achieve a reasonable quality of the mesh at all time steps and avoid excessive computational time spent on moving the mesh, the following strategies are adopted to move the mesh in this paper:

- Ensuring that there are no nearly flat elements in the initial mesh.
- Considering interior nodes and boundary nodes separately.
- Considering nodes on the free surface and on rigid boundaries separately.
- Using relatively stiffer springs near the moving boundaries, such as the free surface.

The quality of initial mesh is to be achieved using mesh generators based on available technologies, such as mixed Delaunay triangulation and the advancing front technique [33]. The interior nodes are moved by using the linear spring analogy method. This method has been developed in computational aerodynamics, see e.g. [28], as indicated above. For completeness, it is only briefly described here. More details can be found in the references given. In this method, the nodal displacement are determined by

$$\Delta\vec{r}_i = \sum_{j=1}^{N_i} k_{ij}\Delta\vec{r}_j \Bigg/ \sum_{j=1}^{N_i} k_{ij}, \tag{12}$$

where $\Delta\vec{r}_i$ is the displacement at node I; k_{ij} is the spring stiffness and N_i is the number of nodes that are connected with node I. Eq. (12) can be interpreted as that the resultant spring forces acting on node I by all springs remain zero after all nodes are displaced. The value of the spring stiffness in the equation is usually chosen to be inversely proportional to the distance between two nodes in other applications [28,30]. For the applications concerned with in this paper, it is found that the spring stiffness is best taken to be

$$k_{ij} = \frac{1}{l_{ij}^2} \mathrm{e}^{\gamma[1+(z_i+z_j)/2d]}, \tag{13}$$

where l_{ij} is the distance between nodes I and J; z_i and z_j are the vertical coordinates of nodes I and J; d is the water depth; and γ is an coefficient that should be assigned a larger value if the springs are required to be stiffer at the free surface. The value of γ is taken as 1.7 in this paper but further numerical tests may be needed to choose the value of γ based on the wave steepness. To solve Eq. (12) for all interior nodes, iteration is required at each time step but it takes only a little CPU time according to our experience. The nodes at all rigid boundaries are also moved using the spring method.

3.1. Moving the nodes on the free surface

Special attention must be paid to nodes on the free surface because they play a decisive role in producing results of high accuracy for water wave problems. In order to track precisely the free surface, the node positions on the surface are determined by physical boundary conditions in Eq. (2), i.e., following the fluid particles, at most time steps. As indicated above, however, the nodes updated in this way may become too close to or too far from each other. To prevent this from happening, these nodes are relocated at some time steps (e.g., every 40 time steps). For the purpose of relocation, they are grouped into those on curved waterlines, such as the intersecting line between the free surface and vertical walls or the wavemaker, and those that do not lie on the waterlines. The nodes in the latter group are called inner-free-surface nodes.

The nodes in the two groups are treated separately. The nodes on the waterlines will be re-distributed by adopting a principle for self-adaptive mesh, see [34] for an example. The basic idea is to make the weighted arc-segment lengths between two successive nodes a constant along the curved waterlines, i.e.

$$\varpi_i \Delta s_i = C_s, \tag{14}$$

where ϖ is a weighted function, Δs_i the arc-segment length between two successive nodes and C_s a constant. The value of C_s is determined by using the fact that the total length of the curved waterline (L_s), should equal the sum of all arc-segment lengths, which gives

$$C_s = L_s \Big/ \sum \frac{1}{\varpi_i} \sum \frac{1}{\varpi_i} = L_s/\chi_s, \tag{15}$$

where $\chi_s = \sum 1/\varpi_i$. Therefore, if the weighted function is given, the arc-segment lengths can be evaluated by

$$\Delta s_i = \frac{L_s}{\varpi_i \chi_s}. \tag{16}$$

The distribution of Δs_i is controlled by the weighted function. An arc-segment tends to be smaller for a larger value of the weighted function or vice versa. In [34], the weight function is specified as

$$\varpi = 1 + \alpha \tilde{f}^{\beta}, \tag{17}$$

where α and β are two coefficients and $\tilde{f}$ is the function of the gradient of a variable, such as velocity. For the applications considered in this paper, $\tilde{f}$ is taken as a function of the curvature of the curved waterline, i.e., it is given by

$$\varpi_i = 1 + \alpha[(\kappa_i - \kappa_{\min})/(\kappa_{\max} - \kappa_{\min})]^{\beta}, \tag{18}$$

where κ_i is the curvature of the curve Δs_i; $\kappa_{\max}$ and $\kappa_{\min}$ are the maximum and minimum curvatures of the waterline, respectively. It is suggested in [34] that the coefficient α is determined by the arbitrarily specified maximum ($\Delta s_{s\max}$) and minimum ($\Delta s_{s\min}$) length of the arc-segment elements, i.e.

$$\alpha = \frac{\Delta s_{s\max}}{\Delta s_{s\min}} - 1. \tag{19}$$

In our applications, it is expected that the arc-segment length is smaller in areas of shorter waves and is larger in areas of longer waves. Hence, it is reasonable to specify

$$\frac{\Delta s_{s\max}}{\Delta s_{s\min}} = \frac{L_{\max}}{L_{\min}}, \tag{20}$$

where $L_{\max}$ and $L_{\min}$ are the maximum and minimum wave lengths, repetitively, which can be roughly estimated for problems considered. It should be noted that α subjected to Eq. (20) becomes zero for waves with single wavelength and so the arc-segment lengths are a constant after the nodes are re-distributed. Such a distribution of nodes is reasonable for these cases. The evaluation of β is not so straightforward. In [34], this value is determined in such a way that the minimum length of arc-segment elements obtained by Eq. (16) is approximately equal to the specified $\Delta s_{s\min}$ To achieve this, iteration must be performed. According to our numerical tests, a value of β in the range of 0.5–1.0 can lead to a satisfactory distribution of Δs_i when $L_{\max}/L_{\min} < 10$.

Once the nodes on the waterlines are redistributed, the inner-free-surface nodes will be moved by the spring analogy method, as used for the interior nodes. Nevertheless, there exists a difficulty, that is, how to ensure the nodes after moved are still on the free surface represented by discrete points. For simplicity, a method to achieve this is that the nodes are first moved in the projected plane of the free surface, i.e., calculating the values of x and y of new nodes using the spring analogy system, and then the elevations

of the free surface corresponding to them are evaluated by an interpolating method. In order to take into account of the local gradient of the free surface, the spring stiffness in moving the nodes in x- and y-directions, however, is determined, respectively, by

$$k_{ij}^{(x)} = \frac{1}{l_{ij}^2}\sqrt{1+\left(\frac{\partial\varsigma}{\partial x}\right)^2} \quad \text{and} \quad k_{ij}^{(y)} = \frac{1}{l_{ij}^2}\sqrt{1+\left(\frac{\partial\varsigma}{\partial y}\right)^2}, \tag{21}$$

where $\partial\varsigma/\partial x$ and $\partial\varsigma/\partial y$ are the local slopes of the free surface in the x- and y-directions, respectively. Inclusion of free-surface slopes in the spring stiffness ensures that the spring forces acting on a node are proportional to relative displacements between nodes measured along the curved free surface rather than along horizontal directions.

Two interpolating methods may be used to estimate the free surface elevations (i.e. coordinate z). One is to find which element each new node belongs to by using the coordinate x and y of the new nodes and then estimate the values of z using the shape function defined on this element. This method is similar to that one used for remeshing in [11]. The other is based on a moving least square (MLS) method, in which the value of z is determined by using the information at a group of old nodes in such a way that the error is minimised. This method has been frequently used to form the interpolating function in meshless methods and details about it may be found in, e.g. [35]. Both of these two methods work well, though the latter needs more computational time and leads to rather more accurate results according to our numerical tests. In this paper, the MLS method is used. Other information, such as the velocity potential at the new nodes is also estimated by the method. It should be noted that the technique for moving the inner-free-surface nodes described here may not work if overturning/breaking waves are involved, which are beyond our consideration based on the FNPT model.

4. Velocity calculation on the free surface in QALE-FEM

It is crucial in simulating water waves to evaluate the fluid velocities on the free surface because they are used to update the information on the surface every time step. The velocity at a node may be estimated by using a finite difference technique from the velocity potentials at this node and nodes connected to it. The approach is quite efficient. However, since the neighbours of a node on the free surface are distributed either on or below the surface, the normal (or nearly vertical) component of the velocity estimated by the approach generally possesses relatively low accuracy, which is understandable from the fact that backward or forward finite difference schemes approximating a derivative have a lower order of accuracy than a central scheme. In order to enhance the overall accuracy, Ma et al. [11] suggested that the horizontal components of the velocities at nodes on the free surface are evaluated separately from their vertical components. For estimating the vertical component, they developed a three-point formula that needs the velocity potentials at the node considered and at two other nodes on the same vertical line as this node, which are next but just below the free surface. After the vertical component is found, the horizontal components are computed by averaging those given by the difference of the velocity potentials at all neighbour nodes on the free surface. This approach is very efficient and accurate. However, it is limited to structural meshes with vertical grid lines.

In this section, the above approach will be extended to unstructured meshes generally without vertical grid lines. The basic idea of the new approach is similar to the above approach. The main differences are that (1) the vertical line is replaced by a normal line perpendicular to the free surface at the node considered; (2) the two nodes on the vertical line are replaced by two points on the normal line, which do not necessarily coincide with any nodes; and (3) the normal component of the velocity is found before computing the components in tangential directions. More details are given below.

Consider node I on the free surface of an unstructured mesh with nodes J_k ($k = 1,2,3,\ldots,m$) as its neighbours on the free surface. A normal line is drawn from node I to the inner domain and two points P_{I1} and P_{I2} are chosen on this line, as shown in Fig. 2. The distance between I and P_{I1} is h_{I1} and the distance between P_{I1} and P_{I2} is h_{I2}. The normal component ($\vec{v}_n$) of the velocity at node I is calculated by

$$\vec{v}_n = \left[\frac{2}{3h_{I1}}\left(\frac{2h_{I1}+h_{I2}}{h_{I1}+h_{I2}}+\frac{1}{2}\right)\phi_I - \left(\frac{2}{3h_{I2}}+\frac{1}{h_{I1}}\right)\phi_{P_{I1}} + \frac{2}{3h_{I2}}\left(\frac{h_{I1}}{h_{I1}+h_{I2}}\right)\phi_{P_{I2}}\right]\vec{n}, \tag{22}$$

where $\vec{n}$ is the unit normal vector of the free surface at node I. This equation is similar to Eq. (16) in [11]. Nevertheless, $\phi_{P_{I1}}$ and $\phi_{P_{I1}}$ here are not nodal values and may be found using the method discussed in Section 4.1 below. The normal vector is taken as the average of the normal vectors of all surface elements (such as I–J_1–J_2) with node I as one of their nodes.

In order to estimate the velocity in tangential directions, the unit tangential vector ($\vec{\tau}$) is required. The vector may be determined by using $\vec{\tau}_x \perp \vec{n}$, $\vec{\tau}_x \| \vec{e}_x$, $\vec{\tau}_y \perp \vec{n}$ and $\vec{\tau}_y \| \vec{e}_y$, where $\vec{e}_x$ and $\vec{e}_y$ are the unit vectors in the x- and y-directions, respectively. The tangential components of the velocity are related to the difference of the velocity potential between any pair of nodes containing node I and one of nodes $J_1, J_2, \ldots, J_m$ by

$$\vec{v}_{\tau_x}\cdot\vec{l}_{IJ_k} + \vec{v}_{\tau_y}\cdot\vec{l}_{IJ_k} = \vec{l}_{IJ_k}\cdot\nabla\phi - \vec{v}_n\cdot\vec{l}_{IJ_k} \quad (k = 1,2,3,\ldots,m), \tag{23}$$

where $\vec{l}_{IJ_k}$ is the unit vector from node I to node J_k; $\vec{v}_{\tau_x}$ and $\vec{v}_{\tau_y}$ represent the velocity components in $\vec{\tau}_x$ and $\vec{\tau}_y$ directions, respectively. The number (m) of equations in Eq. (23) is usually larger than 2, the number of unknowns. In order to use all the equations and enhance the accuracy, the least square method is adopted to find $\vec{v}_{\tau_x}$ and $\vec{v}_{\tau_y}$. Once the three components are obtained, the velocity components in x-, y- and z-directions can readily be computed by projecting them on these directions.

4.1. Velocity potential at points P_{I1} and P_{I2}

There are two issues associated with the velocity potential at points P_{I1} and P_{I2} involved in the above velocity computations. One is how to choose the positions of the points and the other is how to estimate the values of the potential at these points. According to our experience, point P_{I1} should be located in the element connected to node I and point P_{I2} should be in another element next to the previous element in the normal direction. This is rational from simple reasoning. If the two points are too close to node I so that they fall into one element, the values of potential calculated at these points are based mainly on the information of one element and so the estimated difference by using these values may possess low accuracy. On the other hand, if the points are too far from each other or from node I, the error of velocity estimated using Eq. (22) may also be big because the error increases with distances between them.

In order to ensure the two points to be in the desired elements, h_{I1} and h_{I2} in Eq. (22) can be determined by

$$h_{I1} = h_{I2} = \varepsilon\hbar, \tag{24}$$

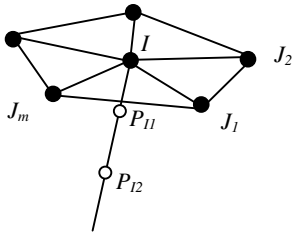

Fig. 2. Sketch of nodes around node I.

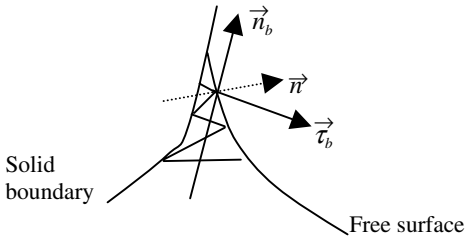

Fig. 3. Nodes near a solid boundary.

where ε is the coefficient and $\hbar$ is the distance from node I to the intersecting point of the normal line I–P_{I1}–P_{I2} with the element surface formed by other three interior nodes of the element connected to node I. Numerical tests show that ε may be a value in the range of 0.6–0.9 and that numerical results obtained are not sensitive to its specific value. The numerical tests also show that the value of $\hbar$ does not necessarily have to be calculated every time step. In fact, it is calculated only at the first time step for cases presented in Section 5.

With the values of h_{I1} and h_{I2} determined, the values of the potential at points P_{I1} and P_{I2} can be estimated either by using the shape function defined on the elements or by the MLS method mentioned above. The former method needs less computational time but gives less accurate values, particularly in cases with large gradients. The latter requires more computational time but results in more accurate potential values. If the waves to be simulated are very steep, the latter should be used; otherwise the former would be the better choice. Because we aim at steep waves, the latter is used in this paper.

4.2. Special treatment for nodes near solid boundaries

It may become impractical to use the above method to calculate the velocity for nodes near a solid boundary because the normal line I–P_{I1}–P_{I2} may intersect with the solid boundary (see Fig. 3). If this happens, either point P_{I2} is put outside the fluid domain if h_{I1} and h_{I2} are still estimated by Eq. (24) or the two points (P_{I1} and P_{I2}) are contracted into one element. Both situations may degrade the results.

In order to avoid such problems, it is proposed that the normal line (coinciding with vector $\vec{n}$ in Fig. 3) at a node near a solid boundary is replaced by a line (coinciding with vector $\vec{n}_b$) obtained by rotating the normal line to the direction perpendicular to the normal vector, passing the node considered, of the boundary surface. Correspondingly, $\vec{\tau}$ is replaced by $\vec{\tau}_b$ that is determined by $\vec{n}_b$, $\vec{e}_x$ and $\vec{e}_y$ using the similar method to that for $\vec{\tau}$. After doing so, Eqs. (22) and (23), by substituting $\vec{n}_b$ and $\vec{\tau}_b$ for $\vec{n}$ and $\vec{\tau}$, are still used to compute the velocity at the node. Using this treatment, the velocity components in x-, y-, and z-directions are directly obtained when the solid boundary is vertical. It should be noted that this treatment may not work well when the angle between the free surface and the solid boundary becomes very small. The situation with very small angles can occur when the wave tends to overturning, which is not considered in this paper, as indicated above.

5. Numerical validations

In this section, the QALE-FEM method is validated by comparing its numerical predictions with analytical solutions and published results using other methods or experiments. Two kinds of problems are considered: one is nonlinear water waves (regular and random) generated by a wavemaker in a tank; and the other is the interaction between the water waves and periodic bars on the tank-bed. For all the cases

in this section, the maximum and minimum wave lengths in Eq. (20) are assumed to be the same. In the following, all parameters with a length scale are nondimensionalised by the water depth d and other parameters by

$$t \to \tau\sqrt{d/g} \quad \text{and} \quad \omega \to \omega\sqrt{g/d}.$$

5.1. Water waves generated by a wavemaker

Water waves generated by a piston wavemaker in the tank are considered in the first instance. The waves may be monochromatic, bichromatic and random depending on the motion of the wavemaker. The meshes used are similar to that in Fig. 4 but much finer.

The cases of monochromatic waves are first modelled, for which the motion of the wavemaker is governed by

$$S(\tau) = -a\cos(\omega\tau), \tag{25a}$$

$$U(\tau) = a\omega\sin(\omega\tau), \tag{25b}$$

where $S(\tau)$ is the displacement of the wavemaker, $U(\tau)$ is its velocity, a and ω are, respectively, its amplitude and frequency. When the amplitude of the wavemaker is very small, the steepness of the generated waves is also very small. The numerical results for such a case can be compared with the linearised analytical solution in [42]. For this purpose, a case with $a = 0.0041$ and $\omega = 1.45$ is simulated in a tank of length $L \approx 14.7$. The mesh is unstructured and the number of elements is about 78,060. The time step is 0.021666, about 200 steps in each period. The wave profiles at two different instants ($t = 10T$ and $15T$, where $T = 2\pi/\omega$ is the

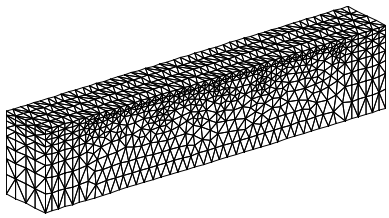

Fig. 4. Illustration of initial mesh used for wavemaker problems.

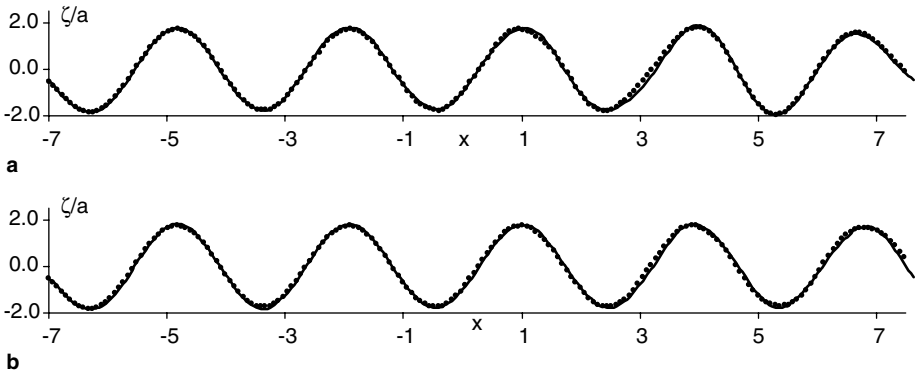

Fig. 5. Comparison of wave profiles with the analytical solution for $\omega = 1.45$ and $a = 0.0041$ (solid line: QALE-FEM; dots: analytical solution). (a) $\tau = 10T$; (b) $\tau = 15T$.

period) are plotted in Fig. 5, which shows that the numerical results are in very good agreement with the corresponding analytical solution.

The numerical results in Fig. 5 are also assessed by estimating relative errors. The relative error (E_r) is defined as

$$E_r = \frac{\|\varsigma - \varsigma_a\|}{\|\varsigma_a\|},$$

where $\|\varsigma\| = \int_{A_e} \varsigma^2 \, dA$, ς_a is an analytical wave elevation and A_e is the area over which the error is estimated. Because the accuracy of the waves within the damping zone should not be of concern, A_e equals the area of the free surface minus the part of the damping zone. The relative errors evaluated in this way for the results in Fig. 5 are less than 0.5%.

The behaviour of the QALE-FEM is then investigated by simulating waves of a larger amplitude and strong nonlinearity. The amplitude of the wavemaker is taken as 0.082. The frequency (ω) and tank length are the same as that in Fig. 5. The steepness of the generated waves is about 0.08. For such steep waves, the linearised solution should not be considered valid. In order to validate the QALE-FEM in this case, its results are compared with those obtained by using conventional FEM described in [11]. The length of the time step (also 200 steps in each period) is the same in both methods, whereas the type of mesh and the number of elements are different. When using the conventional FEM [11], the mesh is structured and the number of elements is 75,264. When using the QALE-FEM, the mesh is unstructured and the number of elements is about 78,060. The latter is also tested using a larger number of elements (133,632) but no significant difference in results was found. The wave profiles at time $\tau = 10T$ and $15T$ from these two methods are depicted in Fig. 6. The agreement between them is quite good. The relative error estimated by the same method for Fig. 5 is found to be less then 1%.

The QALE-FEM has also been used to simulate bichromatic waves. These waves are generated by the following motion of the wavemaker:

$$S(\tau) = -a_1 \cos(\omega_1 \tau) - a_2 \cos(\omega_2 \tau), \tag{26a}$$

$$U(\tau) = a_1 \omega_1 \sin(\omega_1 \tau) + a_2 \omega_2 \sin(\omega_2 \tau), \tag{26b}$$

where a_1 and a_2 are the amplitudes corresponding to the components with frequencies ω_1 and ω_2, respectively. As an example, the values for these parameters are assigned as $a_1 = 0.016$, $a_2 = 0.5a_1$, $\omega_1 = 1.45$ and $\omega_2 = 2.03$. The tank has the same length and the mesh is the same as for Fig. 6. The time step is about 0.01548, about 200 steps in each period given by $2\pi/\omega_2$. For this case, the wave history recorded at a fixed

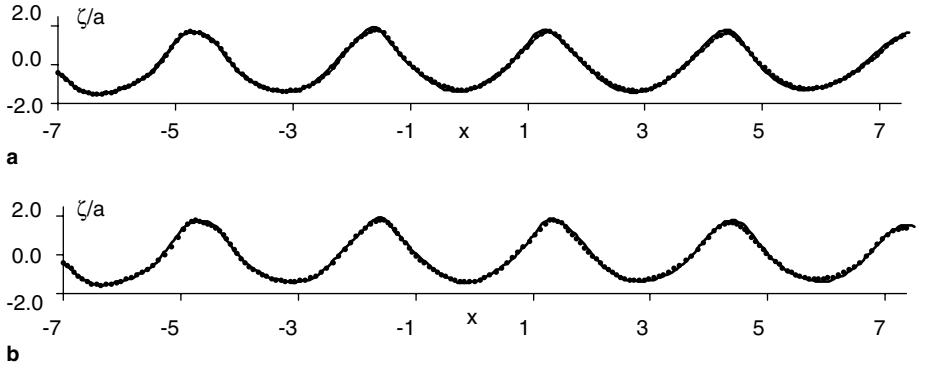

Fig. 6. Comparison of wave profiles for $\omega = 1.45$ and $a = 0.082$ (solid line: QALE-FEM; dots: conventional FEM [11]). (a) $\tau = 10T$; (b) $\tau = 15T$.

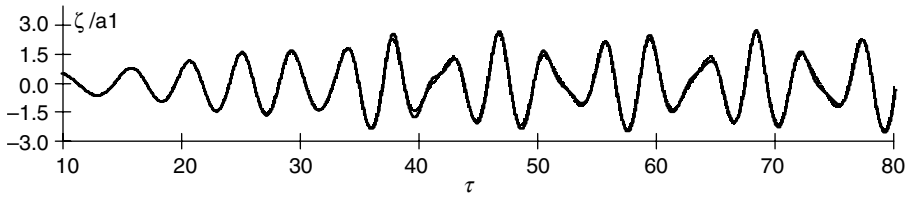

Fig. 7. Comparison of wave histories for $a_1 = 0.016$, $a_2 = 0.5a_1$, $\omega_1 = 1.45$ and $\omega_2 = 2.03$ (solid line: QALE-FEM; dotted lines: conventional FEM [11]).

point is plotted in Fig. 7, together with the results from the conventional FEM. Again, it is observed that results from the two methods are in very good agreement and the relative error is in the same level as for Fig. 6.

As another example, the QALE-FEM is applied to simulating random waves. In order to compare with the experimental results given by Nestegard [43], the same motion of the wavemaker as those described in [43] is used, which is specified by a Fourier series with different scaling factors (α_{random}). The water waves generated by this motion are focused at a point in the tank to form a large and steep wave. To model this case, the tank length is chosen as 20. The time step is about 0.0242 and the number of elements is 183,240. Fig. 8 shows the wave histories recorded at $x = 3.436$ (where the wave is expected to focus) together with the experimental data provided by Nestegard [43] for the scaling factors equal to 0.612 and 0.749. It can be seen that the agreement of the numerical results with the experimental data is satisfactory. Particularly, the largest wave crests are excellently predicted by the numerical analysis.

5.2. Reflection due to periodic bars on the seabed

The QALE-FEM is now employed to simulate the interaction between waves and periodic bars on the seabed. Since the experimental demonstration by Heathershaw [36], the problem has been studied by many researchers using various mathematical models with particular attention paid to Bragg resonance that leads to large reflecting waves. These models were developed by making various approximations, including linear perturbation approach [37], multiple scale analysis [38], mild-slope approach [39], fully linear analysis [40] and so on. The results obtained from these models agreed well with experiments carried out by Heathershaw [36] and Davies and Heathershaw [37] in cases with small surface wave and bar wave steepness. Liu and Yue [41] performed a fully nonlinear analysis using a spectral method and pointed out that the

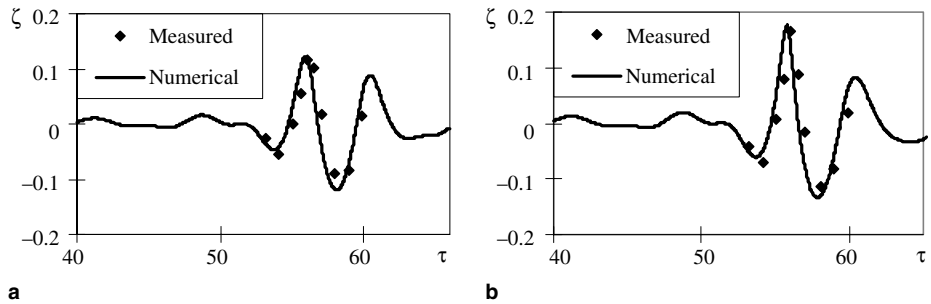

Fig. 8. Comparison of wave histories at $x = 3.436$ with measured data given by Nestegard [43]. (a) ($\alpha_{\text{random}} = 0.612$); (b) ($\alpha_{\text{random}} = 0.749$).

nonlinear effects may cause the downshift of reflection coefficient curves compared with results from the simplified models such as in [38].

In this section, the numerical results obtained by using the QALE-FEM will be compared with published experimental data and analytical/numerical solutions, with particular attention paid to the reflecting wave properties near the Bragg resonance. The main purpose of the comparisons is to further validate the new numerical method. Apart from this, certain results corresponding to larger wave amplitudes will also be presented in order to illustrate the nonlinear effects on the reflection.

The two cases to be considered are the same as those in [37], i.e., bar patches with 4 and 10 sinusoidal bars on the seabed, respectively. The wave generator motions are as specified by Eq.(25) For ease of description, the side of the bar patch near the wavemaker is called the front side, and the other side the lee side. The initial meshes used are similar to that illustrated in Fig. 9 but much finer. The reflection coefficients, defined by $K_r = A_r/A_i$, where A_r and A_i are the amplitudes of reflecting and incident waves, are calculated from wave histories recorded at a series of points along the tank by using the same method as in [37], in which it is assumed that the wave consists of incident and reflected waves with the same frequencies. It should be noted that for the purpose of computing the reflection coefficients, the time history to be used at a point must start from the time when the wave reflected from the lee side has arrived at the point and end before the wave reflected from the front side travels back to the point after interacting with the wavemaker. Otherwise, either the reflecting waves are not fully developed at the point or affected by the re-reflecting waves from the wavemaker. The start (t_{st}) and end (t_{end}) times may be estimated by

$$t_{st} = (L_{wp} + 2L_{bp2})/C_g, \tag{27a}$$

$$t_{end} = (3L_{wp} + 2L_{bp1})/C_g, \tag{27b}$$

where C_g is the group velocity of the water wave; L_{wp} is the distance from the wavemaker to the point considered; L_{bp1} and L_{bp2} are the distances from the point to the front and lee sides of the bar patch, respectively.

First considered are the cases with small wave amplitudes. For these cases, the water waves are generated by small amplitudes ($a = 0.02$ for 4 bars and 0.005 for 10 bars) and the resulting wave steepness (H/λ, where H and λ are the water wave height and length, respectively) is less than 0.002. In order to compare our results with experimental data in [37], the dimensionless bar wave number ($k_b d$) is assigned a value of $\pi/10$, the ratios of the bar amplitude (a_b) to the water depth are taken, respectively, as $a_b/d = 0.32$ for 4 bars and $a_b/d = 0.16$ for 10 bars. The wave histories recorded at two points about 5 bar-lengths before the front side of the bar patch are used. Reflection coefficients near the resonant condition ($2k/k_b = 1$, where k is the water wave number) are presented in Fig. 10 together with experimental data from [37]. For the case with 10 bars, the nonlinear numerical results from [41] and analytical results from the simplified model [38] are also included. For the case with 4 bars, the analytical results from [40] are plotted apart from the experimental and our numerical results. From Fig. 10(a) for this case, it can be seen that the numerical results obtained by using the QALE-FEM method agree well with the analytical results given in [40] and satisfac-

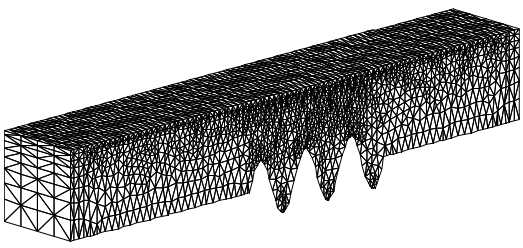

Fig. 9. Illustration of initial mesh for scattering problems due to periodic bars.

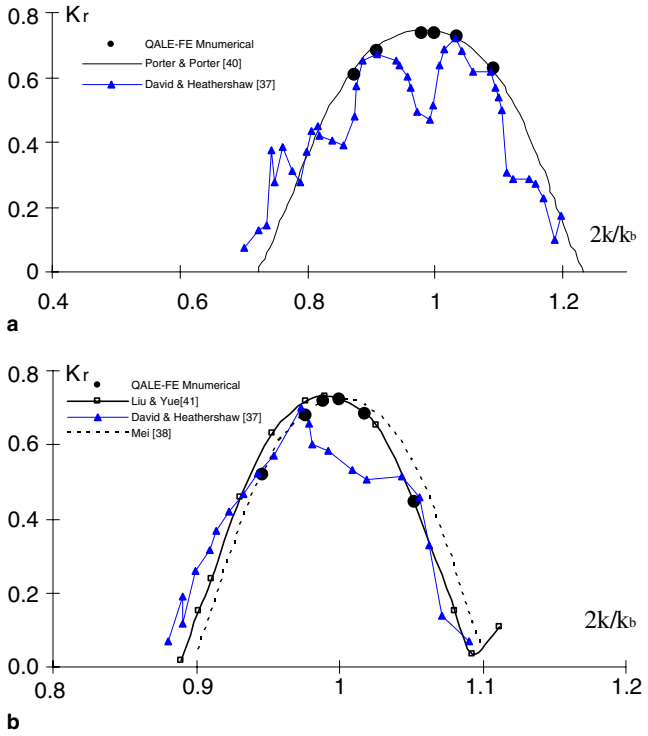

Fig. 10. Reflection coefficients ($K_r = A_r/A_i$ with A_r and A_i being the amplitudes of reflection and incident waves, respectively) with $k_b d = \pi/10$. (a) 4 bars, $a_b/d = 0.32$; (b) 10 bars, $a_b/d = 0.16$.

torily with experimental data in [37]. Fig. 10(b) for 10 bars indicated that our numerical results are almost identical to those from [41] and closer to the experimental data than the analytical solution based on the simplified model [38] on the side of $2k/k_b > 1$. On the side of $2k/k_b < 1$, our results differ slightly from [41] but are closer to the experimental data in [37] and the analytical results from the simplified models [38].

To further show the properties of the reflected waves, the wave profiles at different instants for 4 bars are plotted in Fig. 11, in which the coordinate system is shifted so that its origin is at the centre of the bar patch, and the bar patch is located in the range of $-2 < x/\lambda_b < 2$ ($\lambda_b = 2\pi/k_b$). It can be observed that when the incident wave reaches the bars, the reflected wave begins to be produced. The reflected wave propagates towards the wavemaker, is superimposed onto the incident wave and makes the resultant wave before the front side ($x/\lambda_b = -2$) higher than the incident wave. It can also be observed that the wave after the lee side ($x/\lambda_b = 2$) is considerably smaller than the wave before the front side, as expected.

In order to investigate the nonlinear effects, the case with 4 bars is simulated with different amplitudes. All other parameters except for the amplitude are the same as those for Fig. 10. The reflection coefficients corresponding to $2k/k_b \approx 1$ are presented in Fig. 12. In Fig. 12(a), the coefficients at different positions are plotted together with the experimental results and analytical solution from [37]. It can be seen that the reflection coefficients are close to the linear analytical solution when the amplitude is small but close to the measured data when the amplitude is larger. In addition, the reflection coefficients before the front side ($x/\lambda_b = -2$) tend to decrease with the increase of the amplitudes, though the reduction is not very significant. To further show this trend, the numerical reflection coefficients at a point $x/\alpha_b = -4$ are plotted in

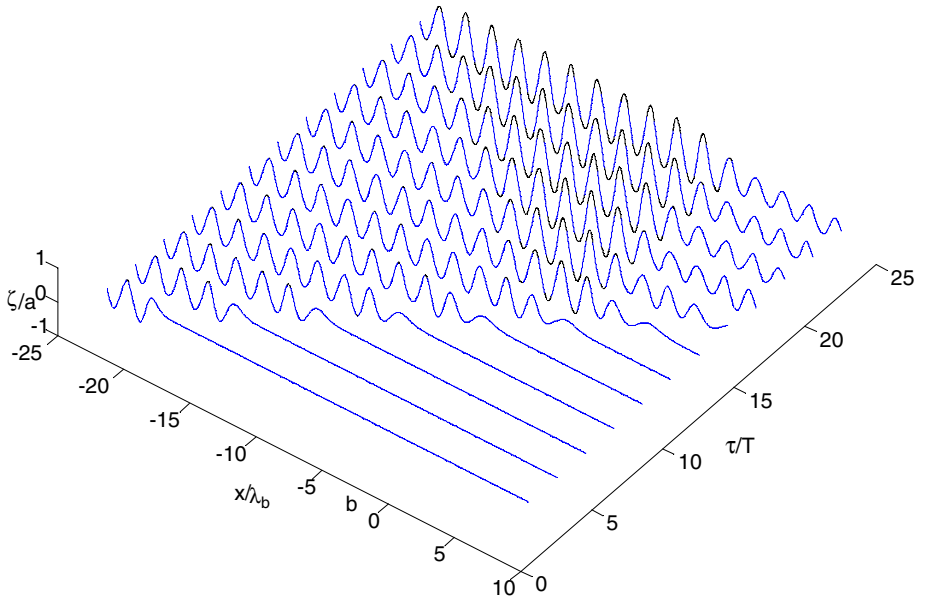

Fig. 11. The wave profiles at different instants for 4 bars.

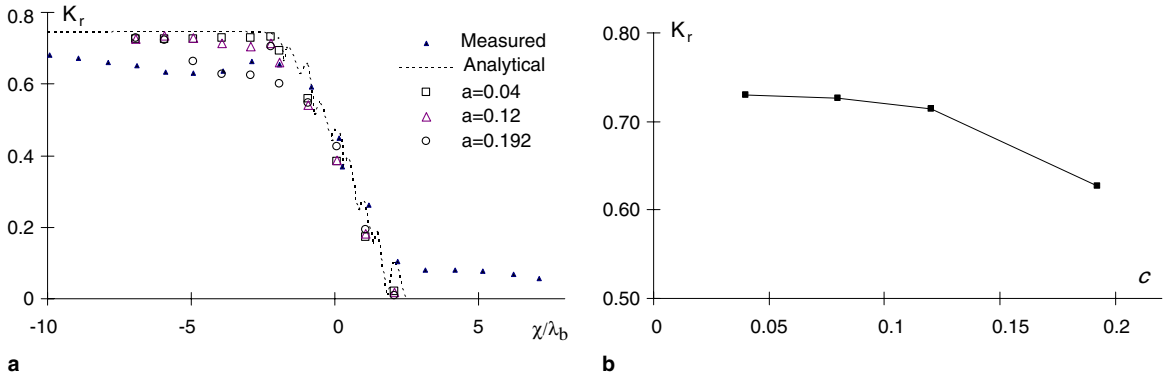

Fig. 12. Nonlinear effects on reflection coefficients at $2k/k_b \approx 1$ for 4 bars. (a) K_r at different positions; (b) K_r at $x/\lambda_b \approx -4$.

Fig. 12(b). It should be noted that the method for estimating the reflection coefficients is the same as in Fig. 10 in order to compare the results with those in [37]. However, when the nonlinearity becomes strong, high-order waves are involved. The reflection coefficients found in this way correspond only to the wave with the same frequency as the first-order wave and do not include the reflection of high-order waves.

In order to look at overall reflection of nonlinear waves by the bar patch, the shapes of wave profiles for the same case are illustrated in Fig. 13. In this figure, the wave profiles for smaller and larger amplitudes are depicted to show the different reflection properties. As can be seen, the wave profiles on the left of the bar patch for the smaller amplitude seems to be formed by superimposing two harmonic waves with the same

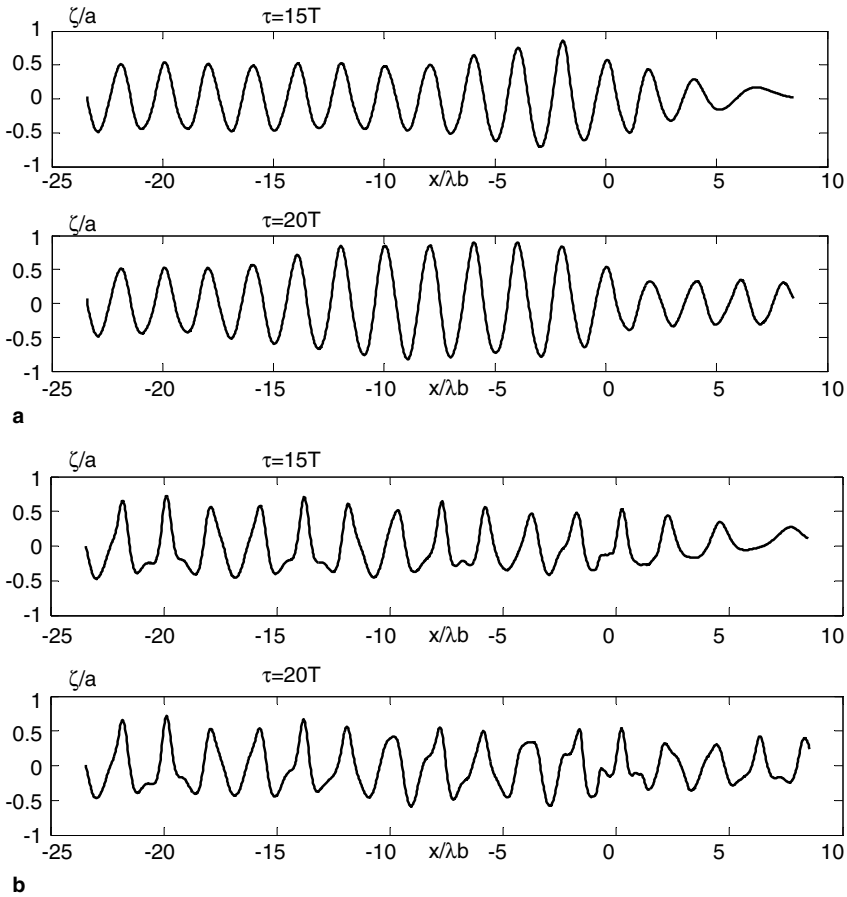

Fig. 13. Wave profiles corresponding to different amplitudes for 4 bars. (a) $a = 0.01$; (b) $a = 0.192$.

length travelling in opposite directions (so the wave become higher) but the shape is still similar to that of harmonic waves. For the larger amplitude, the wave amplitude on the left of the bar batch seems not to be changed dramatically by the reflection waves, instead, the shape of the waves is significantly modified.

6. Quality of mesh during simulation and efficiency of mesh moving

A significant development in this paper is to move, instead of re-generating, the unstructured mesh at every time step when simulating water waves based on the FNPT model. As pointed out in Section 3, the mesh obtained should have satisfactory element shapes and preserve the reasonable refinement and distribution in regions of interest, such as those close to the free surface and the bars on the tank-bed at all time steps. In order to show the quality of mesh moved by using the method discussed in Section 3, Figs. 14 and 15 are presented for a case with 4 bars. Fig. 14(a) shows a part of the initial unstructured mesh while Fig. 14(b) illustrates the enlarged mesh in areas near the two sides of the bar patch. Fig. 15 depicts the corresponding part of the mesh at about $\tau = 332$. These figures demonstrate that the original refinement and distribution are kept and all elements are of satisfactory shape after long time simulation. In addition, negative elements, which are of concern when using the linear spring analogy method, do not appear in the

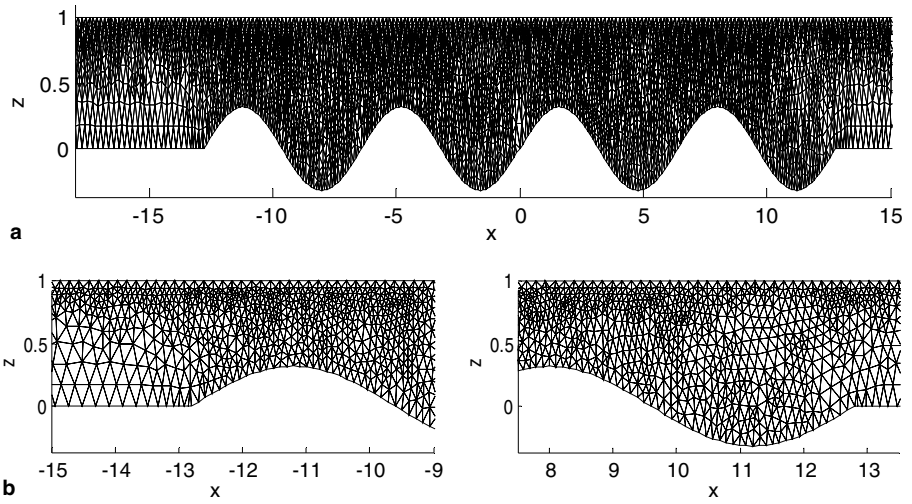

Fig. 14. Initial mesh for 4 bars.

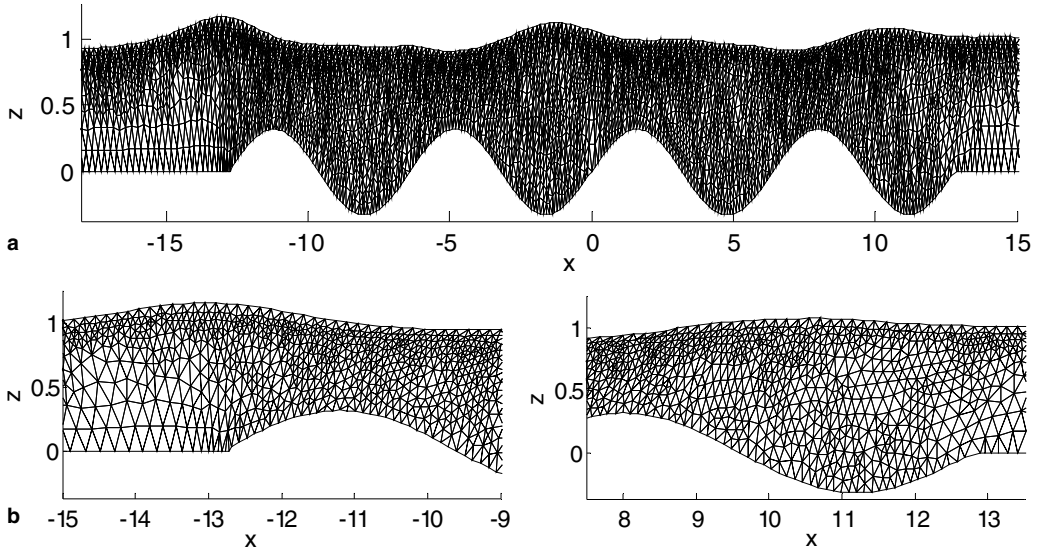

Fig. 15. Mesh at $\tau \approx 332$ (4 bars, $a = 0.192$).

simulation. This implies that the techniques used for moving mesh described in Section 3 work very well. Nevertheless, certain changes in the sizes and shapes of individual elements are observed and expected because the fluid domain varies with propagation of waves. It is these changes that make it possible to conform to the moving boundaries at all time steps and so to achieve satisfactory results as demonstrated in Section 5.

Another concern about the QALE-FEM is the computational cost of moving the mesh. The efficiency of moving the mesh can be deduced by comparing the computational time required by the QALE-FEM with that required if a conventional FEM with unstructured mesh regenerated at each time step is employed. For

this purpose, a case of nonlinear water waves without bars is tested, in which, $a = 0.082$, $\omega = 1.45$ and $L = 14.7$. The fluid domain is discretised into about 133,632 elements. The time-step length is taken as $t = 0.021666$ and 3000 steps are marched. The case is run on a PC (Pentium 2.53 GHz processor, 1G RAM). The CPU time spent on generating the mesh is about 53 s. After the mesh is generated, the CPU time spent on all the other calculations is on average about 8 s each time step, including 0.09 s for moving interior nodes and 1 s for adjusting all nodes on the free surface. In this simulation, adjustment of nodes on the free surface is performed every 40 time steps and takes about 39 s each time, so the additional CPU time for this is roughly 1 s each time step. If the same case had been simulated using a conventional FEM with the same mesh regenerated at each time step, the total CPU time on each time step would have been about 60 s. This implies that the QALE-FEM requires less than 15% of the CPU time required by the conventional FEM. It should be noted that the time spent on adjusting nodes on the free surface depends on how often it is undertaken. Its frequency depends on the number of time steps used each period and on the wave steepness. The more time steps in each period, the less frequently adjustment has to be performed. On the other hand, the steeper the waves, the more often adjustment is needed. According to our experience so far, the adjustment frequency is unlikely to be less than every ten time steps if a reasonable time step is chosen. Even with a frequency of every ten time steps to adjust the nodes on the free surface, the CPU time required by using the QALE-FEM is still considerably less than that by the conventional FEM. Therefore, it can be confidently stated that the QALE-FEM is much faster than the conventional FEM when using unstructured meshes.

7. Conclusion

In this paper, the QALE-FEM is developed to simulate nonlinear water waves based on the FNPT model. In this method, the boundary value problem about the velocity potential is solved by using a finite element method and the mesh is moved in order to conform to the wavy free surface and other moving boundaries. The method allows the efficient use of unstructured meshes without the need to regenerate it at every time step, which is a necessary and very costly feature of the conventional FEM. To achieve overall high efficiency and accuracy, several numerical techniques involved in the QALE-FEM have been developed, including the method to move interior nodes, the technique to re-distribute the nodes on the free surface, the scheme to compute velocities, and so on.

The newly developed method has been validated by comparing its numerical predictions with published analytical solutions, experimental data and results from other methods. The validation cases included the monochromatic, bichromatic and random waves and also included the interaction between waves and periodic bars on the seabed. In all cases, the results given by the present method agree well with published data. Assessments are made on the efficiency of moving mesh and quality of elements obtained by moving the mesh, which shows that unstructured mesh quality is satisfactorily maintained and the QALE-FEM requires only a small fraction of CPU time that would be spent on using the conventional FEM if unstructured mesh is used.

Although the newly developed method is applied only to problems without floating bodies in this paper, it is not very difficult to extend it to deal with problems with floating bodies. When floating bodies are involved, the nodes on the body surfaces may be adjusted in a way similar to that for the free surface.

Acknowledgements

This work is sponsored by EPSRC, UK (GR/R78701), for which the authors are most grateful. The authors also thank Dr. Ping Dong, Dundee University, UK, for him bringing the problem about periodic bars to our attention.

References

[1] D.M. Greaves, Simulation of interface and free surface flows in a viscous fluid using adaptive quadtree grids, Int. J. Numer. Methods Fluids 44 (2004) 1093–1117.

[2] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free surface boundaries, J. Comput. Phys. 39 (1981) 210–225.

[3] S. Navti, R. Lewis, C. Taylor, Numerical simulation of viscous free surface flow, Int. J. Numer. Methods Heat Fluid Flow 18 (1998) 445–464.

[4] W. Yue, C.L. Lin, V.C. Patel, Numerical simulation of unsteady multidimensional free surface motions by level set method, Int. J. Numer. Methods Fluids 42 (2003) 853–884.

[5] J. Donea, P. Fasoli-Stella, S. Giuliani, Finite element solution of transient fluid-structure problems in Lagrangian coordinates, Proceedings of the International Meeting on Fast Reactor Safety and Related Physics, Chicago 3 (1976) 1427–1435.

[6] Duarte Fabián, Gormaz Raúl, Natesan Srinivasan, Arbitrary Lagrangian–Eulerian method for Navier–Stokes equations with moving boundaries, Comput. Methods Appl. Mech. Eng. 193 (2004) 4819–4836.

[7] Braess Henning, Wriggers Peter, Arbitrary Lagrangian–Eulerian finite element analysis of free surface flow, Comput. Methods Appl. Mech. Eng. 190 (2000) 95–109.

[8] M. Souli, J.P. Zolesio, Arbitrary Lagrangian–Eulerian and free surface methods in fluid mechanics, Comput. Methods Appl. Mech. Eng. 191 (2001) 451–466.

[9] C. Lachaume, B. Biausser, S.T. Grilli, P. Fraunie, S. Guignard, Modeling of breaking and post-breaking waves on slopes by coupling of bem and vof methods, Proc. Int. Offshore Polar Eng. Conf. (2003) 1698–1704.

[10] G.F. Clauss, U. Steinhagen, Numerical simulation of nonlinear transient waves and its validation by laboratory data, in: Proceedings of the 9th International Offshore and Polar Engineering Conference, Brest, France, 1999, pp. 368–375.

[11] Q.W. Ma, G.X. Wu, R. Eatock Taylor, Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 1: Methodology and numerical procedure, Int. J. Numer. Methods Fluids 36 (2001) 265–285.

[12] Q.W. Ma, G.X. Wu, R. Eatock Taylor, Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 2: Numerical results and validation, Int. J. Numer. Methods Fluids 36 (2001) 287–308.

[13] M.S. Longuet-Higgins, E.D. Cokelet, The deformation of steep waves on water: I. A numerical method of computation, Proc. R. Soc. Lond. A 350 (1976) 1–26.

[14] M.S. Celebi, M.H. Kim, R.F. Beck, Fully nonlinear 3D numerical wave tank simulation, J. Ship Res. 42 (1998) 33–45.

[15] S.T. Grilli, P. Guyenne, F. Dias, A fully nonlinear model for three-dimensional overturning waves over arbitrary bottom, Int. J. Numer. Methods Fluids 35 (2001) 829–867.

[16] Y. Cao, W.W. Schultz, R.F. Beck, Three-dimensional desingularised boundary integral method for potential problems, Int. J. Numer. Methods Fluids 12 (1991) 785–803.

[17] P. Wang, Y. Yao, M. Tulin, An efficient numerical tank for nonlinear water waves, based on the multi-subdomain approach with BEM, Int. J. Numer. Methods Fluids 20 (1995) 1315–1336.

[18] G.X. Wu, R. Eatock Taylor, Finite element analysis of two dimensional non-linear transient water waves, Appl. Ocean Res. 16 (1994) 363–372.

[19] G.X. Wu, R. Eatock Taylor, Time stepping solution of the two dimensional non-linear waveradiation problem, Ocean Eng. 22 (1995) 785–798.

[20] X. Cai, H.P. Langtangen, B.F. Nielsen, A. Tveito, A finite element method for fully nonlinear water waves, J. Comput. Phys. 143 (1998) 544–568.

[21] J.H. Westhuis, Andonowati, Applying the finite element method in numerically solving the two dimensional free-surface water wave equations, in: Proceedings of the 13th International Workshop on Water Waves and Floating Bodies, Alphen aan den Rijn, The Netherlands, 1998, pp. 171–174.

[22] G.X. Wu, Z.Z. Hu, Simulation of nonlinear interactions between waves and floating bodies through a finite-element-based numerical tank, Proc. R. Soc. A 460 (2004) 2050, 3037–3058.

[23] C. Heinze, Nonlinear hydrodynamic effects on fixed and oscillating structures in waves, Ph.D. Thesis, Department of Engineering Science, Oxford University, 2003.

[24] Q.W. Ma, Numerical simulation of nonlinear interaction between structures and steep waves, Ph.D. Thesis, Department of Mechanical Engineering, University College London, UK, 1998.

[25] J.L.F. Aymone, E. Bittencourt, G.J. Creus, Simulation of 3D metal-forming using an arbitrary Lagrangian–Eulerian finite element method, J. Mater. Process. Technol. 110 (2001) 218–232.

[26] M.S. Gadala, Recent trends in ALE formulation and its applications in solid mechanics, Comput. Methods Appl. Mech. Eng. 193 (2004) 4247–4275.

[27] R. Löhner, C. Yang, Improved ALE mesh velocities for moving bodies, Commun. Numer. Methods Eng. 12 (1996) 599–608.

[28] J.T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA Paper 89-0115, 27th AIAA Aerospace Sciences Meeting, January 1989.

[29] W.K. Anderson, Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation, AIAA Paper 97-0643, 35th AIAA Aerospace Sciences Meeting, January 1997.
[30] C.O.E. Burg, A robust unstructured grid movement strategy using three-dimensional torsional springs, AIAA Paper 2004-2529, 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, Oregon, June 2004.
[31] C. Degand, C. Farhat, A three-dimensional torsional spring analogy method for unstructured dynamic meshes, Comput. Struct. 80 (2002) 305–316.
[32] C.L. Bottasso, D. Detomi, R. Serra, The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes, Comput. Methods Appl. Mech. Eng. 194 (2005) 4244–4264.
[33] P.J. Frey, H. Borouchaki, P. George, 3D Delaunay mesh generation coupled with an advancing-front approach, Comput. Methods Appl. Mech. Eng. 157 (1998) 115–131.
[34] A.G. Davies, B. Carol, Venkatapathy, Ethiraj, SAGE: The Self-Adaptive Grid Code, Version 3, NASA/TM-1999-208792 (1999).
[35] S.N. Atluri, T. Zhu, A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics, Comput. Mech. 22 (1998) 117–127.
[36] A.D. Heathershaw, Seabed-wave resonance and sand bar growth, Nature 296 (1982) 343–345.
[37] A.G. Davies, A.D. Heathershaw, Surface-wave propagation over sinusoidally varying topography, J. Fluid Mech. 144 (1984) 419–440.
[38] C.C. Mei, Resonant reflection of surface water waves by periodic sandbars, J. Fluid Mech. 152 (1985) 315–335.
[39] P.G. Chamberlain, D. Porter, The modified mild-slope equation, J. Fluid Mech. 291 (1995) 393–407.
[40] P. Porter, D. Porter, Scattered and free waves over periodic beds, J. Fluid Mech. 483 (2003) 129–163.
[41] Liu, Yuming, Yue, K.P. Dick, On generalized Bragg scattering of surface waves by bottom ripples, J. Fluid Mech. 356 (1998) 297–326.
[42] R. Eatock-Taylor, B.T. Wang, G.X. Wu, On the transient analysis of the wavemaker, in: Proceedings of the 9th International Workshop on Water Waves and Floating Bodies, Kuju, Oita, Japan, 1994.
[43] A. Nestegard, Status of Nonlinear hydrodynamic modelling, Technical Report, DNV, Norway, 1999.