INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **57**:437–452
Published online 19 September 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1602

Simulation of liquid sloshing in curved-wall containers with arbitrary Lagrangian–Eulerian method

H. Zhou*,†, J. F. Li and T. S. Wang

School of Aerospace, Tsinghua University, Beijing, China

SUMMARY

There are many challenges in the numerical simulation of liquid sloshing in horizontal cylinders and spherical containers using the finite element method of arbitrary Lagrangian–Eulerian (ALE) formulation: tracking the motion of the free surface with the contact points, defining the mesh velocity on the curved wall boundary and updating the computational mesh. In order to keep the contact points slipping along the curved side wall, the shape vector in each time advancement is defined to modify the kinematical boundary conditions on the free surface. A special function is introduced to automatically smooth the nodal velocities on the curved wall boundary based on the liquid nodal velocities. The elliptic partial differential equation with Dirichlet boundary conditions can directly rezone the inner nodal velocities in more than a single freedom. The incremental fractional step method is introduced to solve the finite element liquid equations. The numerical results that stemmed from the algorithm show good agreement with experimental phenomena, which demonstrates that the ALE method provides an efficient computing scheme in moving curved wall boundaries. This method can be extended to 3D cases by improving the technique to compute the shape vector. Copyright © 2007 John Wiley & Sons, Ltd.

Received 20 November 2006; Revised 13 July 2007; Accepted 15 July 2007

KEY WORDS: ALE method; curved wall boundary; free-surface flow; fractional step method; finite element method; renew mesh

1. INTRODUCTION

Spheroidal and horizontal cylindrical tanks are commonly installed in spacecraft to load fuel and liquid refrigerant. The motion of free surface has become the subject of much research in the design and attitude control of satellites and rockets. However, the main research results were based on the semi-analytical method which focused on evaluating the low-order natural frequencies and damping [1–3] by solving linear equations for fluid. In addition, the approaches cannot handle

*Correspondence to: H. Zhou, School of Aerospace, Tsinghua University, Beijing, China.
†E-mail: h-zhou04@mails.tsinghua.edu.cn

Contract/grant sponsor: Physical Science Fund

tanks with unusual geometric shapes, such as spheres and horizontal cylinders. The numerical simulation of free surface flows by the arbitrary Lagrangian–Eulerian method is suitable for these curved wall boundary conditions, but there are still several problems to be overcome. It is known that the arbitrary Lagrangian–Eulerian (ALE) algorithm can prescribe the motions of computational nodes in conformity with those of material points as the pure Lagrangian formulation does [4–6]. Actually, in order to avoid mesh distortion, the computational mesh's motion in the ALE algorithm cannot move as freely as one would wish. After rigidly restricting the computational mesh's motion within one direction, this strategy was shown to have a powerful capability in tracking free surfaces within straight boundary containers, such as sloshing in rectangular and vertical cylindrical tanks [4, 5].

However, when the free interface with liquid contact lines moves on a curved boundary, the nodal velocities at the contact lines are not straight along the direction of the reference axis. In order to correctly track the position of contact lines on curved walls, the prescribed mesh motion in the ALE algorithm has at least two freedoms, which differs from most numerical simulations that use the ALE method. Otherwise, the free surface and wall boundary are time-dependent moving boundaries. Hence, after predicting the nodal velocity and position of all boundaries, it is necessary to smooth the interior nodal velocity to avoid the acute aspect ratio of elements.

In order to enable the computational mesh to completely capture the physical interface of sloshing liquid in curved-wall containers, the nodes at the contact line must conform to not only the kinematic boundary conditions on the free surface but also the slipping boundary conditions on the tank's wall. Therefore, these nodes must move along the curved surface while freely tracking the free surface, and the shape vector is defined to increase the nodal motion's freedom to avoid the contradiction mentioned above. Concretely, the shape vector defined here modifies the kinematical conditions on the free surface based on the geometrical constraints of the tank. Thus, the values of the nodal velocities at the contact points on the free surface are altered. On the other hand, it is critical for the validity of the mesh to gain a set of velocity values with the appropriate gradient from the exterior moving nodes to the fixed interior nodes. Unfortunately, it is difficult to find a domain far from moving boundaries with fixed nodes embedded in it, including liquid free surfaces and wet wall boundaries. In this case, there is no nodal velocity component equal to zero as a reference baseline to describe the relative motion of the free surface in advance; hence, it is necessary to develop a specific method assigning the nodal values on the whole boundary. The nodal velocities of the free surface are determined by the modified kinematic boundary conditions on the free surface based on the flow field. The other exterior nodal velocities at the curved wall must keep a deft gradient according to those at the free surface, and be as similar to the real liquid velocity field as possible. As for the interior domain, we adopt the elliptic method [7, 8] to directly smooth the nodal velocity because it can naturally produce a uniform interior mesh velocity in an arbitrary direction without any other physical parameters. This method, which had been used to simulate the flows of broken dam [7], is mainly to resolve a Laplace equation with these exterior nodal velocity boundary conditions.

To solve the Navier–Stokes equations, the standard Galerkin formulation is applied to spatially discretize the ALE formulation governing equations, which has been successfully used by Glowinski and Pironneau [9] to solve incompressible flows. The fractional step method was introduced to split a time step into two parts. This is a time-split method to solve the time-dependent incompressible flow [10, 11].

In the following section, we describe the ALE algorithm and the corresponding expression about the kinematic boundary conditions. Section 3 describes how to select nodal velocities on boundaries

DOI: 10.1002/fld

such as the Dirichlet-type boundary conditions as well as the details about establishing the finite element formula of the remesh equation. The algorithm for tracking free surfaces is discussed in Section 3.1, where a figure illustrates the definition of the shape vector at the contact point, and also a linear function is included to determine other nodes on free surfaces. On the basis of the fractional step method, Section 4 establishes the ALE formula for finite element equations for incompressible viscous fluid flows. Also, the complete boundary conditions and initial conditions are described in this section. At the end of this paper, numerical results obtained using this method demonstrate the effectiveness of the simulation of liquid sloshing in partially filled horizontal cylinders and spherical containers.

2. DESCRIPTION OF THE ALE ALGORITHM AND THE KINEMATIC BOUNDARY CONDITIONS OF MESH

2.1. Kinematical description of the ALE algorithm

In the description of the ALE method [6], the computational domain is defined to combine the advantages of two classical kinematical formulations, the Lagrangian algorithm and the Eulerian algorithm. The mesh nodes on the reference domain could move with an arbitrary velocity appointed according to the material particle velocity. Assume a physical property f described by the ALE algorithm; its absolute derivative should be written as the following expression:

$$\frac{\mathrm{D}f}{\mathrm{D}t}=\left.\frac{\partial f}{\partial t}\right|_{\mathbf{w}}+(\mathbf{u}-\mathbf{w})\cdot\nabla f=0 \tag{1}$$

where $\mathbf{u}$ and $\mathbf{w}$ are the velocities of the material particle and the mesh, respectively, relative to the inertial reference system. In the Eulerian algorithm, the mesh velocity is equal to zero, whereas in the Lagrangian algorithm, $\mathbf{w}=\mathbf{u}$.

2.2. Description of kinematic boundary conditions of the ALE algorithm

Since the mesh velocity could avoid the computational mesh distortion in the ALE algorithm, how to choose the mesh velocity coinciding with the boundary conditions becomes the main problem.

First, the ALE algorithm should provide a concise method to track free surfaces. For the liquid sloshing, the most important kinematic constraint is that the fluid flux is zero on the free surface. The function $F(x,y,z,t)=0$ represents the position of each material particle on the free surface. In the ALE description, $\mathrm{D}F/\mathrm{D}t=0$ is expressed as

$$\frac{\mathrm{D}F}{\mathrm{D}t}=\left.\frac{\partial F}{\partial t}\right|_{\mathbf{w}}+(\mathbf{u}-\mathbf{w})\cdot\nabla F=0 \tag{2}$$

where $\partial F/\partial t|_{\mathbf{w}}=0$, owing to the fact that the material particles on the free surface always remain on the computational mesh's free surface during computation time. The equation, ∇F, is a vector denoting the free surface's normal vector $\mathbf{n}=(\partial F/\partial x,\partial F/\partial y)$ in 2D cases. Equation (2) can be simplified as

$$\mathbf{w}\cdot\mathbf{n}=\mathbf{u}\cdot\mathbf{n} \tag{3}$$

DOI: 10.1002/fld

where **n** is the unit normal vector of the free surface, **w** is the mesh velocity on free surface and **u** is the fluid velocity. Equation (3) indicates that the normal component of mesh velocity on the free surface is equal to that of fluid velocity, while it satisfies the boundary condition that the fluid flux is zero on the free surface.

Secondly, when large-scale displacement occurs on the free surface, the mesh velocity on the side wall should also be assigned according to the fluid boundary conditions. For the instance of liquid sloshing in horizontal cylinders and spherical containers, the slipping boundary conditions are expressed as

$$\frac{\partial \mathbf{u}}{\partial \boldsymbol{\tau}} = \bar{\mathbf{u}}, \quad \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0 \tag{4}$$

Therefore, the side-wall boundary condition of the mesh velocity **w** has the same modality as Equation (4):

$$\frac{\partial \mathbf{w}}{\partial \boldsymbol{\tau}} = \bar{\mathbf{w}}, \quad \frac{\partial \mathbf{w}}{\partial \mathbf{n}} = 0 \tag{5}$$

The value of $\bar{\mathbf{w}}$ should balance the contradiction between the mesh distortion and the displacement of the free surface.

3. MESH UPDATING

The procedure for mesh updating could be divided into two parts. The first is to prescribe the nodal velocities on the exterior interface, which should be subject to certain physical or geometrical constraints, including kinematical conditions on the free boundary, Equation (2), and nonpermeability conditions on the side wall, Equation (5). The remeshing strategy for straight-wall containers mostly updates the node positions only in a single direction along the straight wall. Examples of this strategy are $w_x|_{(\Gamma_t)_\mathrm{f}} = 0$ and $w_y|_{(\Gamma_t)_\mathrm{f}} = \bar{w}$. However, because the motion of the free surface is in curved boundary containers, our focus will be on a multiple-degree-of-freedom mesh renew strategy. The second is to choose an appropriate algorithm to smooth the velocities of the nodes in the interior domain. In the domain with straight boundaries, the algebraic method can smooth the velocity of each interior node by using an algebraic function. The distance between an interior node and the corresponding node on the moving free surface is the main variable in this function. In addition, the elastic mesh update method is also used to smooth velocity. It supposes the computation mesh domain as a feigned elastic solid and solves it as a deformation problem with the known distortion of boundary. However, as a robust mesh update algorithm, the performance parameters for each element of the feigned elastic mesh must be assigned to minimize the distortion of the computational mesh. Literature [12] provides different strategies and compares its applicabilities through a simple example. The elliptic generation method adopted here directly smoothes the velocities of the internal nodes based on the exterior boundary conditions deduced from the liquid's moving interface. To gain a relatively homogeneous mesh velocity field **w** minimizing the element distortion, the Laplace equation is introduced to solve the mesh velocity field **w** with Dirichlet boundary conditions [13].

DOI: 10.1002/fld

3.1. *Track free surface*

The free interface between the air and fluid should satisfy the kinematic condition Equation (2). It can be rewritten as

$$\mathbf{w}^n \cdot \mathbf{n}^n = \mathbf{u}^n \cdot \mathbf{n}^n \tag{6}$$

Now, because the fluid velocity $\mathbf{u}^n$ is known at any iteration process in the $n+1$ time advancement, the shape of the free surface is easily predicted by the nodal velocity on the free surface at time step n. Thus, we can determine the values of the mesh velocity $\mathbf{w}^n$ on the free surface by Equation (6). However, because Equation (6) is a scalar expression with at least two unknown components of the mesh velocity, we have to assign certain components as equal to zero. Thus, the freedoms of mesh velocity are restricted to rising or falling only in one direction. The method mentioned above loses its applicability when the free interface unavoidably moves along the curved side boundary with uniform mass transfer. Compared with the method reported by Marek Behr and Abraham [14] in 2002, which defines nodal displacement with a vector and a scalar value, the mesh velocity field $\mathbf{w}$ here is written in the following form:

$$\mathbf{w} = \phi \mathbf{e} \tag{7}$$

Thus, $\mathbf{w}$ is separated as an unknown scalar velocity and a known unit vector $\mathbf{e}$ called the shape vector, which corresponds to the geometrical constraints on the side surface.

Next, after substitution of $\mathbf{w}$ into Equation (6), ϕ gives an expression through free surface kinematic condition

$$\phi = \frac{\mathbf{u}^n \cdot \mathbf{n}^n}{\mathbf{e}^n \cdot \mathbf{n}^n} \tag{8}$$

During each iterative step, these nodal values, including $\mathbf{u}^n$, $\mathbf{e}^n$ and $\mathbf{n}^n$, are known in advance; hence, ϕ is easily determined by directly solving Equation (8), and obviously ϕ is time dependent.

In this paper, the shape vector $\mathbf{e}(e_x, e_y, e_z)$ at contact points should be consistent with the tangential direction of the side wall as shown in Figure 1. The components (e_x, e_y) of the shape vector denote the sine and cosine values of the included angle θ from the shape vector to the x-axis, respectively. And e_z is equal to zero in 2D cases because it stays on the Oxy plane. In the 3D cases, when we make the Oxy plane rotate around the y-axis forming a new coordinate system, each contact point will have a corresponding angle of rotation. In the new Oxy plane, we can gain the components (e_x, e_y) similar to that in 2D. Subsequently, the components of shape vector, (e_x, e_y, e_z) in old coordinate can be determined by the conversion of coordinates.

The positions of the contact points are determined by the fluid velocities at the last time step; hence, the θ on the contact points can be discovered through the function of tangent line to the curved side wall. Other shape vectors at free surface nodes can be determined by writing a function with the nodes at various distances from the contact points. As shown in Figure 2, the included angles at the two contact points are given as θ_1, and θ_2, and its x-components are x_1 and x_2. The following equation can be used to calculate the included angle θ relative to the x-axis of any nodes on the free surface:

$$\theta = kx + (\theta_1 - kx_1) \tag{9}$$

where

$$k = (\theta_1 - \theta_2)/(x_1 - x_2) \tag{10}$$

DOI: 10.1002/fld

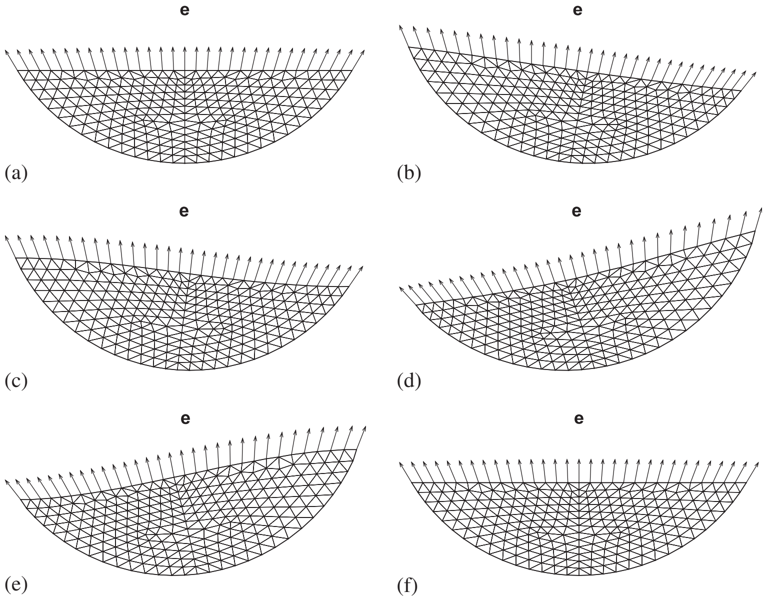

Figure 1. The shape vectors **e** illustrate at equal time increment ($2\pi/5$) during a period: (a) 0; (b) $2\pi/5$; (c) $4\pi/5$; (d) $6\pi/5$; (e) $8\pi/5$; and (f) 2π.

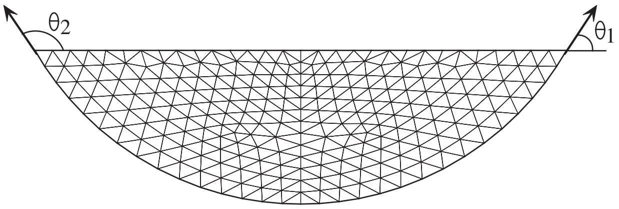

Figure 2. Definition of θ parameter.

The shape vector is $\mathbf{e}=(\cos\theta, \sin\theta)$, and the value of $\mathbf{w}$ in the x-coordinate is determined using Equation (7).

After determining the value of $\mathbf{w}^n$, the finite element computation mesh will be updated into a new domain. First, the position and the shape of the free surface will be changed as follows:

$$\mathbf{x}^{n+1}=\mathbf{x}^n+\mathbf{w}^n\cdot\Delta t \tag{11}$$

Then, the velocities of the nodes in the interior region will also be renewed by solving Laplace equation (16) based on the computational mesh at the last time step. Using Equation (11), the new positions of the internal nodes are established.

3.2. Mesh velocity on exterior interface

In order to figure out the free surface's motion relative to some reference position, generally appointing the minimum velocity as zero, it is necessary to define an area of the ALE computational mesh with the appropriate node velocity. However, in partially filled horizontal cylinders and spherical containers, it is difficult to appoint a certain section of a wet wall as the reference area with zero mesh velocity, especially when the slipping boundary conditions are enforced on the whole wet wall. Fortunately, the motion of the free surface is a reliable guide to attain the nodal motion on a wet wall. Thus, given the mesh velocity field, the value of $\mathbf{w}$ on the moving free surface is

$$\mathbf{w}|_{(\Gamma_t)_\mathrm{f}} = \bar{\mathbf{w}}_\mathrm{f} \tag{12}$$

Note that $\bar{\mathbf{w}}_\mathrm{f}$ is not the real velocity of the fluid material point on the free surface, and the details for calculating the nodal value of $\bar{\mathbf{w}}_\mathrm{f}$ on the free surface are discussed in Section 3.1. Otherwise, considering the mesh distortion, the prescribed mesh velocity $\mathbf{w}$ on a wet wall would also be variable in every time step. There are two factors influencing the value of $\bar{\mathbf{w}}_\mathrm{w}$: one is the distances between the free surface and nodes on the wet wall, and the other is the nodal velocity on the free surface. Sometimes the optimal strategy is to prescribe the real fluid velocity on the wet wall as the mesh velocity $\bar{\mathbf{w}}_\mathrm{w}$. However, it is not a feasible method to adjust the contradiction between the tracking surface and elements' distortion. In the following equation,

$$\mathbf{w}|_{(\Gamma_t)_\mathrm{w}} = \bar{\mathbf{w}}_\mathrm{w} \tag{13}$$

the elements along the curved wall boundary are subject to two kinds of distortion, pulling or pressing, depending on the value of $\bar{\mathbf{w}}_\mathrm{w}$. The obvious distortion mainly results from the wide variation in velocity; hence, the $\bar{\mathbf{w}}_\mathrm{w}$ can be written as the following formula to keep the gradient of velocity in the proper range:

$$\bar{\mathbf{w}}_\mathrm{w} = \mathbf{u}_\mathrm{w} f(h) \tag{14}$$

Here, $f(h)$ is a function assuming the value of unity for $h=0$, and as h increases to the maximal distance from the liquid free surface, $f(h)$ decreases to zero. If the element distortion is too little to effect the characteristics of the element, the mesh velocity $\bar{\mathbf{w}}_\mathrm{w}$ equals $\mathbf{u}_\mathrm{w}$ on the wall boundary. Summing up, $\bar{\mathbf{w}}u_\mathrm{w}$ can be expressed as

$$\bar{\mathbf{w}}_\mathrm{w} = \begin{cases} \mathbf{u}_\mathrm{w} \\ \mathbf{u}_\mathrm{w} f(h) \end{cases} \tag{15}$$

3.3. Mesh velocity in the interior region

Given the nodal velocity on the fluid boundary interface, the elliptic method may calculate the mesh velocity of internal domain by solving

$$\Delta \mathbf{w} = \mathbf{0} \quad \text{in } \Omega_t \tag{16}$$

with the following Dirichlet boundary conditions:

$$\begin{aligned} \mathbf{w}|_{(\Gamma_t)_\mathrm{f}} &= \bar{\mathbf{w}} \quad \text{on } (\Gamma_t)_\mathrm{f} \\ \mathbf{w}|_{(\Gamma_t)_\mathrm{w}} &= \bar{\mathbf{w}}_\mathrm{w} \quad \text{on } (\Gamma_t)_\mathrm{w} \end{aligned} \tag{17}$$

Using the principle of the standard Galerkin method to spatially discretize Equation (16), the weight function has the same form as the shape function. The interpolation spaces are defined here as

$$\begin{aligned} s^h &= \{\varphi^h | \varphi^h \in H^{1h}(\Omega_t),\ \varphi^h = \overline{\varphi} \text{ on } (\Gamma_t)_{\mathrm{f}}\} \\ v^h &= \{\varphi^h | \varphi^h \in H^{1h}(\Omega_t),\ \varphi^h = 0 \text{ on } (\Gamma_t)\} \end{aligned} \tag{18}$$

and the Laplace equation, Equation (16), can be transformed into the finite element equation by employing the Green–Gauss divergence theorem:

$$\begin{aligned} \int_\Omega \varphi \Delta \mathbf{w}\, \mathrm{d}\Omega &= \int_\Omega \nabla\varphi \cdot \nabla \mathbf{w}\, \mathrm{d}\Omega - \int_{\Gamma_t} \varphi(\mathbf{n} \cdot \nabla \mathbf{w}) \\ &= \int_\Omega \nabla\varphi \cdot \nabla \mathbf{w}\, \mathrm{d}\Omega = 0 \end{aligned} \tag{19}$$

In a matrix equation of the following form, the mesh velocity will be calculated with the boundary conditions as Equation (17)

$$\mathbf{A}\mathbf{w} = 0 \tag{20}$$

Obviously, due to the physical feature of the Laplace operator that can average the physical quantity on the boundary into the interior domain, the elliptic method can use the straightforward procedures to smooth the interior velocity [7]. Rainald Lohner [8] added variable diffusivity to the Laplace operator to gain more uniform mesh velocity in the region close to the moving boundary with smaller elements.

4. GOVERNING EQUATIONS FOR VISCOUS INCOMPRESSIBLE FLUID WITH FREE SURFACE

Assuming a rigid container partially filled with viscous incompressible fluid, we adopted the unsteady Navier–Stokes velocity–pressure formula to describe the free surface motion in the abounded region Ω_t with boundary Γ_t. The spatial domain of liquid is time dependent due to the large-scale displacement of the free surface, and the subscript t indicates such time dependence. Derived from the ALE method [6, 15] the Navier–Stokes equations are written as

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathrm{w}} + (\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + v\nabla^2 \mathbf{u} + \mathbf{f} \tag{21}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0} \tag{22}$$

Here, $\mathbf{u}$ is the fluid velocity field with components $\mathbf{u} = (u_x(x, y, t), u_y(x, y, t))$, and $p = p(x, y, t)$ is the dynamical pressure field. These two variables are unknown in these equations. $\mathbf{f}=(f_1, f_2)$ is the body force per unit fluid mass, including inertia force and environmental gravity. $\mathbf{w}=(w_x(x, y, t), w_y(x, y, t))$ is the nodal velocity field relative to the inertial reference frame, while $|_{\mathrm{w}}$ denotes the derivative on the reference frame consisting of arbitrarily moving mesh. The v and ρ are two fluid parameters, respectively, the kinematic viscosity and the density.

The boundary conditions on the issue are organized for the free surface $(\Gamma_t)_{\mathrm{f}}$ and the wall boundaries $(\Gamma_t)_{\mathrm{w}}$, which are complementary subsets of the boundary (Γ_t). With the assumption that the momentum on the air side and the surface tension are equal to zero, the dynamic constraint of incompressible Newtonian fluid on the free surface $(\Gamma_t)_{\mathrm{f}}$ is

$$-pn_i + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)n_j = 0 \tag{23}$$

From this point, the normal component of the free surface can be written as

$$-pn_in_i + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)n_jn_i = 0 \tag{24}$$

This condition guarantees the equilibrium between the forces of external and internal stresses on the free surface. It will be introduced into Equation (21) during the time integral procedure. The tangential component of Equation (23) is written as the following formula:

$$\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)n_js_i = 0 \tag{25}$$

where s_i is the unit tangent vector of the free surface. This is a Newmann-type velocity boundary condition enforced on the free surface.

In the ALE algorithm, the nodes on the free surface must be confined to the free surface during the whole calculation process; hence, the node scattering wall boundary slips along the wall with the motion of the free surface. The fluid velocity on the wall boundaries $(\Gamma_t)_{\mathrm{w}}$ is subject to slipping boundary conditions:

$$u_i \cdot n_i = 0 \tag{26}$$

Thus, the stress tangential component on the $(\Gamma_t)_{\mathrm{w}}$ is neglected naturally.

Moreover, the normal component of stress on the $(\Gamma_t)_{\mathrm{w}}$ is subject to the time-discretized Navier–Stokes equation shown in Equation (21). With the additional velocity boundary conditions on the side walls, including Equation (26), and the incompressibility, $\nabla \cdot \mathbf{u} = \mathbf{0}$, the boundary condition is simplified as

$$p_{,i}^{n+1}n_i = \rho f_i^{n+1}n_i \tag{27}$$

The initial conditions in the problem are $\mathbf{u}_0 = \mathbf{u}(x, y, 0)$ and $p_0 = p(x, y, 0)$. Note that the divergence of the initial velocity field should be equal to zero, $\nabla \cdot \mathbf{u}_0 = \mathbf{0}$, and p_0 is the hydrostatic pressure.

4.1. Fraction-step method

The fraction-step method is known as a popular time discretization method to solve Navier–Stokes equations, which can decompose a one-time process into two or more sub-steps by splitting the complicated numerical treatment of various operators into several simplified sub-steps. This permits the equal-order basic function to be applied to $\mathbf{u}$ and p in spatial discretization with the appropriate pressure numerical error.

On the basis of the principle of the Chorin–Temam projection method, the fraction-step method used here can separately determine pressure and velocity by evaluating an additional variable named inter-medial velocity at one time advancement. The Ladyzhenskaya theorem [11] states that any vector field $\boldsymbol{\omega}$ in Ω admits a unique orthogonal decomposition

$$\boldsymbol{\omega} = \boldsymbol{\upsilon} + \nabla\psi \tag{28}$$

where $\boldsymbol{\upsilon}$ is a field with zero normal components on the boundary Γ of Ω. Therefore, the use of the orthogonal projection operator P eliminates the pressure variable from the Navier–Stokes equation without a source term. The partial differential of $\mathbf{u}$ can be written as

$$\frac{\partial \mathbf{u}}{\partial t} = P[-(\mathbf{u}\cdot\nabla)\mathbf{u} + v\nabla^2\mathbf{u}] \tag{29}$$

and $\boldsymbol{\upsilon}$ is still subject to the initial boundary conditions on normal components, which is an important characteristic for finite element Galerkin approximations.

On the basis of this theorem, in the Chorin–Temam projection method the pressure in the first sub-step is directly omitted and the total pressure is modified. Here, in order to gain improved convergence properties, we adopted an incremental projection method [6], which is embodied as

the first sub-step:

$$\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla\mathbf{u}^n = -\frac{1}{\rho}\nabla p^n + v\nabla^2\mathbf{u}^n + \mathbf{f}^{n+1} \quad \text{in } \Omega \tag{30}$$

$$\tilde{\mathbf{u}}^{n+1} = \bar{\mathbf{u}}^{n+1} \quad \text{on } \Gamma \tag{31}$$

the second sub-step:

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1} + \frac{1}{\rho}\nabla p^n \quad \text{in } \Omega \tag{32}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0 \quad \text{in } \Omega \tag{33}$$

$$\mathbf{n}\cdot\mathbf{u}^{n+1} = \mathbf{n}\cdot\bar{\mathbf{u}}^{n+1} \quad \text{on } \Gamma \tag{34}$$

Obviously, this method can be adopted to determine the inter-medial velocity here by the explicit Euler method or the semi-implicit method in the first sub-step; in the second sub-step the value of the pressure should be calculated by establishing the pressure-Poisson equation due to enforcing the incompressible constraints in Equation (32). The final step, virtually included in the second sub-step, is to modify the inter-medial velocity field into the actual velocity field using the pressure value from the previous computation sub-step. The ALE formulation governing equations are discretized as the following three sub-steps with component equations:

$$\frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^n - \mathbf{w}^n)\cdot\nabla\mathbf{u}^n = -\frac{1}{\rho}\nabla p^n + v\nabla^2\mathbf{u}^n + \mathbf{f}^{n+1} \tag{35}$$

$$\Delta p^{n+1} = \frac{\rho}{\Delta t}\nabla\cdot\tilde{\mathbf{u}}^{n+1} + \Delta p^n \tag{36}$$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - \frac{\Delta t}{\rho}\nabla(p^{n+1} - p^n) \tag{37}$$

4.2. Finite element formulation for fluid equations

Now, we can easily obtain the finite element approximation of the weak formulation problems (35)–(37), by the standard Galerkin method combined with the boundary conditions (24), (25), (27) and the initial conditions (26). The finite element discretization of the domain Ω is acquired by separating it into elements. The interpolation space of the finite element for the velocity and pressure is defined.

With the weight function $W^*=(\mathbf{u}^*,\mathbf{p}^*)$, the fraction-step finite element equation for liquid equations are obtained as the following formula:

$$\int_\Omega\left(\frac{\tilde{\mathbf{u}}^{n+1}-\mathbf{u}^n}{\Delta t}+(\mathbf{w}^n-\mathbf{u}^n)\cdot\nabla\mathbf{u}^n\right)\mathbf{u}^*\,\mathrm{d}V=\int_\Omega\left(-\frac{1}{\rho}\nabla p^n+\nu\nabla^2\mathbf{u}^n\right)\mathbf{u}^*\,\mathrm{d}V+\int_\Omega\mathbf{f}^{n+1}\mathbf{u}^*\,\mathrm{d}V \tag{38}$$

$$\int_\Gamma\nabla p\cdot\mathbf{n}p^*\,\mathrm{d}S-\int_\Omega\nabla p\nabla p^*\,\mathrm{d}V=\frac{\rho}{\Delta t}\int_\Omega\nabla\cdot\tilde{\mathbf{u}}^{n+1}p^*\,\mathrm{d}V+\int_\Gamma\nabla p\cdot\mathbf{n}p^*\,\mathrm{d}S-\int_\Omega\nabla p\nabla p^*\,\mathrm{d}V \tag{39}$$

$$\int_\Omega\mathbf{u}^*\mathbf{u}^{n+1}\,\mathrm{d}V=\int_\Omega\mathbf{u}^*\tilde{\mathbf{u}}^{n+1}\,\mathrm{d}V-\frac{\Delta t}{\rho}\left(\int_\Omega\mathbf{u}^*\nabla p^{n+1}\,\mathrm{d}V-\int_\Omega\mathbf{u}^*\nabla p^n\,\mathrm{d}V\right) \tag{40}$$

The first expression on the right side of Equation (38) can be transformed into two integral formulae

$$\begin{aligned}&\int_\Omega\left(-\frac{1}{\rho}\nabla p^n+\nu\nabla^2\mathbf{u}^n\right)\mathbf{u}^*\,\mathrm{d}V\\&\quad=\int_\Gamma\left[-\frac{1}{\rho}p^n n_i^n+\nu\left(\frac{\partial u_i^n}{\partial x_j}+\frac{\partial u_j^n}{\partial x_i}\right)n_j^n\right]u_i^*\,\mathrm{d}S-\int_\Omega\left[-\frac{1}{\rho}p^n+\nu\left(\frac{\partial u_i^n}{\partial x_j}+\frac{\partial u_j^n}{\partial x_i}\right)\right]u_{i,j}^*\,\mathrm{d}V\end{aligned} \tag{41}$$

After introducing the boundary conditions, including Equations (27) and (24), into the finite element equations (38)–(40), we obtain the integrated finite element equations:

$$\mathbf{M}^{n+1}\tilde{\mathbf{U}}^{n+1}=\mathbf{M}^{n+1}\mathbf{U}^n-\Delta t\left[\mathbf{B}^n\mathbf{U}^n-\frac{1}{\rho}\mathbf{C}^n\mathbf{P}^n+\mathbf{D}^n\mathbf{U}^n-\mathbf{F}^{n+1}\right] \tag{42}$$

$$\mathbf{A}^{n+1}\mathbf{P}^{n+1}=-\frac{\Delta t}{\rho}\mathbf{C}^{n+1}\tilde{\mathbf{U}}^{n+1}+\mathbf{A}^n\mathbf{P}^n+\mathbf{Q}^{n+1}-\mathbf{Q}^n \tag{43}$$

$$\mathbf{M}^{n+1}\mathbf{U}^{n+1}=\mathbf{M}^{n+1}\tilde{\mathbf{U}}^{n+1}-\frac{\Delta t}{\rho}[\mathbf{C}^{n+1}\mathbf{P}^{n+1}-\mathbf{C}^n\mathbf{P}^n] \tag{44}$$

In these equations, **U** is the vector of the nodal values of the fluid velocity, **P** is the vector of the element values of the fluid dynamical pressure, **M** is the mass matrix, **B** is the matrix determining the convection operator, **C** is the matrix denoting the gradient and divergence operators, **D** is the dissipation operator, and **A** is the matrix representing the Laplace operator. Finally, there are two important vectors: **F**, the vector of the nodal values of body force, and **Q**, the vector denoting the normal component of the body force enforced on the side wall boundary, which are derived from the pressure wall boundary conditions, Equation (27).

DOI: 10.1002/fld

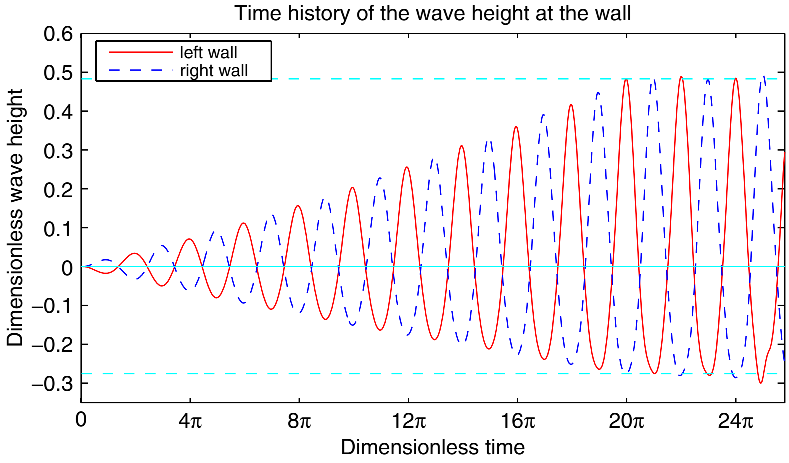

Figure 3. Time history of wave height at the wall.

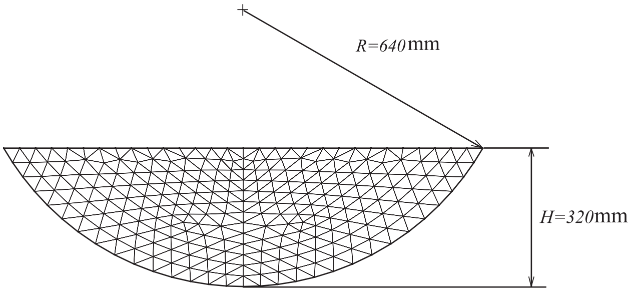

Figure 4. Initial computational mesh.

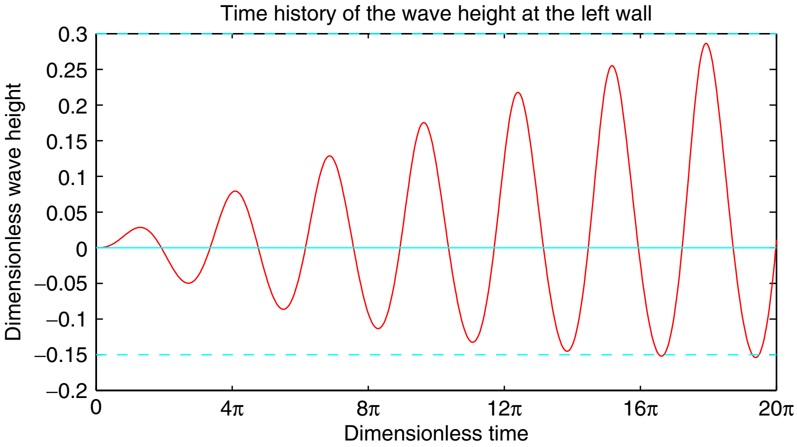

Figure 5. Time history of vibration height at the left corner node.

5. NUMERICAL EXAMPLE

To demonstrate the validity of the proposed method for large amplitude sloshing, the method was applied to simulate the motion of free surface in a rectangular tank. The width of the tank was 0.8 m and the depth of the liquid was 0.3 m. The liquid in the tank was subject to sinusoidal horizontal

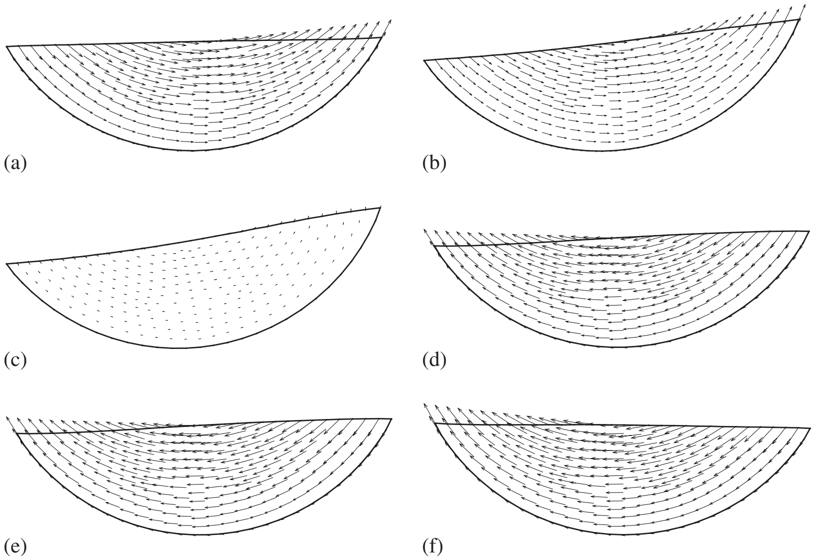

Figure 6. Liquid velocity fields at equal time increment during a half period: (a) $5\pi/2$; (b) $27\pi/10$; (c) $29\pi/10$; (d) $31\pi/10$; (e) $33\pi/10$; and (f) $7\pi/2$.

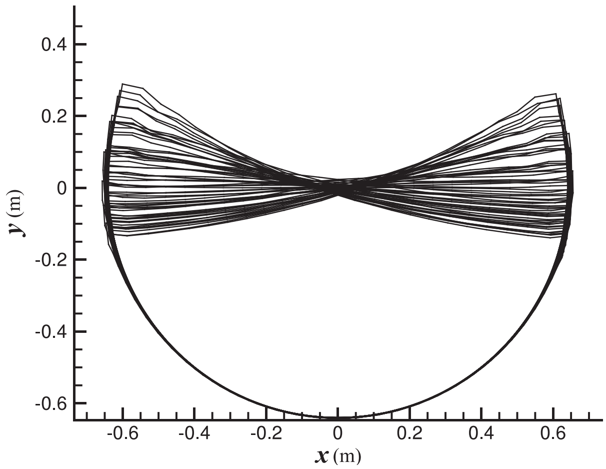

Figure 7. Free surface shapes during all calculational procedures.

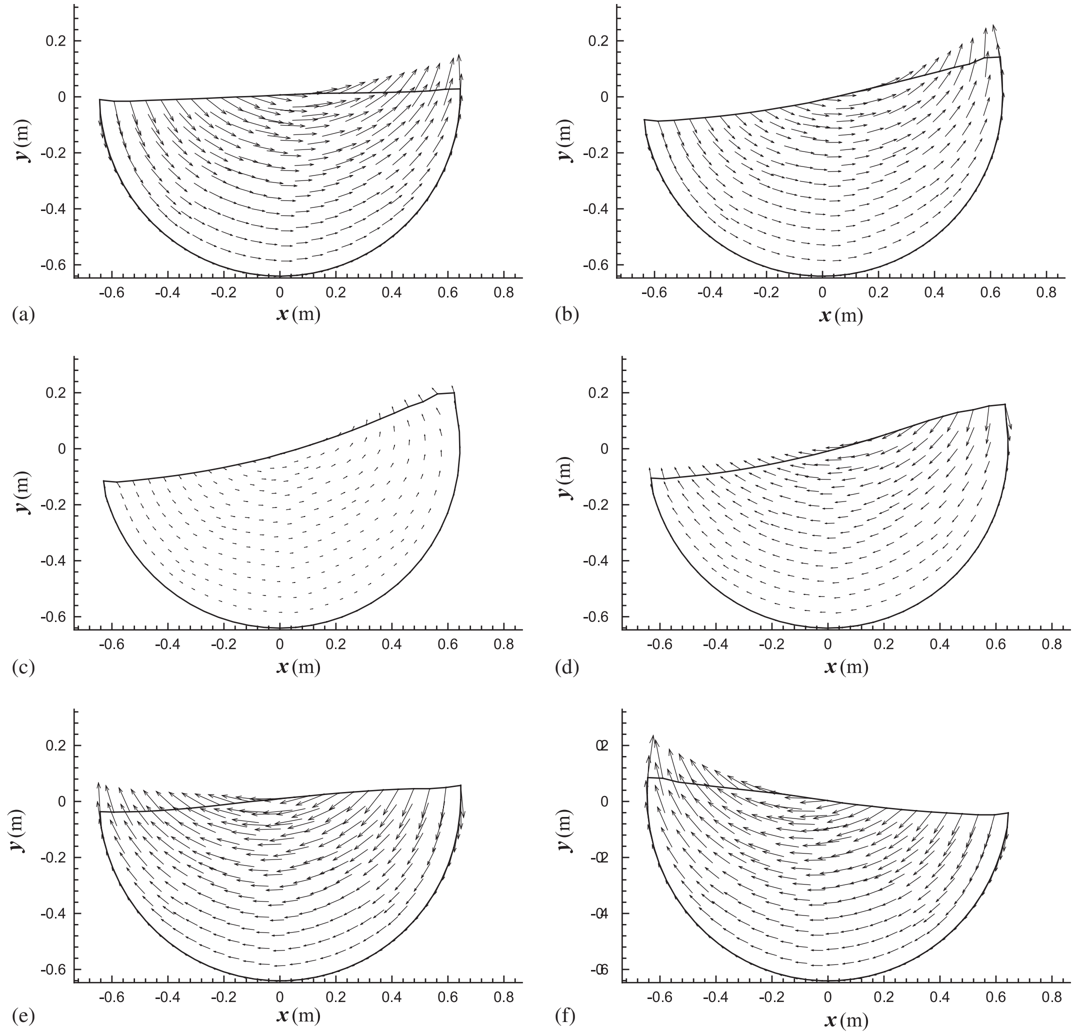

Figure 8. Liquid velocity fields at equal time increment during a half period: (a) $5\pi/2$; (b) $27\pi/10$; (c) $29\pi/10$; (d) $31\pi/10$; (e) $33\pi/10$; and (f) $7\pi/2$.

acceleration in a normal gravity environment as given in Equation (45). This is a classical example of the numerical simulation of sloshing, which has been studied by Hurta and Liu [16] and Souli and Zolesio [5].

$$f_1 = A \cdot g \cdot \sin(\omega t) \tag{45}$$

where A is an arbitrary constant amplitude of the horizontal acceleration, g is the environmental gravity, ω is the excitation frequency and t is the time. In the example, $A = 0.01$, $g = 9.8\,\mathrm{m/s^2}$ and $\omega = 5.642\,\mathrm{rad/s}$ when the approximate resonance frequency was 0.898 Hz: the value of resonance

DOI: 10.1002/fld

frequency was obtained from the experimental data in Reference [1]. In addition, the time increment $\Delta t = T/60$, where T is the approximate resonance period, $T = 2\pi/\omega$.

The results exhibited in Figure 3 are in good agreement with those reported in [5, 16].

The purpose of this algorithm is to simulate liquid sloshing in containers with curved side walls using the ALE finite element method. The following two examples are 2D liquid sloshing in spherical containers with different liquid depths. In the first example, a container was filled with fluid at a depth of 25% of the horizontal cylindrical container's diameter, and the initial domain dimensions are shown in Figure 4. For the second example, the liquid depth was 50% of the diameter of the same horizontal cylindrical container. This tested whether the algorithm could accommodate different geometrical constraints with motions of the free surface.

The excitation acceleration with the same formula as Equation (45) gives $A = 0.02$, $g = 9.8\,\mathrm{m/s^2}$ and $\omega = 4.3\,\mathrm{rad/s}$. The finite element mesh consists of 434 triangular elements. The top boundary is a fluid free surface, and the other boundary is the side wall of the container, where the fluid is subject to slipping boundary conditions in terms of geometrical constraints. The history of the elevational displacement at the corner nodes of the free surface is shown in Figure 5. When the wave height is beyond 15% of the free surface's width, the computational mesh can maintain good aspect ratio to guarantee the accuracy of simulation results. The whole velocity field in a half period is shown in Figure 6.

In the second example, sloshing in a half-filled container was simulated. The results confirmed that the algorithm is capable of dealing with different liquid-filled ratios. The calculation parameters were $A = 0.02$ and $\omega = 4.6\,\mathrm{rad/s}$, and the others were the same as the first example. The number of computational elements is 456. The free surface shape during the simulation is illustrated in Figure 7, which shows the deformation of the free surface. Obviously, the simulation takes on the important characteristic of liquid sloshing: the increasing height of the free surface is larger than the decreasing displacement.

The velocity field in a half period is exhibited in Figure 8. Although the smoothness of the free surface did not reach the ideal status, the mass conserved ratio was within 2.5% of the original fluid mass without extra technique to refine the grid.

6. CONCLUSION

In conclusion, the numerical algorithm succeeded in simulating liquid sloshing in containers with curved side wall using the ALE finite element method. In order to overcome the contradiction between the kinematic boundary conditions on the free surface and the geometrical constraints, the shape vector modified the kinematical boundary conditions on the free surface. For the ALE formulation, this is a concise method to renew the nodal velocity with more than one freedom on the moving boundary, including the free surface and the curved side wall. The numerical results in Section 5 demonstrated that the algorithm could be adapted to 2D simulation. The mesh distortion could be prevented only if the maximum wave height were not beyond 25% of the character length of the container during the whole calculation procedure. To simulate more violent sloshing within the 3D tanks, the elements near the moving boundary should be smaller than before, and the elliptic equation used in updating the nodal velocities of the interior domain should be optimized, which is the key point of the advanced study in this field.

DOI: 10.1002/fld

ACKNOWLEDGEMENTS

The study was financially supported by the Physical Science Fund. The author also wishes to express appreciation for the constructive suggestions and helpful discussions from Mr Li, Mr Wang and our project team.

REFERENCES

1. Abramson HN, Chu WH, Garza LR. Liquid sloshing in spherical tanks. *AIAA Journal* 1963; **1**(2):384–389.
2. Papaspyrou S, Karamanos SA, Valougeorgis D. Response of half-full horizontal cylinders under transverse excitation. *Journal of Fluids and Structures* 2004; **19**:985–1003.
3. Papaspyrou S, Valougeorgis D, Karamanos SA. Refined solutions of externally induced sloshing in half-full spherical containers. *Journal of Engineering Mechanics* (ASCE) 2003; **129**(12):1369–1379.
4. Amsden AA, Hirt CW, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**:227–253.
5. 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.
6. Donea J, Huerta A. *Finite Element Methods for Flow Problems* (1st edn), vol. 1. Wiley: Chichester, 2003.
7. Nithiarasu P. An arbitrary Lagrangian–Eulerian (ALE) formulation for free surface flows using the characteristic-based split (CBS) scheme. *International Journal for Numerical Methods in Fluids* 2005; **48**(12):1415–1428.
8. Lohner R, Yang C. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering* 1996; **12**:599–608.
9. Glowinski R, Pironneau O. Finite element methods for Navier–Stokes equations. *Annual Review of Fluid Mechanics* 1992; **24**:167–204.
10. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
11. Quartapelle L. *Numerical Solution of the Incompressible Navier–Stokes Equations*, ISNM vol. 113. Birkhäuser: Basel, 1993.
12. Chiandussi G, Bugeda G, Onate E. A simple method for automatic update of finite element meshes. *Communications in Numerical Methods in Engineering* 2000; **16**:1–19.
13. Robertson I, Sherwin SJ, Graham JMR. Comparison of wall boundary conditions for numerical viscous free surface flow simulation. *Journal of Fluids and Structures* 2004; **19**:525–542.
14. Behr M, Abraham F. Free-surface flow simulations in the presence of inclined walls. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:5467–5483.
15. Benson DJ. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Computer Methods in Applied Mechanics and Engineering* 1989; **72**(3):305–350.
16. Hurta A, Liu WK. Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering* 1988; **69**:277–324.

DOI: 10.1002/fld