Arbitrary Lagrangian–Eulerian method for Navier–Stokes equations with moving boundaries

Fabián Duarte [a], Raúl Gormaz [a,*], Srinivasan Natesan [b]

[a] *Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Casilla 170/3, Correo 3, Santiago, Chile*

[b] *Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati 781039, India*

Received 31 March 2003; received in revised form 1 December 2003; accepted 8 May 2004

Abstract

This article deals with the numerical approximation of Navier–Stokes equations in a domain with moving boundaries. Using the arbitrary Lagrangian–Eulerian (ALE) method we transform the problem from a moving domain to a fixed reference domain through an artificial domain velocity. Then, we apply the characteristic method to solve the Navier–Stokes equation in the fixed domain. Suitable boundary conditions are used for the interfaces. Three examples are provided to show the efficiency of the present method.

Keywords: Free boundary problems; ALE method; Method of characteristics; Finite element method

1. Introduction

Numerical simulation of two-dimensional viscous incompressible fluid with free boundaries is getting more attention for the past few decades; these types of problems arise frequently in several important industrial applications, such as melting and solidification, crystal growth, glass and metal forming processes, etc. In particular, modeling of mould filling during the casting of metals into moulds is an example for the present situation where the domain of interest has an unknown boundary in the beginning of the analysis.

Applications of the finite element method (FEM) to hydro-structural problems have been studied by Belytschko and Kennedy [1], and Donea et al. [4]. In these studies, a purely Lagrangian method was employed for the kinematical description of the fluid domain. The Lagrangian description, that is, the association of the state variables to the fluid particles is very convenient when the geometry of the domain

[*] Corresponding author.

E-mail address: rgormaz@dim.uchile.cl (R. Gormaz).

0045-7825/

changes. One can write the equations of motion with respect to a reference set which is in a bijective correspondence with the fluid particles. More precisely, in this formulation, the coordinate system moves with the fluid. The Lagrangian method has several useful properties: (i) material interfaces can be specifically delineated and precisely followed; (ii) free surface boundary conditions are easily applied; and (iii) curved rigid boundaries of any arbitrary shape can be treated. Whereas the main drawback of this method is that it will face severe problems to deal with strong distortions in the computational domain.

On the other hand, in the Eulerian formulation, the coordinate system is stationary, or moving in a certain prescribed manner in order to take into account the continuously changing solution domain. The grid movement is thus independent of the fluid particle movement. This results in the method being unable to deal easily with fluids that undergo large distortions at the interface. The main disadvantages of the Euler method are: (i) material interfaces lose their sharp definitions as the fluid moves through the mesh, so that basic Eulerian calculations requires special logic for interfaces, which is very complicated and often leads to inaccuracies; (ii) local regions of fine resolution are difficult to achieve. Several authors have used the Eulerian approach for free boundary flows, to list a few, Hirt and Nichols [9] and Frederiksen and Watts [7]. For more details about these approaches one may refer the articles by Donea et al. [5] and Navti et al. [16].

An intermediate approach is the arbitrary Lagrangian–Eulerian (ALE) method, which will provide a hybrid description not associated with the fluid particles and the laboratory coordinates. We associate the description with a moving imaginary mesh which follows the fluid domain. Let us denote the velocity of the domain by v. In the Eulerian approach this velocity is zero, whereas it is equal to the velocity of the fluid particles in the Lagrangian approach. But in the ALE method, v is equal to neither zero nor the velocity of the fluid particles, it varies smoothly and arbitrarily between both of them. This arbitrary mesh velocity keeps the movement of the meshes under control according to the physical problem, and it depends on the numerical simulation. More precisely, this method seems to be the Lagrangian description in zones and directions where 'small' motion takes place, and the Eulerian description in zones and directions where it would not be possible for the mesh to follow the motion of the fluid. The important merits of the method depends on the following: (i) it helps to keep the free surface, interfaces and solid boundaries with its suitable boundary conditions by moving the boundary and interior nodes; therefore, there is no need for re-meshing at every time step. A suitable choice of the domain velocity will minimize the re-meshing process; (ii) one can avoid the projection errors, that is, in the re-meshing strategy, one has to re-mesh the domain for each time step and project the numerical solution from the old mesh to the new one, at this juncture one encounter the so-called projection error, in general, several unstable schemes suffer a lot from the projection errors; and (iii) one can save some CPU time by avoiding the re-meshing criterion.

In [15], Maury has introduced the characteristic ALE method for the unsteady 3D free surface flows. Actually, he applied three times the characteristic method, (i) to define the ALE quantities and to establish the first-order ALE formulation of the Navier–Stokes equations, (ii) to move the free surface, and (iii) to take into account the convection term in the momentum equation. Further, he included a surface tension term on the free surface, and presented a stability estimate for a general scheme. For the numerical applications, he treated the impinging of a Jet on a plane, that is, falling of a fluid on a horizontal plane.

Our principal goal in this paper is to provide the underlying idea of the ALE method in a precise way by explaining the fundamentals in the 2D case and apply to a wide variety of free boundary problems to check the accuracy of the method comparing its results with other existing methods. We assume that the domain has a fixed topology and no interior boundary curve can appear or disappear during the simulation. By introducing the domain velocity v, we transform the problem from the moving domain to a fixed reference domain with suitable change of variables. The numerical solution of the problem is splitted into two steps.

We solve first a fixed domain problem, after applying a suitable change of variable. Then, we modify the domain according to v and transport the solution to the newly modified domain. The transformed equation in the fixed domain can be solved by various ways. Here, we apply the characteristic method introduced in [17].

Pironneau [17] presented an algorithm for the numerical solution of diffusion or convection dominated equations which is a combination of the method of characteristics with the finite element method and applied it to the Navier–Stokes equations. In general, this method looks like one does one step of transport plus one step of diffusion (or projection) but the mathematics show that it is also an implicit time discretization of the partial differential equation (PDE) in Lagrangian form. To solve the Navier–Stokes equation in the fixed domain, we apply the characteristic method which transforms the non-linear Navier–Stokes equation to a linear Stokes-like equation. Finite element method is used to solve the Stokes-like equation. In particular, we use the piecewise linear finite element functions ($\mathscr{P}_1$) on triangles for the pressure and $\mathscr{P}_1$-Iso$\mathscr{P}_2$ functions for the velocity. Again using the change of variables, we transform the solution from the reference domain to the original moving domain.

To show the efficiency and accuracy, the present method is applied to three model problems, namely the propagation of a solitary wave, broken dam problem and movement of an air bubble in a fluid. The first problem was studied by several researchers, which include Hansbo [8], Laitone [12], Navti et al. [16], and Ramaswamy and Kawahara [18]. By a proper choice of the mesh velocity the ALE method is able to follow the wave motion between the left and right vertical walls for an arbitrary time, without having distortion of meshes or triangles, and produces good approximations for the run-up height and maximum pressure with the earlier results. For the broken dam problem, the right side wall and the upper surface are free boundaries, the outputs produced by the present method coincides with the earlier computational and experimental results. Further, these two examples are solved by the computational fluid dynamics software Fluent for further comparison. In the third example, we are simulating the movement of an air bubble in a viscous incompressible fluid. Here, the bubble boundary and the upper surface are free surfaces, the ALE method preserves the interface conditions, and move the bubble for a while without having any problem in the meshes. For each example, the simulation continues as long as there is no need of re-meshing and stops when the quality of the triangles is poor.

From the simulations presented in Section 3, one can understand the merits of the present method, like, applicability, efficiency and accuracy. Further, the ALE method preserves the interface conditions and avoids partially the need of re-meshing the domain for each time step. If the quality of the triangles become poor, then one can re-mesh the domain before proceeding the simulation. By this infrequent re-meshing, we hope that the mesh distortions may be held down enough to permit the problem to run satisfactorily for a long time. This helps us to save CPU time and avoid the projection errors. The present method can be applied to a wide range of problems arising in engineering and applied areas, by this way it is clear that the ALE method will be generalized.

Before concluding the introduction, we cite some of the earlier works in this area. In [8], Hansbo introduced the characteristic streamline diffusion method for the time dependent incompressible Navier–Stokes equations. The method is based on space–time elements, discontinuous in time and continuous in space, which is closely related to the ALE method. Hughes et al. [10] used the Lagrangian–Eulerian finite element method for free incompressible viscous flows. Süli provided optimal error estimates for the Lagrange–Galerkin mixed finite element approximation of the Navier–Stokes equations in a velocity/pressure formulation in [19]. Douglas and Russell [6] presented a numerical method for convection–dominated diffusion problems which is a combination of characteristics with finite difference and finite element methods.

The rest of the paper is organized as follows. The continuous model, and its corresponding numerical method are described in Section 2. To show the applicability of the present method some numerical simulations are provided in Section 3. Section 4 presents the conclusions.

2. Mathematical description of the method

In this section, we will give the underlying idea of the present method together with a detailed description of the continuous model and then the numerical formulations for the approximations. Here, we consider the Navier–Stokes equation in a moving domain $\Omega(t) \times [\tau, T]$ (a short name for $\bigcup_{t\in[\tau,T]} \Omega(t) \times \{t\}$, which many authors refers to as a non-cylindrical domain). As a first step, we build a domain velocity and transform the equation from the moving domain to a fixed reference domain $\Omega(\tau) \times [\tau, T]$ with an additional term corresponding to this artificial velocity.

2.1. Domain velocity

In the present case, we have the moving domain, that is, the geometrical shape of the domain depends on time (Fig. 1). The principal idea behind to build the domain velocity is to associate the moving domain to a fixed reference domain. More precisely, we need a bijection between the reference and moving domains.

We assume that, there exist, for each $\tau \in [0, T]$, a transformation

$$\psi_\tau : \Omega(\tau) \times [\tau, T] \to \widetilde{\Omega} = \bigcup_{t\in[0,T]} \Omega(t)$$

$$(x_\tau, t) \mapsto x_t = \psi_\tau(x_\tau, t)$$

such that

- For each $t \in [\tau, T]$, the mapping $\psi_\tau(\cdot, t) : \Omega(\tau) \to \Omega(t)$ is a one-to-one correspondence. Thus $\Omega(t) = \{\psi_\tau(x_\tau, t) | x_\tau \in \Omega(\tau)\}$.
- $\psi_\tau(\cdot, \tau) = id(\cdot)$.

Let $\Omega(\tau)$ be our reference domain. Its points will be denoted by x_τ, and the points of the moving domain at a later time t will be denoted by x_t. In particular, with this correspondence it is possible to associate a trajectory to each point x_τ of the domain at time τ according to

$$X(t) = \psi_\tau(x_\tau, t).$$

We define the domain velocity v as the velocity of $X(t)$, i.e.,

$$v(x_\tau, t) := X'(t) = \frac{\partial}{\partial t}\psi_\tau(x_\tau, t).$$

From this, we can obtain

$$\frac{\psi_\tau(x_\tau, t) - \psi_\tau(x_\tau, \tau)}{t - \tau} = v(x_\tau, \tau) + O(t - \tau).$$

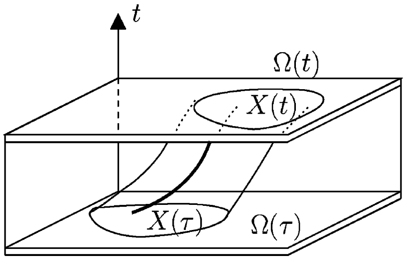

Fig. 1. The moving domain.

From this expression, omitting the error term $O(t-\tau)$ we obtain easily a linear affine approximation for ψ_τ, which we will denote by ϕ_τ:

$$\phi_\tau(x_\tau, t) = x_\tau + (t-\tau)v(x_\tau, \tau). \tag{2.1}$$

2.2. Determination of domain velocity

In order to obtain a possible domain velocity, we note that for a first order approximation, we only need $v = v(x_\tau, \tau)$, the velocity at time τ. Moreover, as this velocity can be chosen independently of the fluid particles velocity, it is clear that we are only constrained to verify some compatibility conditions between the fluid velocity and the domain velocity on the boundary $\partial\Omega(\tau)$ (Fig. 2). For the interior points, we can choose any smooth interior extension of the prescribed velocity on the boundary. This choice should take account of the simplicity of the computation, the smoothness of the resulting velocity field v and the invertibility of the transformation $\phi_\tau(\cdot, t)$.

For the two-dimensional case, one can write the fluid and domain velocities on the boundary $\partial\Omega(\tau)$, respectively, as

$$\begin{cases} u = (u\cdot\widehat{n})\widehat{n} + (u\cdot\widehat{t})\widehat{t}, \\ v = (v\cdot\widehat{n})\widehat{n} + (v\cdot\widehat{t})\widehat{t}, \end{cases} \tag{2.2}$$

where $\widehat{n}$ and $\widehat{t}$ denote, respectively, the unit outward normal and tangential vectors to the boundary.

To maintain the equality between the fluid domain and the geometric domain, one should take the following compatibility condition on the boundary

$$(v\cdot\widehat{n})\widehat{n} = (u\cdot\widehat{n})\widehat{n}. \tag{2.3}$$

One can assign any arbitrary value for the tangential component in v. If the flow on the boundary is dominated by the tangential velocity, then this degree of freedom allows to obtain a very small mesh velocity. This will not be the case for a Lagrangian method which uses the fluid velocity as the mesh velocity, producing large mesh deformation even if the geometry suffers slight deformations. For simplicity, we may take $(v\cdot\widehat{t}) = 0$. In some special situations this choice does not give an appropriate domain velocity. In that case, one has to assign a suitable value for the tangential component. In particular, it depends on the problem to be solved; for the numerical examples given in Section 3, we take different type of boundary conditions for the determination of the domain velocity.

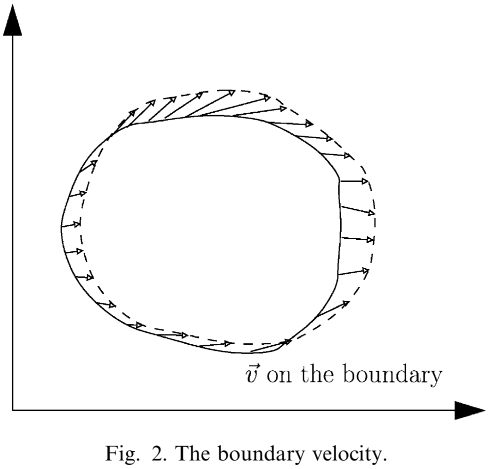

Fig. 2. The boundary velocity.

For the interior points of the domain, the smooth extension of the boundary mesh velocity will be obtained as the solution of the following elliptic problem

$$\begin{cases} -\Delta v = 0 & \text{in } \Omega(\tau) \\ v = (u \cdot \widehat{n})\widehat{n} & \text{on } \partial\Omega(\tau). \end{cases} \tag{2.4}$$

One has to solve two elliptic equation (2.4) at each time step to obtain the mesh velocity for the interior points. In fact, we are not solving this Laplace equation, instead we determine the mesh velocity at each vertex by taking the average of its neighboring vertices, which is simpler.

Remark 2.1. This is not the well-known Laplacian mesh smoothing, here the aim is not to improve the mesh quality, but to extend the mesh velocity on the boundary to the interior points. If the boundary velocity is zero, the Laplacian problem gives the zero velocity solution. Then, even if the quality of the mesh is poor, in this case nothing will happen to it. If one encounters some problems with the triangles, then we suggest to use the optimization based smoothing ideas given in Canann et al. [2]. Also one can use the Winslow smoothing technique [20]. For more details, we suggest the reader to refer the book by Knupp and Steinberg [11].

From Eq. (2.1), it is evident that the invertibility of $\phi_\tau(\cdot,t)$ depends on the domain velocity v. Let us examine the required conditions for the existence of $\phi_\tau^{-1}(\cdot,t)$, the inverse function of $\phi_\tau(\cdot,t)$ for a fixed t. The gradient of the transformation given in (2.1) is

$$\nabla_{x_\tau}\phi_\tau(x_\tau,t) = I + (t-\tau)\nabla_{x_\tau}v(x_\tau,\tau).$$

To ensure the local invertibility of ϕ_τ, we apply the inverse function theorem. In order to apply this theorem, we assume the following two conditions:

- $v(x_\tau,t) \in \mathscr{C}^1(\Omega(\tau))$
- $\|(t-\tau)\nabla_{x_\tau}v(x_\tau,t)\| < 1$

where $\|\cdot\|$ denotes any matrix norm.

From the second condition, we have

$$|t-\tau| < \frac{1}{\sup_{x_\tau \in \Omega(\tau)} \|\nabla_{x_\tau}v(x_\tau,\tau)\|} \tag{2.5}$$

which gives a bound for the time step.

The detailed method is given via the following algorithm.

Algorithm. To start the procedure, we assume that the fluid velocity $u(x,t)$ and the pressure $p(x,t)$ are known at the time step $t=t_n$, and we denote them, respectively, as u_n, p_n and the domain by $\Omega_n = \Omega(t_n)$.

Step 1. Domain velocity: Obtain the domain velocity v_n, for example, by solving the elliptic problem given in (2.4). With this velocity field it is possible to define the transformation $\phi_{t_n}(\cdot,t_{n+1})$ as defined in Eq.(2.1), which is the linearization of the one to one correspondence between $\Omega(t_n)$ and $\Omega(t_{n+1})$.

Step 2. Fluid velocity and pressure in the reference domain: With the newly obtained domain correspondence ϕ_{t_n} pose the problem of finding u_{n+1}, p_{n+1} at time t_{n+1} on the reference domain $\Omega(t_n)$ by using ϕ_{t_n} as the change of variables (the details are given in the following Section 2.3). Then, solve the Stokes-like equation as given in Eqs. (2.24)–(2.28) numerically. Let us denote the obtained velocity and pressure, respectively, by $\widetilde{u}_{n+1}, \widetilde{p}_{n+1}$.

Step 3. *Domain* $\Omega(t_{n+1})$: Obtain the new domain $\Omega(t_{n+1})$. In fact, one obtain the triangulation for the domain $\Omega(t_{n+1})$ by transforming the vertices of the triangles in the domain $\Omega(t_n)$ by using the transformation $\phi_{t_n}(\cdot, t_{n+1})$. If P is a vertex, the transformed vertex will be $\phi_{t_n}(P, t_{n+1}) = P + \Delta t v(P)$.

Step 4. *Fluid velocity and pressure in the new domain*: In principle, if the computational program uses for the vertices, a vertex, data structure with fields `x, y, u, p`, then it is simply correspond to update the values of the vertex coordinates `x, y, u, p`.

Step 5. *Next time step*: If t_{n+1} is less than the final time T, then $n = n + 1$ and go to *Step* 1.

2.3. Transformed equations

The transformation ϕ_τ defines the change of variables between x_t and x_τ. For each vector or scalar field defined on $\Omega(\tau)$, this transformation gives back a field defined on $\Omega(t)$. The following diagram summarizes this fact in the case of the velocity vector field

$$\begin{tikzcd}
u_\tau(\cdot,t): & \Omega(\tau) \arrow[r] \arrow[d, "\phi_\tau"] & \mathbb{R}^N \\
u(\cdot,t): & \Omega(t) \arrow[r] & \mathbb{R}^N \arrow[u, "id"']
\end{tikzcd}.$$

In general, with ϕ_τ we can transform every function depending on $x_t \in \Omega(t)$ into a function depending on $x_\tau \in \Omega(\tau)$. Then, for any given function $g(x_t, t)$ we can apply this linearized change of variable and define

$$g_\tau(x_\tau, t) = g(\psi_\tau(x_\tau, t), t) = g(x_\tau + (t-\tau)v(x_\tau, \tau), t).$$

From this expression, one can compute the partial derivatives of $g_\tau(x_\tau, t)$ with respect to the spatial variable $(x_\tau)_k$ as

$$\frac{\partial g_\tau(x_\tau,t)}{\partial (x_\tau)_k} = \frac{\partial g(x,t)}{\partial x_j}\frac{\partial}{\partial (x_\tau)_k}\{(x_\tau)_j + (t-\tau)v_j(x_\tau,\tau)\} = \frac{\partial g(x,t)}{\partial x_j}\left\{\delta_{kj} + (t-\tau)\frac{\partial v_j}{\partial (x_\tau)_k}\right\} = \frac{\partial g(x,t)}{\partial x_k} + o(t-\tau) \tag{2.6}$$

and with respect to the time variable t as

$$\frac{\partial g_\tau(x_\tau,t)}{\partial t} = \frac{\partial g(x,t)}{\partial t} + v \cdot \nabla g(x,t). \tag{2.7}$$

That is, from (2.6) and (2.7), we have, to the first order

$$\frac{\partial g(x,t)}{\partial x_k} = \frac{\partial g_\tau(x_\tau,t)}{\partial (x_\tau)_k}, \tag{2.8}$$

$$\frac{\partial g(x,t)}{\partial t} = \frac{\partial g_\tau(x_\tau,t)}{\partial t} - v \cdot \nabla g_\tau(x_\tau,t). \tag{2.9}$$

This tells that the spatial partial derivatives are identical up to the first order, but the time derivative has an additional term, namely $v \cdot \nabla g(x,t)$.

For example, $\mathrm{div} f_\tau = \mathrm{div} f$, $\nabla\phi_\tau = \nabla\phi$, since they include only the spatial derivatives. To make precise notations for the domains, we take

$$B = \bigcup_{t\in[0,T]} \Omega(t) \times \{t\}, \quad \Sigma_1 = \bigcup_{t\in[0,T]} \Gamma_1(t) \times \{t\}, \quad \text{and} \quad \Sigma_2 = \bigcup_{t\in[0,T]} \Gamma_2(t) \times \{t\}, \tag{2.10}$$

where $\Gamma_1(t)$ and $\Gamma_2(t)$ are, respectively, the portion of the boundary where the velocity and the normal stress are prescribed.

The mathematical formulation of the free boundary flow consists of the following Navier–Stokes system of equations in the variable domain

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u - v\Delta u + \nabla p = f \quad \text{in} \quad B, \tag{2.11}$$

$$\operatorname{div} u = 0 \quad \text{in} \quad B, \tag{2.12}$$

$$u(x,0) = u_0(x) \quad \text{in} \quad \Omega(0), \tag{2.13}$$

$$u(x,t) = w(x,t) \quad \text{on} \quad \Sigma_1, \tag{2.14}$$

$$\sigma(x,t) \cdot \widehat{n} = g(x,t) \quad \text{on} \quad \Sigma_2, \tag{2.15}$$

where $u(x,t), p(x,t)$ and $f(x,t)$, respectively, denote the fluid velocity, pressure and the density of body forces per unit mass, v is the kinematic viscosity coefficient, $u_0(x)$ is the initial velocity, $w(x,t)$ is the prescribed velocity on the boundary, $\sigma(x,t) = -p(x,t)I + v(\nabla u(x,t) + \nabla^t u(x,t))$ is the stress tensor and $g(x,t)$ is the prescribed stress on the boundary.

We apply the change of variables to transform the Navier–Stokes system from a variable domain to a fixed reference domain. The transformed Navier–Stokes equation according to the change of variables as given in (2.8) and (2.9) yields

$$\frac{\partial u_\tau}{\partial t} + ((u_\tau - v) \cdot \nabla)u_\tau - v\Delta u_\tau + \nabla p_\tau = f_\tau \quad \text{in} \quad \Omega(\tau) \times [\tau, T], \tag{2.16}$$

$$\operatorname{div} u_\tau = 0 \quad \text{in} \quad \Omega(\tau) \times [\tau, T], \tag{2.17}$$

$$u_\tau(x_\tau, \tau) = u(x_\tau, \tau) \quad \text{in} \quad \Omega(\tau), \tag{2.18}$$

$$u_\tau(x_\tau, t) = w_\tau(x_\tau, t) \quad \text{on} \quad \Gamma_1(\tau) \times [\tau, T], \tag{2.19}$$

$$\sigma_\tau(x_\tau, t) \cdot \widehat{n} = g_\tau(x_\tau, t) \quad \text{on} \quad \Gamma_2(\tau) \times [\tau, T], \tag{2.20}$$

where $\Gamma_1(\tau) \cup \Gamma_2(\tau) = \partial\Omega(\tau)$. The initial condition will be the solution at time $t = \tau$, that is, $u(x_\tau, \tau)$. One can think of solving the variable domain problem by solving a sequence of fixed reference domain problems each one on a time strip $[\tau_i, \tau_{i+1}]$. In this way, the solution at the previous time step becomes the initial condition for the present problem.

Now, the modified equation is defined in a fixed domain and one can be intended to solve this problem by any standard methods available in the literature. In our case, by applying the characteristic method given in [17], we convert the Navier–Stokes equation to a Stokes-like equation and then solve it by the finite element method.

2.4. The characteristic method

In the following, we describe the characteristic method which will be used to treat the non-linear term in Eq. (2.16).

Suppose $X_w(s)$ is the solution of the following ordinary differential equation (which is a terminal-value problem):

$$\begin{cases} \dfrac{\mathrm{d}X_w(s)}{\mathrm{d}s} = w(X_w(s), s) \quad s \in (\tau, t) \\ X_w(t) = x_\tau \end{cases}, \tag{2.21}$$

where $w(x, s)$ is a vector field defined on $\Omega(\tau) \times [\tau, t]$.

If we define $f(s) := g(X_w(s), s)$, where g is a vector or scalar field defined on $\Omega(\tau) \times [\tau, t]$, then its total derivative becomes

$$f'(s) = \frac{\partial g}{\partial t} + w \cdot \nabla g. \tag{2.22}$$

One can compare the following term as given in the transformed Navier–Stokes equation (2.16)

$$\frac{\partial u_\tau}{\partial t} + ((u_\tau - v) \cdot \nabla) u_\tau \tag{2.23}$$

with the right hand side of (2.22). This term can be viewed as the "material derivative" of u_τ along the trajectory defined by $w = u_\tau - v$. After implementing the value of w in the ordinary differential equation (2.21) and solving it numerically, we obtain $X_w(\tau)$.

Now substitute the value of $X_w(\tau)$ in the "material derivative" as given in (2.23), we obtain the following differential-difference term

$$\frac{\partial u_\tau}{\partial t} + ((u_\tau - v) \cdot \nabla) u_\tau = \frac{\mathrm{d}}{\mathrm{d}t} u_\tau(X_w(s), s) \approx \frac{u_\tau(X_w(t), t) - u_\tau(X_w(\tau), \tau)}{\Delta t} = \frac{u_\tau(x_\tau, t)}{\Delta t} - \frac{u_\tau(X_w(\tau), \tau)}{\Delta t},$$

where $\Delta t = t - \tau$.

In the previous expression the first term is an unknown which will be obtained as the solution of the Stokes-like equation (2.24) given below, whereas the second term is computed from the numerical solution of the ordinary differential equation (2.21). In practice, if we think of t as t_{n+1} and τ as t_n, then $u_\tau(x_\tau, t) = u_\tau(x_\tau, t_{n+1})$ represents the unknown velocity field at the next time step but on a known point x_τ and $u_\tau(X_w(\tau), \tau) = u_\tau(X_w(t_n), t_n)$ is the solution already computed but evaluated at the point $X_w(t_n) = X_w(\tau)$ obtained as solution of (2.21).

After incorporating the above modification in the transformed Navier–Stokes equation (2.19), one can obtain the following steady-state Stokes-like equation

$$\left(\frac{1}{\Delta t} I - \nu \Delta\right) u_\tau + \nabla p_\tau = f_\tau + \frac{u_\tau(X_w(\tau), \tau)}{\Delta t} \quad \text{in} \quad \Omega(\tau) \times [\tau, T], \tag{2.24}$$

$$\operatorname{div} u_\tau = 0 \quad \text{in} \quad \Omega(\tau) \times [\tau, T], \tag{2.25}$$

$$u_\tau(x_\tau, \tau) = u(x_\tau, \tau) \quad \text{in} \quad \Omega(\tau), \tag{2.26}$$

$$u_\tau(x_\tau, t) = w_\tau(x_\tau, t) \quad \text{on} \quad \Gamma_1(\tau) \times [\tau, T], \tag{2.27}$$

$$\sigma_\tau(x_\tau, t) \cdot n = g_\tau(x_\tau, t) \quad \text{on} \quad \Gamma_2(\tau) \times [\tau, T]. \tag{2.28}$$

3. Numerical examples

To show the accuracy and efficiency of the ALE method, we present three model problems here. We use the finite element method to solve the Stokes-like equation (2.24)–(2.28). For the finite element basis functions, we use piecewise linear functions ($\mathscr{P}_1$) on triangles for the pressure and $\mathscr{P}_1$-Iso$\mathscr{P}_2$ functions for the velocity. The $\mathscr{P}_1$-Iso$\mathscr{P}_2$ function is a piecewise linear function on a triangulation obtained from the original one, dividing each triangle into four sub-triangles. Fig. 3 shows both triangulations.

As pointed out earlier, to obtain the domain velocity we are not solving the Laplace equation given in Eq. (2.4). Instead of that we compute the velocity at each vertex of the mesh as the average of its neighboring vertices. The velocity for the boundary points will be fixed according to the fluid velocity on the boundary, which depends on the particular problem. The problems are given below in detail.

3.1. Propagation of a solitary wave

The first model problem is the propagation of a solitary wave, which was used earlier for comparative purposes by many authors including Hansbo [8], Navti et al. [16] and Ramaswamy and Kawahara [18]. We use this example to test the present method and compare the outputs, like the maximum run-up height and the corresponding time, and the maximum pressure with the available results in the literature. This model consists of the motion of a viscous incompressible fluid in a tank with fixed walls on three sides (bottom and two vertical walls). The upper surface is a free surface, the fluid moves freely. The gravitational force will control the motion of the waves. Fig. 4 shows the model situation.

The mathematical formulation of the present example is described by the following Navier–Stokes equations with initial and boundary conditions

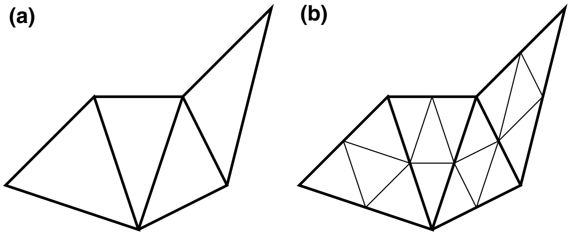

Fig. 3. $\mathscr{P}_1$ and $\mathscr{P}_1$-Iso$\mathscr{P}_2$ triangulations. (a) Triangles for pressure, (b) triangles for velocity.

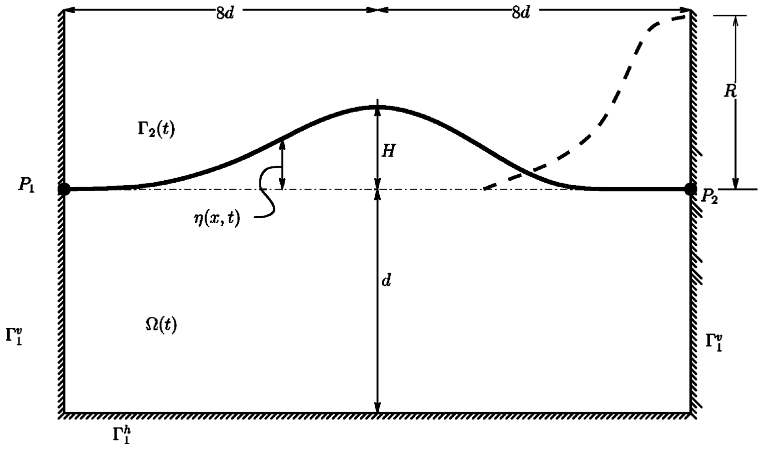

Fig. 4. Wave propagation.

$$
\begin{aligned}
&\frac{\partial u}{\partial t} + (u\cdot\nabla)u - \nu\Delta u + \nabla p = f \quad \text{in } B,\\
&\operatorname{div} u = 0 \quad \text{in } B,\\
&u(x,0) = u_0(x) \quad \text{in } \Omega(0),\\
&u(x,t) = 0 \quad \text{on } \Gamma_1^h \times [0,T],\\
&u_1(x,t) = 0 \quad \text{on } \Gamma_1^v \times [0,T],\\
&(\sigma(x,t)\cdot\widehat{n})_2 = 0 \quad \text{on } \Gamma_1^v \times [0,T],\\
&\sigma(x,t)\cdot\widehat{n} = 0 \quad \text{on } \Sigma_2,
\end{aligned}
$$

where B and Σ_2 are defined in (2.10) and Γ_1^h, Γ_1^v, respectively, denote the horizontal and vertical walls. The time step used for the numerical simulation was $t = 0.05$ s, fluid viscosity 1.0 kg/ms, and density 1.0 kg/m^3. A gravitational acceleration of magnitude 9.8 m/s^2 acts vertically on the downward direction. The initial conditions for this problem are taken from the approximations given by Laitone [12]. Theoretically, Laitone's formula holds for only an infinitely long channel. But the computations can be performed only in a finite domain and the fluid at a distance from the wave crest is essentially still, it is desirable to define a finite practical length of the solitary wave. More precisely, the initial conditions are given as

$$u_1 = \sqrt{gd}\left(\frac{H}{d}\right)\operatorname{sech}^2\left(\sqrt{\frac{3H}{4d^3}}x\right),$$

$$u_2 = \sqrt{\frac{3g}{d}}\left(\frac{H}{d}\right)^{3/2} y \operatorname{sech}^2\left(\sqrt{\frac{3H}{4d^3}}x\right) \tanh\left(\sqrt{\frac{3H}{4d^3}}x\right),$$

$$\eta = d + H \operatorname{sech}^2\left(\sqrt{\frac{3H}{4d^3}}x\right),$$

where g is the gravitational force, $d = 10, H = 2$, u_1 and u_2 are the velocities in the x and y directions and η is the free surface elevation.

From the given boundary condition on Σ_1, a natural question arises, what will happen in the points of contact between the free surface and the vertical walls at P_1 and P_2 as shown in Fig. 4. More precisely, according to the boundary condition on Σ_1 the velocity is zero at P_1 and P_2, and the level of the liquid on the walls will remain fixed. This is contrary to the common experience, which shows that the level changes according to the motion of the fluid. So, we will use, as in [8,16], the slip boundary conditions for the vertical walls.

As mentioned earlier, there are several choices for the domain velocity. For this model, it seems that a good choice for the domain velocity is to restrict it to a vertical velocity. In this way the triangles will maintain basically its shape and the re-meshing process will be avoided. For the boundary conditions of the domain velocity on the free surface we look for a vertical velocity whose normal component is equal to the normal component of u, the fluid velocity. That is, find α such that $v = (u\cdot\widehat{n})\widehat{n} + \alpha\widehat{t}$ is a vertical vector. If $\widehat{e_1}$ and $\widehat{e_2}$ are the unit vertical and horizontal vectors, respectively, then the condition becomes $v\cdot\widehat{e_1} = 0$, from this we obtain

$$v = (u\cdot\widehat{n})\left[\widehat{n} - \frac{\widehat{e_1}\cdot\widehat{n}}{\widehat{e_1}\cdot\widehat{t}}\widehat{t}\right]. \tag{3.1}$$

On the bottom of the domain we take $v = 0$.

The choice taken in Eq. (3.1) is allowed as long as the free surface does not become vertical, which means that the motion of the free surface is not so strong. Also, we have tested the domain velocity on the free surface equal to the fluid velocity and equal to the normal component of the fluid velocity, and obtained very similar results, except that the mesh quality decreased rapidly.

To compare the numerical results obtained from the ALE, we use the computational results given by Hansbo [8], Navti et al. [16] and Ramaswamy and Kawahara [18]. In addition, we have performed the numerical simulations by using Fluent. The run-up height, the time when the wave hits the right side wall and the maximum pressure are presented in Table 1. From Laitone's approximation, one can obtain the run-up height as $R = 14.2$. Table 1. gives a clear picture that how much accurate results have been obtained from the ALE method. Further, we are able to continue the simulation for a long time, means that the wave crest can move several times between the vertical walls, without facing any problems with the meshes. The choice of the mesh velocity helps to preserve the triangulation of the first time for the entire simulation.

The triangulation of the domain at time $t = 0$ is shown in Fig. 5. In Fig. 6, we presented the velocity vector field at $t = 7.6$ s, when the wave stop its motion to the right side and begin to move to the left side. Further, Fig. 7 shows the velocity magnitude at some subsequent times.

Fig. 8 shows the plot of the run-up height versus time. The solid line denotes the data obtained from Fluent, whereas the broken line represents the ALE output. This figure again highlights our claim that the ALE method produces sharp approximate results.

3.2. Broken dam problem

In this example, a block of water is kept at rest in a cavity in the initial stage using a thin paper film. The paper film was removed (instantaneously, at time $t = 0$), the water spreads out under the influence of gravity. This broken dam problem was studied numerically and experimentally in [8,13,14]. Hansbo [8] used the characteristic streamline diffusion method and performed the numerical simulation with and without remeshing the domain, whereas in [13] the authors applied a splitting method for the time discretization, and at each time step, they have solved two advection problems—one for the predicted velocity field and the other for the volume fraction of liquid. The outputs of the present method are compared with the results obtained in [8,13] and Fluent.

Table 1
Comparison table: height, time and pressure

	Hansbo [8]	Navti [16]	Ramasamy [18]	Fluent	ALE
Height	14.4	13.4	14.48	14.19	14.27
Time	7.66	7.6	7.7	7.55	7.6
Press	132.2	–	130	–	131.66

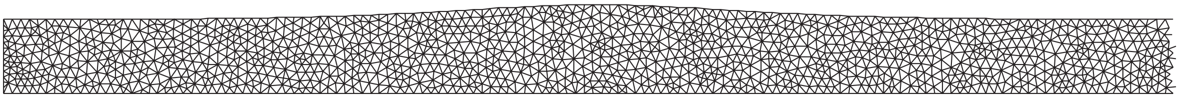

Fig. 5. Triangulation of the domain at the initial time $t = 0$ s.

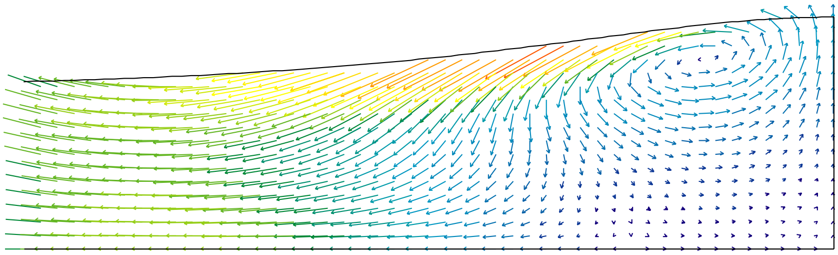

Fig. 6. Detail of the velocity vector field when the wave reverse its motion at time $t = 7.6$ s.

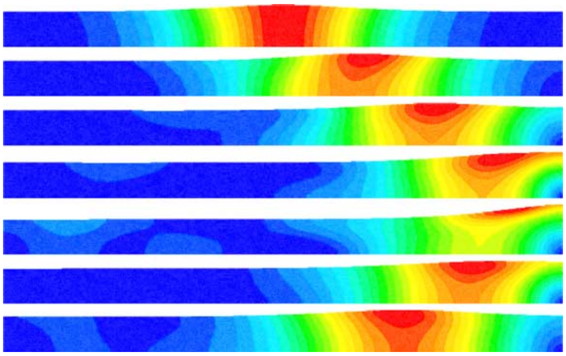

Fig. 7. The velocity magnitude contours obtained at time $t = 0$, 2, 4, 6, 8, 10 and 12 s.

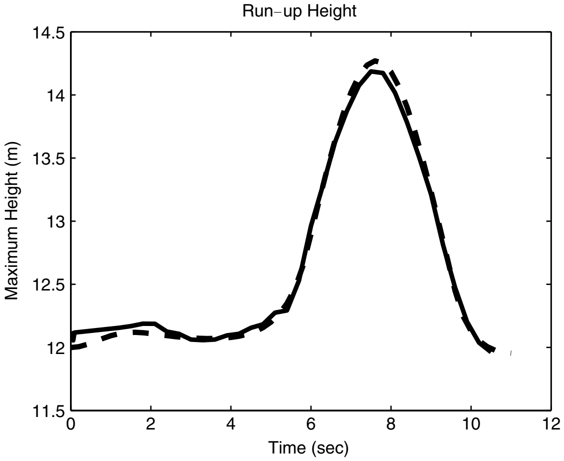

Fig. 8. Run-up height: fluent (solid), ALE (dashed).

The following Navier–Stokes equations corresponding to the present model, are solved subject to the initial and boundary conditions

$$\begin{aligned}
&\frac{\partial u}{\partial t} + (u \cdot \nabla)u - \nu\Delta u + \nabla p = f \quad \text{in } B,\\
&\operatorname{div} u = 0 \quad \text{in } B,\\
&u(x,0) = u_0(x) \quad \text{in } \Omega(0),\\
&u_1(x,t) = 0 \quad (\sigma(x,t)\cdot\widehat{n})_2 = 0 \quad \text{on } \Gamma_1^v \times [0,T],\\
&u_2(x,t) = 0 \quad (\sigma(x,t)\cdot\widehat{n})_1 = 0 \quad \text{on } \Gamma_1^h \times [0,T],\\
&\sigma(x,t)\cdot\widehat{n} = 0 \quad \text{on } \Sigma_2.
\end{aligned}$$

The initial stage of the dam and the triangulation of the domain at time $t = 0$ are shown in Fig. 9.

The velocity and pressure contours obtained from the ALE method at various simulation times are presented in Fig. 10. Apart from the comparisons with the results of Hansbo [8], Maronnier et al. [13], we have performed the simulations by using Fluent. The dimensionless position x/L (where L is the initial length of the dam) of the liquid front along the bottom of the cavity, versus the dimensionless time $t\sqrt{2g/L}$ is plotted in Fig. 11. The black dotted line, solid line and the dashed line, respectively, denote the data

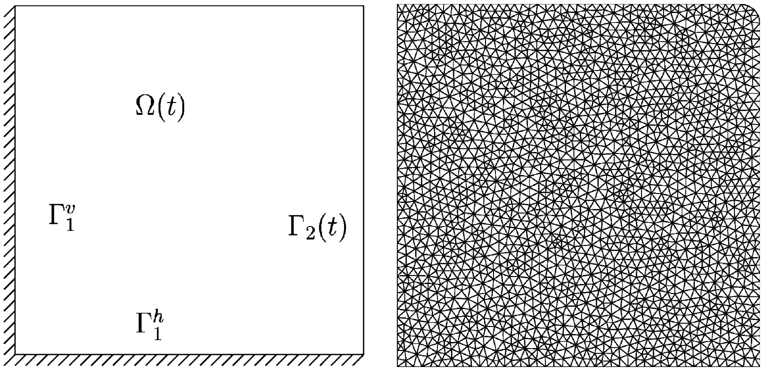

Fig. 9. The Dam problem.

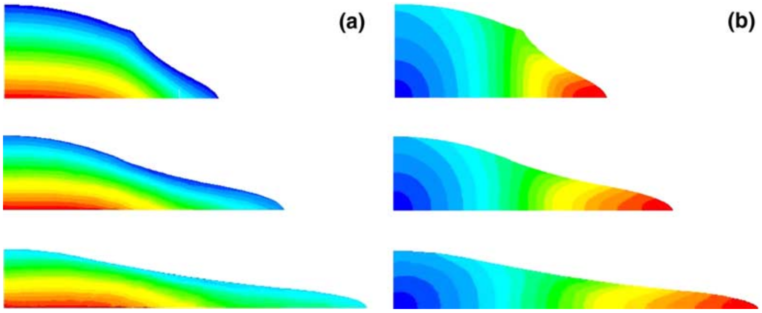

Fig. 10. The contours obtained at time $t = 0.08$, 0.12 and 0.16 s. (a) Pressure, (b) velocity magnitude.

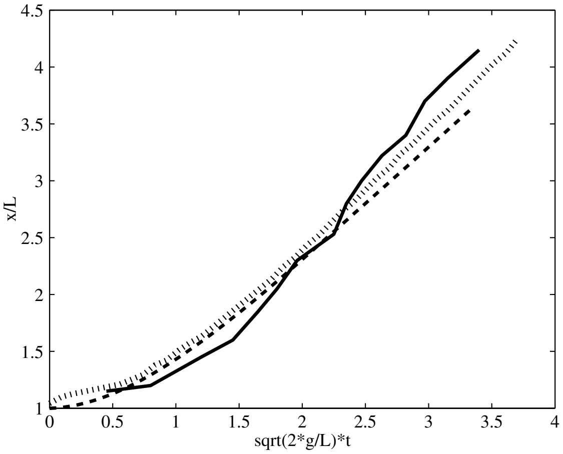

Fig. 11. Broken dam problem: experimental (solid), ALE (dotted) and fluent (dashed).

obtained from the ALE method, experimental and Fluent. One can notice the accuracy of the present method from the three curves.

Here also, one can have different choices for the mesh velocity v as used in Section 3.1. The best result was obtained for the choice of the mesh velocity which equals to the fluid velocity on the free boundary. In addition, we have a small problem with the mesh velocity to the upper right corner point, which causes a fast degradation of the quality of the triangulation near to this point. To avoid this difficulty, we smoothen the upper right corner.

3.3. Movement of a gas bubble in a fluid

The third example deals with the movement of an air bubble in a viscous incompressible fluid, where the fluid is kept in a cavity with a free upper surface. Here, the upper surface as well as the path of the bubble depend on time. Fig. 12(a) represents the model.

In the present case, the domain $\Omega(t)$ has been decomposed into the region occupied by the bubble $\Omega_{\mathrm{b}}(t)$ with boundary $\Gamma_{\mathrm{b}}(t)$ and the region occupied by the fluid denoted by $\Omega_{\mathrm{f}}(t)$ with the exterior boundary made of two parts, namely Γ_1 (walls) and $\Gamma_2(t)$ (free surface). Thus, the domain of interest is given by $\Omega_{\mathrm{f}}(t) = \Omega(t) \setminus \Omega_{\mathrm{b}}(t)$, with the boundary $\Gamma_{\mathrm{f}}(t) = \Gamma_{\mathrm{b}}(t) \cup \Gamma_1 \cup \Gamma_2(t)$.

We assume that the bubble is filled with an ideal gas, therefore, we have

$$P_{\mathrm{b}}V_{\mathrm{b}} = C,$$

where P_{b} is the internal pressure, V_{b} is the volume of the bubble and C is the constant arising from the ideal gas law $PV = nRT$.

The following Navier–Stokes equations with the initial and boundary conditions describe the mathematical formulation of the present model

$$\begin{aligned}
&\frac{\partial u}{\partial t} + (u \cdot \nabla)u - v\Delta u + \nabla p = f \quad \text{in } \Omega_{\mathrm{f}}(t),\\
&\operatorname{div} u = 0 \quad \text{in } \Omega_{\mathrm{f}}(t),\\
&\sigma(x,t) \cdot \widehat{n} = (\alpha\kappa - P_{\mathrm{b}}) \cdot \widehat{n} \quad \text{on } \Gamma_{\mathrm{b}}(t),\\
&u(x,t) = 0 \quad \text{on } \Gamma_1,\\
&\sigma(x,t)\widehat{n} = 0 \quad \text{on } \Gamma_2(t),\\
&u(x,0) = 0 \quad \text{in } \Omega_{\mathrm{f}}(0),\\
&P_{\mathrm{b}}(t)V_{\mathrm{b}}(t) = C,\\
&V_{\mathrm{b}}(t) = \int_{\Omega_{\mathrm{b}}(t)} 1\,\mathrm{d}x,
\end{aligned}$$

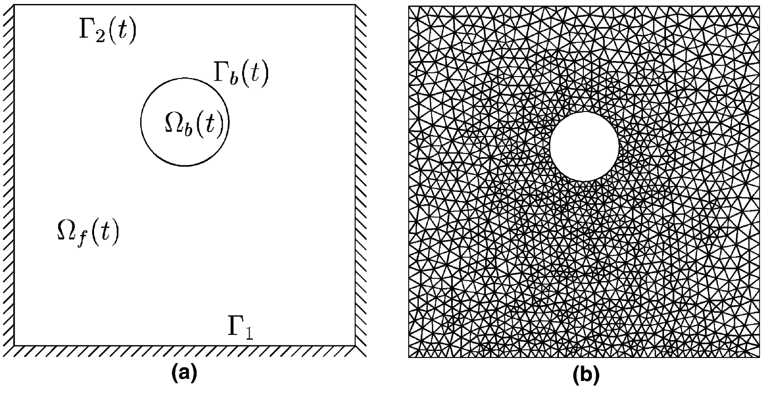

Fig. 12. Motion of a gas bubble in a fluid.

where α is the surface tension coefficient, and κ is the curvature of the boundary of the bubble. We took $\alpha = 0$ in our numerical computations.

The triangulation of the domain at time $t = 0$ is presented in Fig. 12(b). Here, we are not using the remeshing strategy to the domain throughout the numerical simulation. A good choice of the mesh velocity helps to preserve the quality of the triangulation as long as possible. More precisely, we assign a mesh velocity to the vertices on the bubble boundary which guarantees a uniform distribution of these vertices as

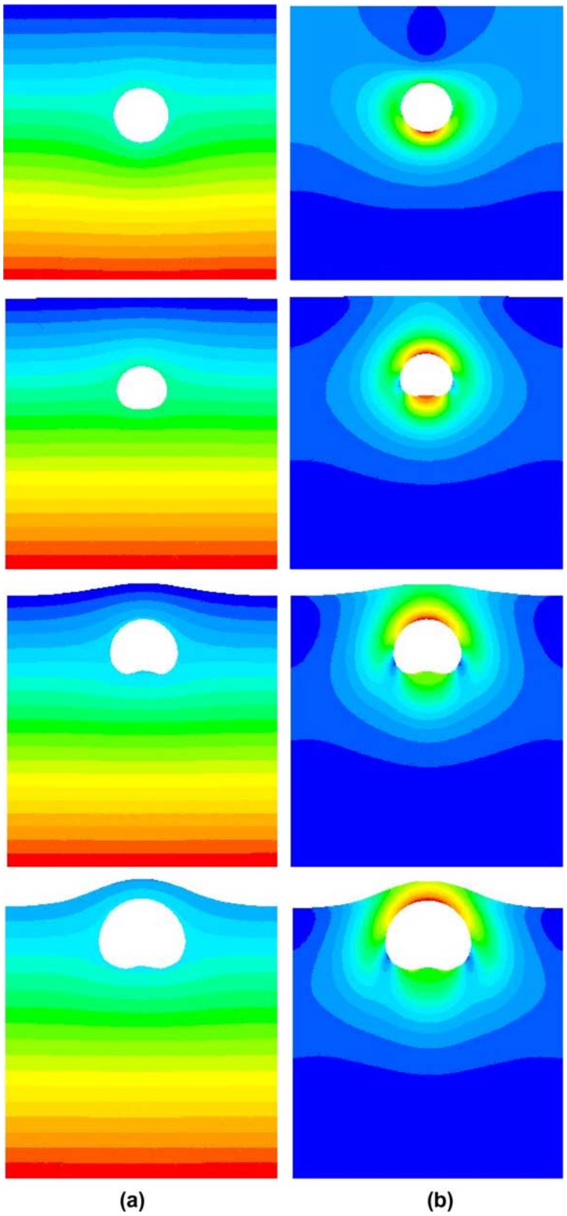

Fig. 13. The contours of pressure at time $t = 0.00$, 0.03, 0.06 and 0.076 s (a), and velocity magnitude at time $t = 0.02$, 0.04, 0.06 and 0.076 s (b).

the bubble rises up. For this, at $t = 0$, we compute the angle between the horizontal direction and the line passing through the center of mass of the bubble and each vertex. Then at each time step we compute a tangential velocity $\vec{v}_k$ for each vertex on the bubble boundary in such a way that choosing $v = (u \cdot \widehat{n})\widehat{n} + \vec{v}_k$ preserves this angle. We have observed problems even with this new mesh velocity with the triangles near the bubble. They start to accumulate on the top of the bubble after a few steps. To avoid this problem, when computing an interior extension of the mesh velocity already defined on the boundary, we force the vertices in the neighborhood of the boundary of the bubble to move faster than the vertices far from the bubble. This was accomplished by computing a mesh velocity for each interior vertex as a weighted average of the mesh velocity of its neighbors. The weight was chosen as a decreasing function of the vertex distance to the bubble boundary. The distance is measured as the minimal number of intermediate vertices needed to walk from the given vertex to the bubble boundary.

The time step has been taken as $t = 0.002$. The contour of the pressure and the velocity magnitude are presented in Fig. 13. From these figures one can justify the present method observing that the shape of the bubble at different time steps is the expected one from the common experiences.

4. Conclusions

We have presented the ALE method in this article for the simulation of free boundary problems arising in fluid dynamics. Introducing an artificial domain velocity we transform the moving boundary problem to a fixed domain problem. This provides the possibilities to use the available software to solve the fixed domain problem. Here, we have applied the characteristic method to transform the Navier–Stokes equations (defined in the fixed domain) into a Stokes-like equation. Our choices for the mesh velocity help to avoid the re-meshing up to certain time steps. For the wave example, the proposed choice of the mesh velocity preserves the meshes of the initial time for the entire simulation, and avoids the re-meshing process for any arbitrary time interval. In the other two examples, one can re-mesh the domain when the meshes started to distorted, and continue the simulation for a long time. By this way, one can avoid the frequent re-meshing and as a consequence, one can avoid the projection error, and save CPU time and memory. Finally, the proposed method performs well and produces sharp approximations in comparison with the earlier experimental and numerical results, and one can generalize it to several free boundary problems arising in practical applications. An interesting extension of the proposed method will be the usage of non-standard boundary condition for the Navier–Stokes equation. More precisely, many engineering problem are stated in terms of condition on the pressure instead of the normal stress. In Conca et al. [3] this non-standard boundary condition is studied for fixed domain, and from here an extension to moving boundaries can be considered.

Acknowledgements

The authors convey their gratitude to the anonymous referees for their valuable suggestions and comments, which really helps to strengthen the article by improving the presentation.

This work has been partially supported by FONDAP through its Programme on Mathematical-Mechanics. The authors gratefully acknowledge the Chilean and French Governments through the Scientific Comité Ecos-Conicyt.

Finally, our sincere thanks goes to Dr. B. Maury, Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, 75252 Paris Cedex 05, France, for his introductory lectures on ALE method during his visit to Chile, and Dr. J. San Martín, Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, for his fruitful discussions and help to overcome the computational difficulties.

References

[1] T. Belytschko, J. Kennedy, Finite element approach to pressure wave attenuation by reactor fuel subassemblies, J. Press. Technol. (1975) 172–177.

[2] S. Canann, J. Tristano, M. Staten, An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes, in: Seventh International Meshing Roundtable, Sandia National Labs, 1998, pp. 479–494.

[3] C. Conca, F. Murat, O. Pironneau, The Stokes and Navier–Stokes equations with boundary conditions involving the pressure, Jpn. J. Math 20 (2) (1994) 279–318.

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

[5] J. Donea, S. Giuliani, J. Halleux, An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions, Comput. Methods Appl. Mech. Engrg. 33 (1982) 689–723.

[6] J. Douglas, T. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM J. Numer. Anal. 19 (1982) 871–885.

[7] C. Frederiksen, A. Watts, Finite element method for time-dependent incompressible free surface flow, J. Comp. Phys. 39 (1981) 282–304.

[8] P. Hansbo, The characteristic streamline diffusion method for the time-dependent incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 99 (1992) 171–186.

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

[10] T. Hughes, W. Liu, T. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Engrg. 29 (1981) 329–349.

[11] P. Knupp, S. Steinberg, The Fundamentals of Grid Generation, CRC Press, Boca Raton, FL, 1993.

[12] E. Laitone, The second approximation to cnoidal and solitary waves, J. Fluid Mech. 9 (1960) 430–444.

[13] V. Maronnier, M. Picasso, J. Rappaz, Numerical simulation of three dimensional free surface flows, Int. J. Numer. Methods Fluids 42 (2003) 697–716.

[14] J. Martin, W. Moyce, An experimental study of the collapse of liquid columns on a rigid horizontal plate, Philos. Trans. R. Soc. London Ser. A 244 (1952) 312–324.

[15] B. Maury, Characteristics ALE method for the unsteady 3D Navier–Stokes equations with a free surface, Int. J. Comp. Fluid Dyn. 6 (1996) 175–188.

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

[17] O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier–Stokes equations, Numer. Math. 38 (1982) 309–332.

[18] B. Ramaswamy, M. Kawahara, Arbitrary Lagrangian–Eulerian finite element method for unsteady convective incompressible viscous free surface flow, Int. J. Numer. Methods Fluids 7 (1987) 1053–1074.

[19] E. Süli, Convergence and nonlinear stability of the Lagrange–Galerkin method for the Navier–Stokes equations, Numer. Math. 53 (1988) 459–483.

[20] A. Winslow, Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh, J. Comp. Phys. 2 (1967) 149–172.