ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 187 (2003) 255–273

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

A finite element technique for multifluid incompressible flow using Eulerian grids

P.D. Minev [a,*], T. Chen [b], K. Nandakumar [b]

[a] *Department of Mathematical Sciences, University of Alberta, Edmonton, Alta., Canada T6G 2G1*
[b] *Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alta., Canada T6G 2G6*

Received 25 June 2002; received in revised form 10 February 2003; accepted 14 February 2003

Abstract

The paper presents a finite element method for 3D incompressible fluid flows with capillary free boundaries. It uses a fixed Eulerian grid of 10 nodes ($\mathbb{P}_2 - \mathbb{P}_1$) tetrahedra and tracks the free boundary using a six nodes ($\mathbb{P}_2$) triangular surface grid. In order to improve the mass conservation properties of the method, a local enrichment of the finite element basis in the elements intersected by the free boundaries is employed. In addition to the surface tracking, it also advects a smooth indicator function for an easy identification of the fluid properties in the different parts of the domain. The advective part of the Navier–Stokes equations is split and integrated with a characteristic method. The remaining generalized Stokes problem is resolved by means of an inexact outer–inner (Uzawa) iteration with a properly chosen preconditioner. The performance of this technique is evaluated on several problems involving droplets in viscous liquids.

Keywords: Free boundary problems; Incompressible flow; Finite element method

1. Introduction

The numerical solution of incompressible flows involving more than one fluid is, with no doubt, one of the most challenging tasks in Computational Fluid Dynamics (CFD). The problems of tracing the interfaces, properly interpolating the unknown quantities in the time dependent domains and the solution of the resulting (often badly conditioned) linear systems is a serious problem in 2D but it becomes a much harder task in 3D. In addition, one often needs to cope with the singularities imposed by surface forces on the time dependent interfaces between the fluids. The problem can be further complicated if contact lines of different types are to be taken into account. The need for solution of such problems comes from a variety of appli-

[*] Corresponding author.
E-mail addresses: minev@ualberta.ca (P.D. Minev), tc8@ualberta.ca (T. Chen), kumar.nandakumar@ualberta.ca (K. Nandakumar).
URL: www.math.ualberta.ca/pminev.

0021-9991/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00098-6

cations, the most important probably being the understanding of the dynamics of multiphase flows. Initially simulations have been used to study the behaviour of a single bubble or droplet in another fluid (see for example [1,2]). The development of the computational techniques and resources made it possible nowadays to perform a direct simulation of multiparticle flows. Probably the most impressive computations of a large number of droplets in a viscous fluid have been recently presented by Tryggvason et al. [3] (see also the references therein). Other areas of application are film coating, plastic molding, Marangoni convection in thin films and many others but since the final goal of the present study is to achieve a truly direct simulation of low and medium void fraction multiphase flows we will not address the problems related to these applications further. Moreover, the former application contains most of the problems for the numerical algorithms and therefore a successful algorithm developed for it can be applied in other areas as well.

Tryggvason et al. [3] classify the different types of methods for flows involving fluid–fluid interfaces into four categories: front capturing techniques, boundary-fitted techniques, Lagrangian (or arbitrary Eulerian–Lagrangian, ALE) techniques and front tracking techniques. We refer the reader to [3] for obtaining a very complete list of publications on methods in all these categories. There is a direct ideological relation between the front capturing and front tracking techniques. Therefore, we will refer to this group of methods as Eulerian (with unsteady local refinement in some cases) since it solves the Navier–Stokes equations on fixed (possibly adapted) grid unlike the Lagrangian (or ALE) methods that solve them on a grid that follows (or partially follows) the characteristics of the flow. The boundary-fitted methods are based on global coordinate transformations and therefore are not suitable for 3D flows involving more than one interface. In addition to these methods, we should also mention here the boundary integral methods that are suitable for linear (Stokes or potential) equations only and are especially useful for simulation of thin films and foams (see [4]). The technique that we discuss in the present paper belongs to the class of the Eulerian methods.

Quite surprisingly, most of the available Eulerian methods for multifluid flows employ either a finite difference or a finite volume spatial discretization. The finite or spectral elements are applied almost exclusively in Lagrangian formulations (see for example [5–7]). An Eulerian type of method for simulation of flows containing rigid particles was developed by Glowinski and collaborators [8], using the fictitious domain idea with distributed Lagrange multipliers. This idea, however, is not directly applicable to free-surface flows. As we shall try to demonstrate in this paper, the Galerkin type of methods is ideally suited for Eulerian formulations. They avoid the introduction of artificial distribution functions to account for the singular forces (surface tension for example) and naturally incorporate the interfacial boundary conditions. Moreover, they allow for a very easy correction of the interpolation near the interfaces so that the method has good mass conservation properties (a crucial feature when free boundaries are considered). One of the difficulties of the Eulerian methods is that for an accurate computation of the surface forces, it is necessary to compute the precise position of the interfaces with respect to the Eulerian grid. Since the shape of these interfaces can be very complicated, this task is very difficult (especially in 3D). Within the framework of the Galerkin formulation, it is not necessary anymore to know the intersections of the interfaces with the volume grid cells. One only needs to determine the phase to which the different integration points belong. In fact, for such an approach the numerical integration becomes a very important issue. Therefore, we developed an adaptive Newton–Cotes integration procedure which accounts for the position of the interfaces.

One of the big computational advantages of the Eulerian methods is that for simple geometries (which is the case of the direct multiphase simulations) they allow for the use of structured grids which facilitates the parallel implementation of the iterative algorithms for the linear solves. The present technique preserves this feature although it corrects locally the interpolation of the velocity and pressure in order to account for their singularity at the fluid–fluid interfaces. So, the resulting procedure is computationally effective and appropriate for large scale problems.

The remainder of the paper is organized as follows. In Section 2 we discuss the mathematical model and in Section 3 we describe the numerical technique. In Section 4 we present the results of the numerical tests and demonstrate the capabilities of this technique by solving several problems involving drops in viscous flows.

The analysis of the Eulerian methods for free-surface flows is usually quite complicated. To the best of our knowledge, there are no convergence proofs for numerical approximations of the unsteady Navier–Stokes equations in domains with free surfaces. There exist a theory for the so-called immersed interface method for elliptic problems with singularities (see [9]). The method presented in this paper has some common features with the immersed interface method. Its numerical analysis will be a subject of another study.

2. Mathematical formulation

We consider a bounded domain Ω_1 with an external boundary Γ filled with a Newtonian liquid with density ρ_1 and viscosity μ_1. Within this liquid we consider a drop Ω_2 of another liquid with density ρ_2 and viscosity μ_2. The interface between Ω_1 and Ω_2 is denoted by Σ. For simplicity of the presentation we presume the presence of only one particle but the extension to more particles is straightforward.

The equations of motion of the fluids in Ω_1 and Ω_2 are the Navier–Stokes equations (presented here in a dimensionless, stress-divergence form)

$$\frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} = \nabla \cdot \boldsymbol{\sigma}_1 + \mathbf{f}_1, \quad \nabla \cdot \mathbf{u}_1 = 0 \quad \text{in } \Omega_1, \tag{1}$$

$$\lambda \frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} = \nabla \cdot \boldsymbol{\sigma}_2 + \mathbf{f}_2, \quad \nabla \cdot \mathbf{u}_2 = 0 \quad \text{in } \Omega_2, \tag{2}$$

where $\mathbf{u}_i$ are the velocities, $\boldsymbol{\sigma}_i$ are the stress tensors and $\mathbf{f}_i$ are the external forces in Ω_i, $i = 1, 2$. In the present case $\mathbf{f}_1 = 1/(Fr)\mathbf{g}$, $\mathbf{g}$ being the unit vector in the direction of the gravity acceleration and $\mathbf{f}_2 = \lambda/(Fr)\mathbf{g}$, with $\lambda = \rho_2/\rho_1$ and the Froude number defined as $Fr = U^2/(gR)$, g being the gravity acceleration. U being the characteristic velocity and R being the radius of a sphere with the same volume as the particle. $\boldsymbol{\sigma}_i$ is defined as usual: $\boldsymbol{\sigma}_i = p_i\boldsymbol{\delta} + 2/Re_i\mathbb{D}[\mathbf{u}_i]$ with p_i being the pressure in the corresponding phase, $\mathbb{D}[\mathbf{u}_i] = 0.5[\nabla\mathbf{u}_i + (\nabla\mathbf{u}_i)^{\mathrm{T}}]$ being the rate-of-strain tensor and $\boldsymbol{\delta}$ being the Kronecker tensor. The Reynolds numbers are defined as

$$\begin{aligned} Re_1 &= \frac{UR\rho_1}{\mu_1}, \\ Re_2 &= \frac{UR\rho_1}{\mu_2} = Re_1/\eta. \end{aligned} \tag{3}$$

The boundary conditions on Γ are not of a concern for the present method and therefore we assume the simplest case of Dirichlet conditions. At the interface Σ we prescribe the well-known conditions

$$(\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2)\mathbf{n} = \frac{1}{We}\kappa\mathbf{n}, \quad \mathbf{u}_1 = \mathbf{u}_2 \quad \text{on } \Sigma, \tag{4}$$

where κ is the mean curvature of Σ and $\mathbf{n}$ is the normal pointing towards the interior of the drop. As it will be clear later, the method implicitly imposes also continuity of the tangential stress across the free boundary. The Weber number is defined as $We = U^2R\rho_1/\sigma$, σ being the surface tension coefficient. In addition to this condition that expresses the force balance on the interface, we must enforce the continuity of the velocity across the boundary which is usually done implicitly while constructing the velocity interpolation.

The interface $\Sigma(\mathbf{x},t)$ is defined as a solution of the kinematic equation

$$\frac{\partial\Sigma}{\partial t}+(\mathbf{u}\cdot\nabla)\Sigma=0. \tag{5}$$

To the best of our knowledge, the well-posedness of this problem has not been studied yet, mostly because of its high complexity. Solonnikov [12] has proved well-posedness of the problem for a steady fall of a drop in a liquid medium under the very restrictive condition that the density difference of the two fluids is very small. Desjardins and Esteban [13] proved the existence of weak solutions in the case of rigid particles in viscous liquids but the nonlinear dependence of the surface tension on the solution and the presence of the (truly) free boundary makes the present problem harder to analyze. Nevertheless, many people tried (and succeeded) in obtaining numerical solutions under much more relaxed conditions. This is the approach that we admit in the present paper too.

3. Discretization procedure

In this section we will describe the present technique through a description of the three main components of any numerical method for time dependent equations: the starting formulation (including time discretization and splitting), approximation of the solution space and eventually, a numerical integration procedure which, as we shall see further, is a very non-trivial issue in the present case.

3.1. Integration of the advective terms

In order to deal with the advection nonlinearity we split the advective part of the Navier–Stokes equations and integrate it alongside an approximation of the characteristic curves. It is a kind of a Lagrange–Galerkin procedure and its detailed description is given in [14]. The intermediate (advected) velocity is given by

$$(\tilde{\mathbf{u}}^n,\mathbf{v})_\Omega=(\mathbf{u}^n(\underline{\mathbf{x}}),\mathbf{v})_\Omega, \tag{6}$$

where n denotes the time level, $\underline{\mathbf{x}}$ is the foot of the characteristic starting at the point $\mathbf{x}\in\Omega$, $\Omega=\Omega_1\cup\Omega_2$ and $\mathbf{v}$ is the Galerkin test function to be introduced below. Usually $\underline{\mathbf{x}}$ is linearly approximated using an appropriate extrapolation for the velocity in the time interval $[t^n,t^{n+1}]$. The advantages of this method are the relatively low cost and good stability properties. The most serious problem is the computation of the integral on the right-hand side of (6). We assume that $\mathbf{u}^n(\underline{\mathbf{x}})$ is approximated using the basis of $\tilde{\mathbf{u}}^n$ and this yields a scheme with a pointwise characteristic tracking in the points of the grid. This integration is by no means exact and this causes the error to behave like $\mathrm{O}(h^k/\delta t)$ as discussed by Morton et al. [15]. This scheme is relatively cheap but, particularly in the case of pure advection, it can have unsatisfactory mass conservation property and therefore we currently consider some alternatives especially when solving for the indicator function.

3.2. Galerkin formulation and basis enrichment

If a second order backward difference scheme is used for time discretization then the remaining generalized Stokes problem with free boundaries reads

$$\begin{aligned}
&\frac{3\mathbf{u}_1^{n+1}-4\tilde{\mathbf{u}}_1^n+\tilde{\mathbf{u}}_1^{n-1}}{2\Delta t}=\nabla\cdot\boldsymbol{\sigma}_1^{n+1}+\mathbf{f}_1^{n+1},\quad \nabla\cdot\mathbf{u}_1^{n+1}=0\quad\text{in }\Omega_1,\\
&\lambda\frac{3\mathbf{u}_2^{n+1}-4\tilde{\mathbf{u}}_2^n+\tilde{\mathbf{u}}_2^{n-1}}{2\Delta t}=\nabla\cdot\boldsymbol{\sigma}_2^{n+1}+\mathbf{f}_2^{n+1},\quad \nabla\cdot\mathbf{u}_2^{n+1}=0\quad\text{in }\Omega_2.
\end{aligned} \tag{7}$$

Here, $\tilde{\mathbf{u}}^{n-i},\ i = 0, 1$, are velocity fields from time levels $n - i$, that are advected to time level $n + 1$. $\mathbf{u}_i^{n+1},\ i = 1, 2$, are subjects to the boundary conditions described above. Before we derive the weak formulation of this problem we would like to consider some properties of the solution. First of all, the strong formulation (7) requires the pressures p_i to be continuous all over Ω_i; however, they do not match each other at the interface of the particle (if the surface tension coefficient is non-zero). In a weak form, only an L^2 regularity for the pressure is required and Crouzeix–Raviart, discontinuous pressure elements can be used to approximate such solutions (see [6]). However, these elements are not usable if projection schemes (with a pressure Poisson equation) are employed because they require at least H^1 regularity for the pressure. Therefore most of the available techniques approximate the pressure jump across Σ by means of continuous interpolations which are clearly not optimal in such cases. The velocity will be continuous in the entire domain Ω but its normal derivative across the interface can be discontinuous if $\eta \neq 1$. These considerations are a consequence of the boundary condition (4) and the jump of the Reynolds number, and give us an idea of how to build approximations for these quantities. Of course, they can be approximated using standard finite elements used to solve the Stokes problem without free boundaries. The finite difference version of this approach is used in the method developed by Tryggvason and his collaborators (see for example [3]). The convergence rate of such a method, however, cannot be expected to be optimal. If the interface is aligned with element faces or the finite difference grid is constructed in similar fashion, the velocity interpolation will remain optimal but the pressure interpolation would very likely remain suboptimal. Moreover, this would require a continuous adjustment of the grid to the moving interface. This is the approach used in the Lagrangian and the Eulerian–Lagrangian techniques for solving interface problems (see [5,6]). From computational point of view, however, it is much more convenient to keep the velocity–pressure grid fixed. Then the approximation for the pressure and velocity can be corrected properly around the interface so that to account for the singularities there. This is the main idea of the immersed element method (see [9,10]). Another method developed in the same spirit is presented in [11]. The inconvenience of this type of methods is that it requires an explicit information for the intersection of the interface with the finite elements, which is very hard to compute in the 3D case, especially if higher (second) order elements for the velocity are used. In fact, if we know explicitly the intersection of the interface with a particular finite element we can simply locally refine the grid (temporarily, just for the current time step) so that it is aligned with the interface. However, this becomes a very awkward task in 3D and especially if a higher order (e.g., second) approximation is used. Therefore, we suggest here a correction to the approximation that does not require an explicit knowledge of the intersection of the interface with the elements of the Eulerian grid. Let us first define $\mathbf{u}$ and p as $\mathbf{u}|_{\Omega_i} = \mathbf{u}_i,\ p|_{\Omega_i} = p_i$. Also, throughout this section the variables with subscript h denote the approximations of the corresponding unsubscripted variables. Informally speaking we choose the pressure to be linear within each element which is not intersected by Σ. The velocity is a quadratic polynomial within such elements. In other words in elements belonging entirely to Ω_1 or entirely to Ω_2, we use the standard Taylor–Hood ($\mathbb{P}_2 - \mathbb{P}_1$) approximation. In order to ensure that the pressure is discontinuous across Σ we correct the approximation within elements that are intersected by Σ in the following fashion. We present the approximation in the 3D case, in terms of the barycentric coordinates $\mathbf{l} = (l_1, l_2, l_3, l_4),\ i = 1, \ldots, 4$, associated with a given element K. If $\mathbf{x}$ are the physical coordinates of a given point in K then $\mathbf{x} = \mathbf{T}_K(\mathbf{l})$, $\mathbf{T}_K$ being the transformation into the standard element. Then the approximation for the pressure within K, p_h, reads

$$p_h(\mathbf{l}) = \sum_{i=1}^{4} P_i l_i + \sum_{i=1}^{4} P_i^* l_i H_\Sigma(\mathbf{l}). \tag{8}$$

Here l_i are the barycentric coordinates of the ith vertex of the simplex and the function $H_\Sigma(\mathbf{l})$ is defined as

$$H_\Sigma(\mathbf{l}) = \begin{cases} 1 & \text{if } \mathbf{T}_K(\mathbf{l}) \in \Omega_1 \cap K, \\ -1 & \text{if } \mathbf{T}_K(\mathbf{l}) \in \Omega_2 \cap K, \\ 0 & \text{otherwise.} \end{cases}$$

If we compare this expansion to the usual $\mathbb{P}_1$ expansion we can see that in addition to the standard pressure unknowns p_i, it contains exactly four additional unknowns for the pressure p_i^* which clearly makes the pressure approximation discontinuous across the interface Σ between Ω_1 and Ω_2. The evaluation of the function $H_\Sigma(\mathbf{l})$ is discussed below. The velocity is enriched in a similar fashion but the basis is chosen so that the approximation has a discontinuous normal derivative across Σ. It uses the so-called bubble functions associated with the centroid C_K of the Kth element (see [16, p. 213], for more information on bubble functions). If $\hat{K}$ is the standard simplex associated with K then the bubble function is given by $\hat{b}(\mathbf{l}) = \alpha_k l_1 l_2 l_3 l_4$, where α_k is a scaling parameter. Then the approximation for the velocity within K is chosen to be (it is presented again in 3D but the simplification to 2D is obvious)

$$\mathbf{u}_h(\mathbf{l}) = \sum_{i=1}^{10} \mathbf{U}_i \boldsymbol{\phi}(\mathbf{l})_i + \mathbf{U}_C^* \hat{b}(\mathbf{l}) |D_\Sigma(\mathbf{l})|. \tag{9}$$

Here $D_\Sigma(\mathbf{l})$ is the so-called signed distance function whose value is equal to the shortest distance between the point with local coordinates $\mathbf{l}$ and the interface Σ with + or − sign, depending on the phase to which the point belongs. We shall discuss the evaluation of this function below. The basis functions $\boldsymbol{\phi}_i$ are the standard $\mathbb{P}_2$ basis functions. As it is usually done with bubble-enriched finite elements, the bubble degree of freedom can be condensed from the interpolation in order to avoid the need of updating the storage for the matrices related to the velocity degrees of freedom. However, in the present case, since we use an inner–outer (Uzawa) iteration to resolve the final linear system (see below), we simply express the bubble degree of freedom using the values of the usual $\mathbb{P}_2$ degrees of freedom at the previous iteration step. For the first step of the Uzawa iteration we use the velocity at the previous time step. Heuristically speaking, it may be expected that when the Uzawa iteration is converged to a reasonable accuracy, this approach is equivalent to the usual static condensation (of bubble degrees of freedom), but it is much easier to implement. A strict equivalence, however, has not been established and the similarity was observed only in numerical experiments.

More precisely, the functional spaces for the velocity and pressure are set up in the following way. Let $\mathscr{T}_h$ be a grid covering Ω, composed by tetrahedra (in 3D) or triangles (in 2D). Let also $\mathscr{T}_{h,\Sigma}$ be the subset of elements of $\mathscr{T}_h$ that are intersected by Σ (note that we do not require Σ to be aligned with element faces). Let us recall that T_K is the standard (isoparametric) mapping $\mathbf{T}_K : \hat{K} \to K$, and let $\mathbb{P}_2(\hat{K}) = \text{span}\{\boldsymbol{\phi}_i\}$, $\mathbb{P}_{2,\Sigma}(\hat{K}) = \{\mathbb{P}_2(\hat{K}) \oplus \text{span}\{|D_\Sigma(\mathbf{l})|\hat{b}\}\}$,$\mathbb{P}_1(\hat{K}) = \text{span}\{l_i,\ i = 1,\dots,4\}$,$\mathbb{P}_{1,\Sigma}(\hat{K}) = \text{span}\{l_i H_\Sigma(l_i),\ i = 1,\dots,4\}$. Then the approximation spaces for the velocity, $\mathbf{X}_h$, and the pressure, M_h, read

$$\mathbf{X}_h = \{\mathbf{v}_h \in \mathbf{H}_0^1(\Omega) \cap \mathbf{C}^0(\overline{\Omega}), \mathbf{v}_h|_K \circ \mathbf{T}_k \in \mathbb{P}_2(\hat{K})\ \forall K \in \mathscr{T}_h \setminus \mathscr{T}_{h,\Sigma};\ \mathbf{v}_h|_K \circ \mathbf{T}_k \in \mathbb{P}_{2,\Sigma}(\hat{K})\ \forall K \in \mathscr{T}_{h,\Sigma};\ \mathbf{v}_h|_{\partial\Omega} = 0\},$$

$$M_h = \left\{ q_h \in \mathrm{L}_2(\Omega), q_h|_K \circ \mathbf{T}_k \in \mathbb{P}_1(\hat{K})\ \forall K \in \mathscr{T}_h \setminus \mathscr{T}_{h,\Sigma};\ q_h|_K \circ \mathbf{T}_k \in \mathbb{P}_{1,\Sigma}(\hat{K})\ \forall K \in \mathscr{T}_{h,\Sigma}, \int_\Omega q_h \, \mathrm{d}\Omega = 0 \right\}.$$

Then the discrete Galerkin formulation of (7) is obtained by multiplying each of the momentum equations by an arbitrary element of $\mathbf{X}_h$, $\mathbf{v}_h$, integrating the second order term by parts, substituting the approximation for $\mathbf{u}$, $\mathbf{u}_h$, and summing the resulting variational formulations. Similarly, we multiply the continuity equations by an arbitrary element of M_h, q_h, and integrate over Ω. The unified Galerkin formulation then reads

$$\int_\Omega \rho \frac{3\mathbf{u}_h^{n+1} - 4\widetilde{\mathbf{u}_h}^{n} + \widetilde{\mathbf{u}_h}^{n-1}}{2\Delta t}\mathbf{v}_h\,\mathrm{d}\Omega + \int_\Omega \boldsymbol{\sigma}_h^{n+1} : \mathbb{D}[\mathbf{v}_h]\,\mathrm{d}\Omega = \int_\Sigma \left[\boldsymbol{\sigma}_{1,h}^{n+1} - \boldsymbol{\sigma}_{2,h}^{n+1}\right]\mathbf{n}\mathbf{v}_h\,\mathrm{d}s + \int_\Omega \mathbf{f}^{n+1}\mathbf{v}_h\,\mathrm{d}\Omega$$
$$\forall \mathbf{v}_h \in \mathbf{X}_h,\ 0 \leqslant n \leqslant N. \tag{10}$$

$$\int_\Omega \nabla\cdot\mathbf{u}_h^{n+1} q_h\,\mathrm{d}\Omega = 0 \quad \forall q_h \in M_h,\ 0 \leqslant n \leqslant N.$$

Here,

$$\boldsymbol{\sigma}|_{\Omega_i} = \boldsymbol{\sigma}_i,$$

$$\rho = \begin{cases} 1 & \text{in } \Omega_1 \\ \lambda & \text{in } \Omega_2 \end{cases}$$

and $\mathbf{n}$ is the normal to Σ pointing inside Ω_2. Incorporating the natural boundary condition at the interface we finally obtain

$$\int_\Omega \rho \frac{3\mathbf{u}_h^{n+1} - 4\widetilde{\mathbf{u}_h}^{n} + \widetilde{\mathbf{u}_h}^{n-1}}{2\Delta t}\mathbf{v}_h\,\mathrm{d}\Omega + \int_\Omega \boldsymbol{\sigma}_h^{n+1} : \mathbb{D}[\mathbf{v}_h]\,\mathrm{d}\Omega = \frac{1}{We}\int_\Sigma \kappa\mathbf{n}\mathbf{v}_h\,\mathrm{d}s + \int_\Omega \mathbf{f}^{n+1}\mathbf{v}_h\,\mathrm{d}\Omega$$
$$\forall \mathbf{v}_h \in \mathbf{X}_h,\ 0 \leqslant n \leqslant N, \tag{11}$$
$$\int_\Omega \nabla\cdot\mathbf{u}_h^{n+1} q_h\,\mathrm{d}\Omega = 0 \quad \forall q_h \in M_h,\ 0 \leqslant n \leqslant N.$$

Substituting the expansions for $\mathbf{u}_h$ and p_h, (9) and (8), and for $\mathbf{v}_h$ and q_h the basis functions for these expansions we end up with the usual form of a linear system resulting from a saddle point problem

$$\begin{aligned} &\mathbf{AU} + \mathbf{L}^{\mathrm{T}}\mathbf{P} = \mathbf{F}, \\ &\mathbf{LU} = 0. \end{aligned} \tag{12}$$

Remark 3.1. The basis enrichment as introduced above is similar, from interpolation point of view, to a local mesh refinement. Its advantage is that it avoids the explicit solution for the point of intersection of the interface Σ with the volume finite elements. It is not completely equivalent to a local refinement since the so introduced interpolation is not the same as the one that would be obtained by locally reconnecting the grid at the points of intersection. One alternative for the pressure enrichment in particular, which can improve its interpolation is to use the so-called $\mathbb{P}_2(\mathbb{P}_1 + \mathbb{P}_0)$ elements (see [20, p. 552]) only around the fluid–fluid interface. They include a discontinuous mode at the centroid of the elements thus allowing for an easy introduction of an additional degree of freedom for the pressure in each phase. An advantage of this enrichment would also be that it allows for an elementwise mass balance.

Remark 3.2. In general, the mass conservation properties of the algorithm seem to be crucial when free boundary problems are to be solved. The reason is that the free boundary advection accumulates in time any compressibility of the velocity field and if the divergence of the discrete field is high, the mass loss can become intolerable for relatively short time periods. This is demonstrated on some of the examples in the next section. Most of the existing algorithms incorporate a mechanism to compensate for the mass loss from time to time during the time integration (see for example [17]). In our opinion, it is preferable to enforce more strictly the incompressibility constraint at each time step. At present we are considering alternatives for the spatial discretization, in particular the possibility for constructing 3D divergence-free elements. Actually, our experience shows that the mass conservation is the ultimate problem for the numerical analysis of free boundary problems. This is in complete contrast to single phase flows where probably the most widely used are the approximate projection schemes which do not strictly (up to the machine accuracy) enforce the incompressibility constraint, and nevertheless produce very good results for a wide range of flows.

3.3. Front tracking

In order to compute the surface integrals in the right-hand side of (11) we indicate the position of the free boundary using second order isoparametric surface finite elements containing six points. This surface grid is advected according to (5) which is discretized using a predictor–corrector scheme. The new position is first predicted by a second order characteristic scheme. If $\mathbf{X}(t)$ is a point on the surface Σ then the predictor step reads

$$\mathbf{X}_p^{n+1} = \mathbf{X}^n + 0.5\Delta t(3\mathbf{u}_h^n - \mathbf{u}_h^{n-1}).$$

Then the velocity field at level $n+1, \mathbf{u}_h^{n+1}$, is computed using the so-predicted position of the free boundary. This velocity field is used to correct the free boundary position according to

$$\mathbf{X}^{n+1} = \mathbf{X}^n + 0.5\Delta t(\mathbf{u}_h^{n+1} + \mathbf{u}_h^n).$$

Substituting the second step into the first step, written for the next time level, we obtain the following second order characteristic scheme for advection of the free boundary

$$\mathbf{X}_p^{n+1} = \mathbf{X}_p^n + 0.5\Delta t(4\mathbf{u}_h^n - 3\mathbf{u}_h^{n-1} + \mathbf{u}_h^{n-2}). \tag{13}$$

After a long advection, some of the surface elements may become intolerably stretched. Therefore, the surface grid should be locally refined or unrefined. An algorithm for such correction of the front grid is suggested in [3] and it can be directly implemented in the present case. For the problems considered in this paper, however, this was not necessary.

3.4. Numerical integration

A major difficulty in constructing the system (12) is the computation of the volume integrals in the left-hand side of (11) and the surface integral in the right-hand side. Since the free boundary generally intersects the finite elements of the grid (which are also elementary integration volumes) the integrands in the volume integrals are discontinuous across the free interface. On the other hand, the surface elements used to track the free boundary are not aligned with faces of volume elements and therefore the integrand under the surface integral in the right-hand side has a discontinuous first derivative. Thus, the usual Gaussian quadratures used in the traditional finite element procedures are not suitable in the present case. One possibility is to determine the intersections of the free boundary with each of the finite elements cut by it and then locally refine the grid around it. However, as discussed above, the resulting algorithm would be quite sophisticated especially in 3D. Instead, we developed an adaptive Newton–Cotes integration procedure which hierarchically refines the grid around the free boundary, and thus it can achieve sufficient accuracy. First each element intersected by the free boundary is subdivided into six tetrahedra (first level of refinement, see Fig. 1), which are used to compute the volume integrals by a simple four point Newton–Cotes rule. Then the refinement procedure is applied recursively to each of the tetrahedra of the current refinement level until a convergence (with a predetermined accuracy) is achieved. Example of a 2D refinement is shown in Fig. 2. Note that it makes no sense to use higher order

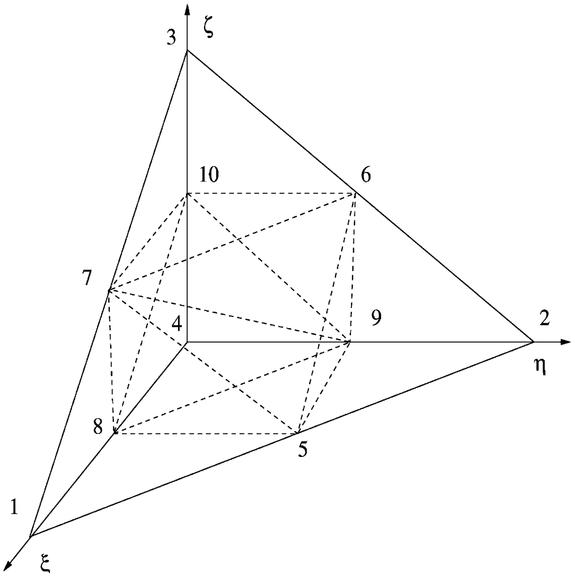

Fig. 1. First level of refinement of the standard tetrahedron.

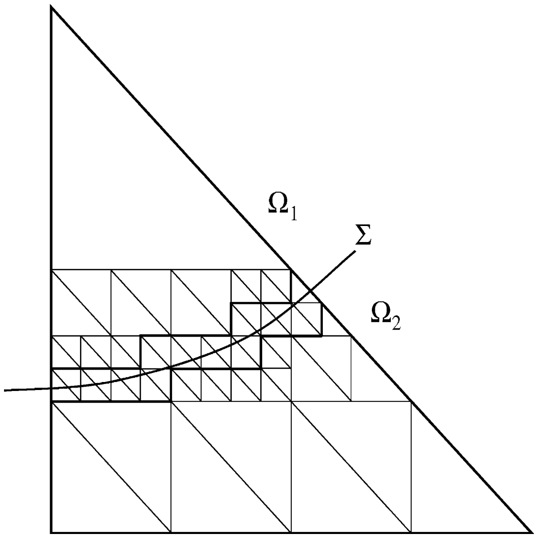

Fig. 2. Four levels of refinement of the standard triangle.

quadrature rule because of the singularities in the integrands. All the subdivisions can be done on the standard tetrahedron corresponding to the original finite element so that each refinement level simply produces a new quadrature formula for the same finite element and the same integrand, i.e., we do not change the Jacobean of the original element while doing the refinements. Although this is a very straightforward procedure it can easily become prohibitively expensive especially if many volume elements are intersected by free boundaries. However, the numerical experiments show that two levels of adaptive refinement are usually enough.

The computation of the surface force is done without explicit computation of the curvature κ. For a closed surface the surface integral can be integrated by parts and the result involves only the normal to the surface Σ

$$\frac{1}{We}\int_{\Sigma}\kappa\mathbf{n}\mathbf{v}_h\,\mathrm{d}s = \frac{1}{We}\int_{\Sigma}\mathbf{n}(\nabla\cdot\mathbf{n})\mathbf{v}_h\,\mathrm{d}s = \frac{1}{We}\int_{\Sigma}[\nabla\mathbf{v}_h - \mathbf{n}(\mathbf{n}\cdot\nabla\mathbf{v}_h)]\,\mathrm{d}s. \tag{14}$$

3.5. Mass conservation

A crucial element in the integration procedure and in determining the values of the integrands in the integration points is the way we determine whether an integration point is inside Ω_1 or Ω_2. A convenient way to indicate this is to use the distance function

$$D_{\Sigma}(\mathbf{x},t) = \begin{cases} d(\mathbf{x},\Sigma,t) & \text{if } \mathbf{x}\in\Omega_1, \\ -d(\mathbf{x},\Sigma,t) & \text{if } \mathbf{x}\in\Omega_2, \end{cases}$$

where $d(\mathbf{x},\Sigma,t)$ is the shortest Cartesian distance from the point $\mathbf{x}$ to the free boundary Σ at the time instant t. Since the initial shape of the free boundary is well known this function is easy to initialize and after that it is advected with the flow according to the advection equation

$$\frac{\partial D_{\Sigma}}{\partial t} + \nabla\cdot(D_{\Sigma}\mathbf{u}) = 0. \tag{15}$$

This is an idea borrowed from the level set method (see for example [17]). It is a very convenient indicator function in the present case too, since it can also be used for enrichment of the velocity basis as discussed above and to compute the values of H_{Σ}. At present, we solve (15) using the same characteristic method as the one used for advection of the velocity field and discussed above. This is a very cheap procedure to solve the advection equation because the search for the feet of the characteristics is to be done anyway for the advective part of the Navier–Stokes equations and this is the most time consuming part of the procedure. Of course, we could recover an indicator function from the surface grid information, as most of the surface tracking methods do (see [3] for instance). This would require an extra Poisson equation solve per time step while the advection of the indicator function does not require an extra computational effort.For long advection, however, the indicator function will loose its coordination with the surface grid. Therefore, it needs to be re-initialized from time to time (depending on the problem). One possible way to do it is discussed below. For the simulations presented in this paper this was not necessary.

If the velocity field is strictly divergence free this method would be strictly mass conservative and could be used over long time periods. However, the strict divergence freedom of the velocity field is very hard (and expensive) to be guaranteed. The same is also valid for the advection scheme for the surface grid (13). On the other hand, the mass loss (gain) accumulates in time and even small non-zero divergence of the velocity field can cause sometimes completely unphysical results over long time periods. A typical problem for which this can happen is discussed in the next section. In order to improve the mass conservation properties of the scheme we adopted the following approach. First we guarantee the divergence freedom of the velocity field at the free surface by imposing with a high accuracy the condition that $\nabla\cdot\mathbf{u}|_{\Sigma} = 0$. This is a linear constraint on the velocity field and it can be imposed via a Lagrange multiplier λ defined only at the interface Σ. Then the variational formulation of the additionally constrained problem (which is a sort of a saddle point problem) reads

$$
\begin{aligned}
&\int_{\Omega} \rho \frac{3\mathbf{u}_h^{n+1} - 4\widetilde{\mathbf{u}_h}^{n} + \widetilde{\mathbf{u}_h}^{n-1}}{2\Delta t} \mathbf{v}_h \, \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\sigma}_h^{n+1} : \mathbb{D}[\mathbf{v}_h] \, \mathrm{d}\Omega + \int_{\Sigma} \lambda_h \nabla \cdot \mathbf{v}_h \, \mathrm{d}s \\
&\quad = \frac{1}{We} \int_{\Sigma} \kappa \mathbf{n} \mathbf{v}_h \, \mathrm{d}s + \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v}_h \, \mathrm{d}\Omega \quad \forall \mathbf{v}_h \in \mathbf{X}_h, \;\; 0 \leqslant n \leqslant N, \\
&\int_{\Omega} \nabla \cdot \mathbf{u}_h^{n+1} q_h \, \mathrm{d}\Omega = 0 \quad \forall q_h \in M_h, \;\; 0 \leqslant n \leqslant N, \\
&\int_{\Sigma} \nabla \cdot \mathbf{u}_h \mu_h \, \mathrm{d}s = 0 \quad \forall \mu_h \in \Lambda_h.
\end{aligned}
\tag{16}
$$

Here μ_h are the test/trial functions for the Lagrange multipliers λ_h and Λ_h is spanned over them. They are continuous piecewise linear polynomials ($\mathbb{P}_1$ functions) defined on the surface grid marking the interface Σ. At first glance, it may seem that imposing $\nabla \cdot \mathbf{u}_h|_{\Sigma} = 0$ repeats the imposition of $\nabla \cdot \mathbf{u}_h|_{\Omega} = 0$. However, the latter condition is weaker than the former one, i.e., even if we obtain an optimal estimate for the divergence of the velocity approximation in Ω, the convergence rate for the divergence at the free boundary is worse. It is possible that no consistent estimate for it can be established. The numerical experiments show that this additional constraint significantly improves the mass conservation properties of the scheme (see next section). Using the so computed velocity field we advect the free boundary (i.e., the surface grid) and the distance function.

Since the advection of the distance function is less conservative than the advection of the free surface, after sufficiently long integration they will not be well coordinated any more and then we need to re-initialize the distance function. This can be done for example with the approach of [19]. For the numerical simulations reported in the present paper we did not need to use re-initialization but in some more extreme cases it will certainly be necessary.

Remark 3.3. Another way to deal with the singularities at the interface is to approximate the δ-function accounting for the surface tension with smooth functions (usually trigonometric approximations). This approach is especially popular among those, using the level set method (see [17]). It is not very clear, however, how such an approximation influences the accuracy of the overall scheme. Moreover, as shown above, the finite element method suggests a very natural way of treating such singular sources and therefore we preferred to track the free interface, primarily for an accurate computation of the surface integrals that arise in the weak formulation.

Remark 3.4. It is quite clear from the discussion above that the present procedure partially avoids the need of a local refinement around the interface and therefore is heavily dependent on the accuracy of the integration procedure in this part of the domain. Its advantage is that it does not require the explicit knowledge of the intersection of the surface marking grid and the volume grid. On the other hand it needs to employ a more elaborate integration procedure. Similar idea is used in [18] for modelling of crack growth.

3.6. Matrix storage and solution of the linear systems

The last problem that we need to address in this section concerns the solution of the linear system that result from the discrete formulation (16). The additional Lagrange multipliers λ_h can be considered as an additional "surface pressure" (of course their dimension is different from the one of pressure but they play a very similar role). Thus the whole problem (16) can be considered as a modification of the generalized Stokes problem. Let us denote the resulting linear system as

$$
\begin{aligned}
&\mathbf{A}\mathbf{U} + \hat{\mathbf{L}}^{\mathrm{T}}\hat{\mathbf{P}} = \mathbf{F}, \\
&\hat{\mathbf{L}}\mathbf{U} = 0,
\end{aligned}
\tag{17}
$$

where $\hat{\mathbf{L}}$ is a modification of $\hat{\mathbf{L}}$ that accounts for the surface Lagrange multipliers and $\hat{\mathbf{P}}$ is a corresponding modification of the pressure vector. From algorithmic point of view the hardest to manage is the data structure for the $\hat{\mathbf{L}}$ matrix because part of its entries should be updated at each time step since the basis is dynamically enriched. An efficient way to do that is to partition it into four parts: $\hat{\mathbf{L}}_{11}$, $\hat{\mathbf{L}}_{12}$, $\hat{\mathbf{L}}_{21}$, $\hat{\mathbf{L}}_{22}$. $\hat{\mathbf{L}}_{11}$ corresponds to the permanent degrees of freedom for both velocity and pressure and it is computed once and stored. $\hat{\mathbf{L}}_{12}$ and $\hat{\mathbf{L}}_{21}$ contain the entries linking the permanent and enriching degrees of freedom. $\hat{\mathbf{L}}_{22}$ contains the entries that link the enriching degrees of freedom for velocity and pressure (including the surface degrees of freedom). These parts are stored separately (using compact storage) and only the last three partitions are updated at each time step. For the velocity matrices this problem is avoided because of the "explicit" static condensation for the bubble degrees of freedom discussed above.

Since the divergence freedom of the velocity field is crucial for the mass conservation properties of the method the approximate projection schemes are not very suitable for they cannot control the velocity divergence level with an arbitrary accuracy (unless of course a mesh refinement is employed). Therefore we chose for the classical inner–outer iteration scheme usually called Uzawa iteration. It is well known that one needs a carefully chosen preconditioner for the outer iteration to converge reasonably fast. For the time being we chose as a pressure preconditioner the matrix $\hat{\mathbf{L}}\mathbf{M}_l^{-1}\hat{\mathbf{L}}^{\mathrm{T}}$ with $\mathbf{M}_l$ being a lumped mass matrix based on a linear approximation of the $\mathbb{P}_2$ mass matrix. For the numerical examples presented in the next section we performed only two to six outer iterations per time step and yet, the mass was very accurately conserved.

4. Numerical results

In order to verify the capabilities of the surface advection scheme we simulated the motion of a closed surface in a severe shear flow, a 3D lid-driven cavity flow. The cavity is discretized using 8^3 cubes which are subsequently divided into six tetrahedra each. A spherical surface is placed initially centred at the centre of the cavity and discretized using 320 surface triangles. The Reynolds number (based on the lid velocity) is 10, the fluid inside and outside the cavity is the same, and the surface tension is zero. The advection is performed until $t = 4$ when the quality of the front mesh becomes intolerably low. The shape of the interface at different time instants is shown in Fig. 3. Due to the limitations of the postprocessor the interface shape is drawn using linear approximation. This example clearly demonstrates that the front tracking scheme adopted in the present technique is quite stable and tolerant to very large deformations, considering the very course volume resolution. The total volume of the particle measured by

$$V = \frac{1}{3}\int_{\Sigma} \mathbf{x}\cdot\mathbf{n}\,\mathrm{d}s$$

differs from the volume of the initial sphere with 4% only, although we performed just two outer iterations at each time step for solving (17). This also clearly demonstrates the quality of the preconditioner for the pressure system.

The first test case that we selected is a trivial (from physical point of view) problem, which however is known to be tough from numerical point of view. This is the problem for relaxation of a fluid particle whose initial shape is different from spherical under a zero gravity condition. Ideally, the particle should recover its stable spherical shape and the magnitude of the velocity field should gradually go to zero. Also the particle should preserve its initial volume. The problem is quite severe numerically because the only driving force in it is the surface tension which acts as a δ-function. We simulated the relaxation of an initially ellipsoidal drop centred at the centre of the cavity given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

with $a = b = 0.25$, $c = 0.128$.

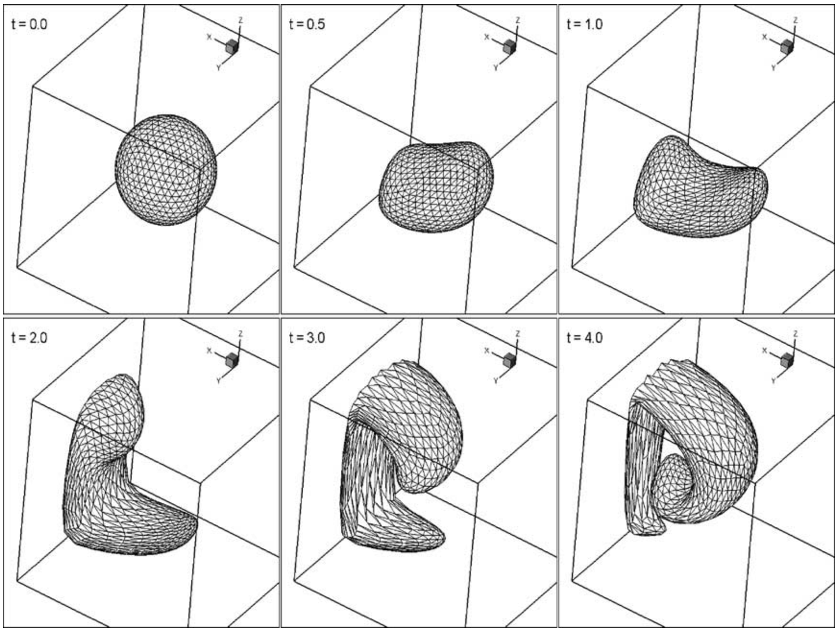

Fig. 3. Close surface advection in a driven cavity flow.

For the first set of simulations the grid contained 10^3 uniform cubes subdivided each again into six tetrahedra and the surface grid contained 320 $\mathbb{P}_2$ elements. The physical parameters of the problem were: $\rho_1 = \rho_2 = 1$, $\mu_1 = \mu_2 = 0.1$, $\sigma = 0.1$. If the characteristic length is the bubble radius R and the characteristic velocity is $U = \mu_1/(\rho_1 R)$, the dimensionless parameters are $Re = 1$, $We = 1$, $\lambda = \eta = 1$. The results for the evolution of the shape of the particle are presented in Fig. 4. The velocity field corresponding to the particle shape in the middle of Fig. 4 ($t = 0.3$) is presented in Fig. 5(a). It clearly shows the relatively rough volume grid that has been used in the present simulation. For comparison, Li and Lubkin [21] used a grid of 80×80 finite difference cells for solution of this problem under the assumption of a Stokes flow in 2D. The mass loss in the present case with the full enrichment of the basis is less than the one achieved by Li and Lubkin [21]. Of course, we pay some price for the use of second order (for the velocity) elements but still our grid contains (at the average) less than $21 \times 21 \times 21$ points as opposed to 81×81 grid points used in [21]. The present results were verified on a grid with a twice smaller stepsize in each direction. The results for the particle shape and the velocity field at $t = 0.3$, and the same parameters in the previous simulation, are presented in Fig. 5(b). The difference between the two solutions is very small and the particle shapes cannot be distinguished if plotted in the same frame. We also checked if the pressure jump was correctly accounted

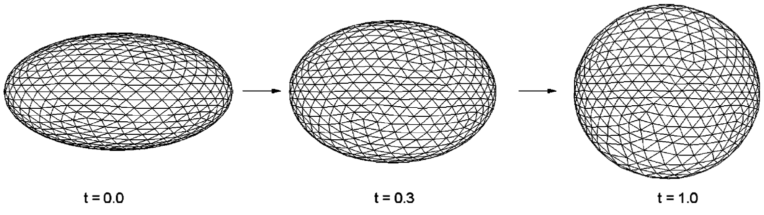

Fig. 4. Relaxation of an initially ellipsoidal particle at zero-gravity condition; the bubble shape at different time instants.

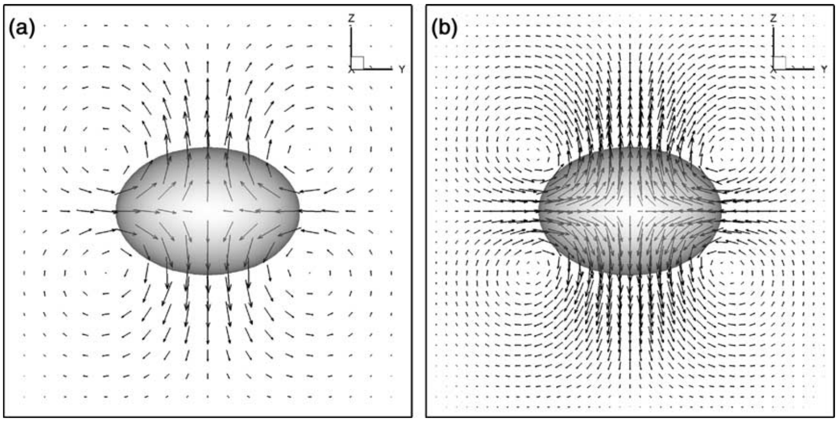

Fig. 5. Relaxation of an initially ellipsoidal particle at zero-gravity condition; the velocity field.

for, and indeed, at the final static stage, the error in pressure jump across the interface was less than 1% (on the coarser grid).

As Table 1 demonstrates, the excellent performance of the present algorithm is due to the basis enrichment. In this table we present the results for the mass loss when solving this problem without any basis enrichment, with surface enrichment only, with volume enrichment only and with both volume and surface enrichments. The results clearly advocate for the need of local improvement of the interpolation when free boundary problems are considered. Even at very small Weber number ($We = 1$) the present technique produces a result with a negligible mass loss (2.1%).

The accuracy of the present technique was also quantitatively verified using a classical solution of a free boundary problem due to Lamb (see [22]). It concerns the oscillations of the surface of a bubble in an inviscid liquid at zero gravity (Lamb used a potential theory to derive this solution). We performed two tests starting again with an initially ellipsoidal bubble ($a = b = 0.55,\ c = 0.4132$) whose volume is equal to the volume of a sphere with a radius 0.5. We resolved the problem in 1/8 of the domain imposing symmetry conditions on the planes $x = 0,\ y = 0,\ z = 0$ and the domain was defined by $0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1,\ 0 \leqslant z \leqslant 1$. The densities were chosen to be $\rho_1 = 0.01,\ \rho_2 = 1$, and the viscosities: $\mu_1 = 0.0002$ and $\mu_2 = 0.01$ The surface tension coefficients in the two cases were $\sigma = 0.5$ and $\sigma = 0.2$ correspondingly. Fig. 6 shows the amplitude of the oscillation of the surface in a point which initially was on the line $z = 0 \cap x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$. The periods of the oscillations are 1.16 and 1.83 correspondingly, and they agree well with the potential flow results of Lamb: 1.11 and 1.76. Since in both cases the density and viscosity ratios are very large, this test clearly demonstrates the capability of the present technique to handle such cases. Note that the grid used for these simulations was a uniform grid containing 10^3 uniform cubes subdivided into six $\mathbb{P}_2 - \mathbb{P}_1$ tetrahedra and no mesh refinement around the interface was performed a priori.

The ability of the present technique to simulate relatively sophisticated free boundary problems is demonstrated in the next example of a rising particle towards a fluid–fluid interface. Fig. 7 shows the

Table 1
Mass loss for the relaxation problem; $Re = 1,\ \lambda = \eta = 1$

We	Without enrichment	Surface only	Volume only	Full
5	21.6%	14.0%	4.3%	1.2%
1	40.8%	29.6%	9.6%	2.1%

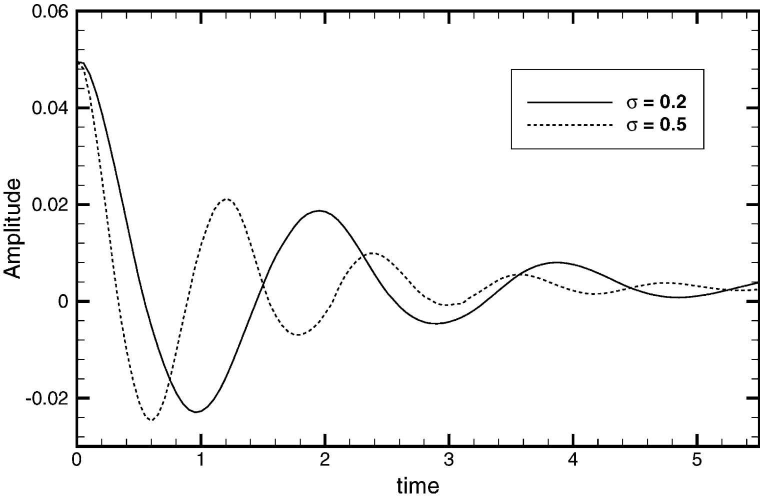

Fig. 6. The amplitude of an oscillating bubble vs. time.

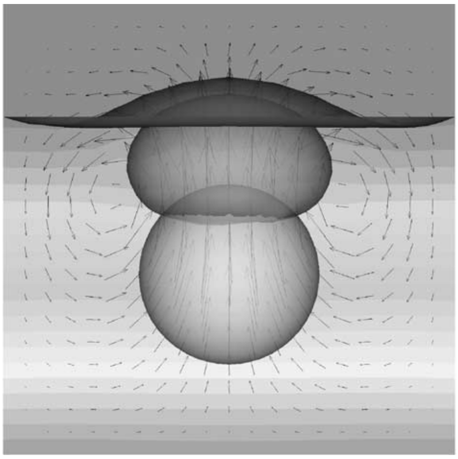

Fig. 7. Droplet rising towards a liquid–liquid interface: the fluid in the droplet and the upper fluid are the same; $Re = 1$, $We = 0.3$, $Fr = 0.076$, $\lambda = 0.005$, $\eta = 0.1$.

deformation of the two surfaces and the corresponding velocity field. The background contour shading is the total pressure distribution. The problem is simulated in 3D using the same grids as the ones in the previous simulation. If we choose the same characteristic velocity too, then the dimensionless parameters are: $Re = 1$, $We = 0.3$, $Fr = 0.076$, $\lambda = 0.005$, $\eta = 0.1$ (the fluid in the droplet and above the initially flat boundary are the same). Even using such a coarse grid we were able to simulate the droplet rise at a relatively large density and viscosity ratios of the two liquids involved. Of course, with this grid we cannot correctly resolve the film drainage phenomenon that occurs when the droplet is very close to the interface and therefore we stopped the calculations when the film thickness became comparable to the element size. But it is well known that the correct description of the film drainage requires to take into account the microphysics of this process (the van der Waals forces for example) which is not incorporated in the Navier–Stokes equations.

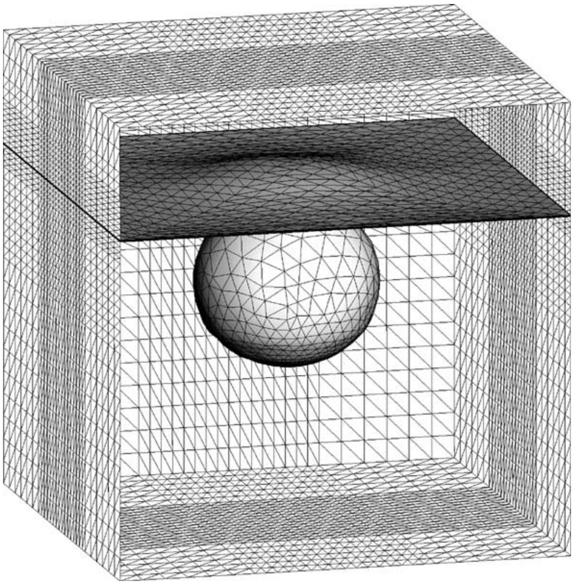

Fig. 8. Droplet rising towards a liquid–liquid interface: the volume mesh and the interface grids at time $t = 0.32$; volume mesh: $14 \times 14 \times 17$ cubes, six ($\mathbb{P}_2$) tetrahedra in each cube; drop surface: 320 ($\mathbb{P}_2$) triangles; horizontal interface: 256 ($\mathbb{P}_2$) triangles.

At the end we made a qualitative comparison for the droplet shape with an experimental result due to Hartland [23]. He investigated experimentally the coalescence of a drop with a bulk of the same liquid lying on top of another, very viscous liquid. He provided a picture of the "terminal" shape of the drop and the initially flat interface between the two liquids i.e. the shape when the long film drainage process started. The viscosity ratio of the two liquids and the capillary number in the experiment were $\eta = 0.021$ and $Ca = 1.037$. We simulated a very similar system which, with respect to the same characteristic parameters as the one in the case of the relaxation problem above, is characterized by the dimensionless

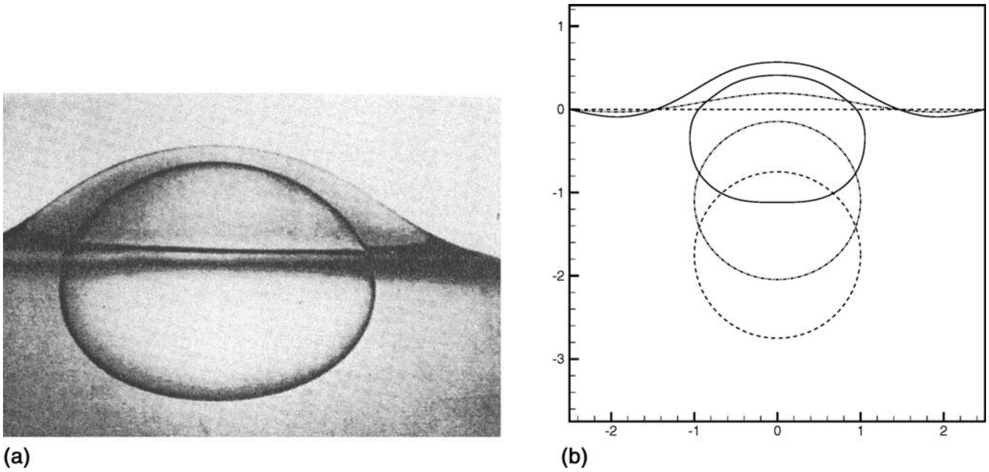

Fig. 9. Droplet rising towards a liquid–liquid interface; the fluid in the droplet and the upper fluid are the same; $Re = 1$, $We = 0.07$, $Fr = 0.022$, $\lambda = 0.5$, $\eta = 0.02$. Left: experimental picture of Hartland [23]; right: our result.

parameters $Re = 1$, $We = 0.07$, $Fr = 0.022$, $\lambda = 0.5$, $\eta = 0.02$. The surface tension force is relatively large in this case and therefore we needed to refine the grid in the vicinity of the two interfaces. The grid used in this numerical simulation with the shape of the interfaces corresponding to time $t = 0.32$ is presented in Fig. 8. [1] The experimental picture of Hartland [23] and the numerical results for the shapes of the two interfaces at time $t = 0, 0.32, 0.6$ obtained in our simulation are presented in Fig. 9. The qualitative similarity of the experimental and numerical results is quite clear. The quantitative comparison is not very representative since in the experiment the drop was released from a needle close to the interface and it was already deformed in the time of release. In our case we started with a spherical initial shape of the drop.

5. Conclusions

The finite element technique for free boundary problems which is introduced in this article should be classified as a front tracking Eulerian technique. However, it also advects a marker function which is similar to the signed distance function used by the level set methods. This is done because on one hand the finite element method incorporates the force balance boundary condition on the interface as a natural boundary condition and therefore it requires the computation of a surface integral. On the other hand, the signed distance function is a convenient indicator for the different phases and can be used for an easy enrichment of the velocity interpolation around the free boundaries. Its advection is much cheaper than the solution of a global Poisson boundary value problem (at each time step) for computing of a phase indicator as done by most existing front tracking techniques. Of course, this causes, after a sufficiently long advection, a discrepancy between the indicator and the surface grid. In such cases we suggest to re-initialize the indicator function which in the present case, unlike in the case of the level set method does not need to be strictly a distance function and therefore can be re-initialized relatively easily. Note, that one can choose to use the "pure" front tracking approach and reconstruct the indicator at each time step. On the other hand, it is also possible to advect the indicator and then account for the surface tension as in the case of the level set method (see [17]). The choice of the approach, however, will not affect the enrichment mechanism suggested in this paper and it can be used in both cases. It is beyond the scope of this study to judge which of these approaches yields better results but in our opinion, the choice is problem dependent.

The most severe problems that this (and not only this) technique faces are: (i) how to perform a conservative advection of the surface and the distance function; (ii) how to enforce efficiently and very accurately the incompressibility constraint. Of course these two problems are very intimately related. The mass conservation problem is most severe around the free boundary and it is caused by the inability of the continuous velocity/pressure approximations to accurately fit the singular solutions. Probably the best way to tackle this problem is to locally refine the grid in the vicinity of the free boundary so that the surface elements are faces of volume elements. This is a relatively easy problem in 2D if linear elements are used. But in 3D and if isoparametric elements are to be employed this is a serious problem. The alternative basis enrichment for both, the velocity and pressure interpolations, suggested in the present paper allows us to avoid this problem. The trouble then comes with the quadrature to be used for the computation of the volume and surface integrals required by the Galerkin formulation. We used an adaptive Newton–Cotes quadrature which allows to compute them with a very high accuracy. The problem (i) mentioned above still requires a more accurate solution and we are presently working on it.

[1] Because of the limitations of the postprocessor each $\mathbb{P}_2$ tetrahedron is represented by eight linear tetrahedra in the grid; similarly each triangular element is represented by four linear triangles.

Another clear advantage of the present technique is that it is probably the only available formally second order (both in space and time) Eulerian technique. This advantage is clearly demonstrated by the numerical examples that we present here. The use of second order isoparametric elements allows for accurate representation of surfaces with a complex geometry which can appear in polymer molding problems for example. At the end, since it is widely believed that the surface tracking does not allow to simulate changes in the topology of the particles, we would like to refer to [24] where it is demonstrated that with a proper remeshing this can be achieved.

Acknowledgements

This study was supported by a research grant of the National Science and Engineering Research Council of Canada.

References

[1] G. Ryskin, L.G. Leal, Numerical solution of free-boundary problems in fluid mechanics, J. Fluid Mech. 148 (1984) 19.

[2] P. Shopov, P. Minev, I. Bazhlekov, Z. Zapryanov, Interaction of a deformable bubble with a rigid wall, J. Fluid Mech. 219 (1990) 241.

[3] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708.

[4] C. Pozrikidis, Interfacial dynamics for Stokes flow, J. Comput. Phys. 169 (2001) 250.

[5] L.-W. Ho, T. Patera, A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows, Comput. Meth. Appl. Mech. Eng. 80 (1990) 355.

[6] P. Shopov, P. Minev, I. Bazhlekov, Numerical method for unsteady viscous hydrodynamical problem with free boundaries, Int. J. Numer. Meth. Fluids 14 (1992) 681.

[7] A.A. Johnson, T.E. Tezduyar, 3D simulation of fluid–particle interactions with the number of particles reaching 100, Comput. Meth. Appl. Mech. Eng. 145 (1997) 301.

[8] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2001) 363.

[9] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019.

[10] T. Lin, Y. Lin, R. Rogers, L, Ryan, A rectangular immersed finite element space for interface problems, in: Minev, Wong, Lin (Eds.), Proceedings of the Second International Workshop of Scientific Computing and Applications, Kananaskis, Canada, May 2000, Advances in Computation: Theory and Practice, vol. 7, Nova Science Publishers, Huntington, 2001, p. 107.

[11] S. Chien, D. Johnson, P. Raad, D. Fadda, The surface marker and microcell method, Int. J. Numer. Meth. Fluids 25 (1997) 749.

[12] V.A. Solonnikov, On the problem of a steady fall of a drop in a liquid medium, J. Math. Fluid Mech. 1 (1999) 326.

[13] B. Desjardins, M.J. Esteban, Existence of weak solutions for the motion of rigid bodies in a viscous fluid, Arch. Ration. Mech. Anal. 146 (1999) 59.

[14] P. Minev, C.R. Ethier, A characteristic/finite element algorithm for the 3-D Navier–Stokes equations using unstructured grids, Comput. Meth. Appl. Mech. Eng. 178 (1999) 39.

[15] K.W. Morton, A. Priestley, E. Sŭli, Stability of the Lagrange–Galerkin method with non-exact integration, Math. Model. Numer. Anal. (M^2AN) 22 (1988) 625.

[16] F. Brezzi, M. Fortin, in: Mixed and Hybrid Finite Element Methods, Springer, New York, 1991, p. 213.

[17] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comput. Phys. 124 (1996) 449.

[18] N. Moes, J. Dolbow, T. Belytchko, A finite element method for crack growth without remeshing, Int. J. Numer. Meth. Engng 46 (1999) 131.

[19] Y.R. Tsai, Rapid and accurate computation of the distance function using grids, J. Comput. Phys. 178 (2002) 175.

[20] P.M. Gresho, R.L. Sani, in: Incompressible Flow and the Finite Element Method, Wiley, Chichester, 1999, p. 552.

[21] Z. Li, S.R. Lubkin, Numerical analysis of interfacial two-dimensional Stokes flow with discontinuous viscosity and variable surface tension, Int. J. Numer. Meth. Fluids 37 (2001) 525.

[22] H. Lamb, in: Hydrodynamics, Dover, New York, 1932, p. 473.

[23] S. Hartland, The effect of the circulation patterns on the drainage of the film between a liquid drop and a deformable liquid–liquid interface, Chem. Eng. Sci. 24 (1969) 611.

[24] B. Tchavdarov, P. Minev, St. Radev, Numerical analysis of compound jet disintegration, Comput. Meth. Appl. Mech. Eng. 118 (1994) 121.