Comput Visual Sci 3: 93–101 (2000)

Computing and
Visualization in Science
© Springer-Verlag 2000

Regular article

A finite element based level-set method for multiphase flow applications

Anna-Karin Tornberg, Björn Engquist

Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden
(e-mail: annak@nada.kth.se/engquist@nada.kth.se)

Received: 1 March 1999 / Accepted: 17 June 1999

Communicated by: M. Espedal and A. Quarteroni

Abstract. A numerical method for simulating incompressible two-dimensional multiphase flow is presented. The method is based on a level-set formulation discretized by a finite-element technique. The treatment of the specific features of this problem, such as surface tension forces acting at the interfaces separating two immiscible fluids, as well as the density and viscosity jumps that in general occur across such interfaces, have been integrated into the finite-element framework. Using a method based on the weak formulation of the Navier-Stokes equations has its advantages. In this formulation, the singular surface tension forces are included through line integrals along the interfaces, which are easily approximated quantities. In addition, differentiation of the discontinuous viscosity is avoided. The discontinuous density and viscosity are included in the finite element integrals. A strategy for the evaluation of integrals with discontinuous integrands has been developed based on a rigorous analysis of the errors associated with the evaluation of such integrals. Numerical tests have been performed. For the case of a rising buoyant bubble the results are in good agreement with results from a front-tracking method. The run presented here is a run including topology changes, where initially separated areas of one fluid merge in different stages due to buoyancy effects.

1 Introduction

When multiphase flow simulations are made with the goal of including the details of interfaces separating different phases, particular difficulties arising from the treatment of these internal boundaries are present when designing a method for this purpose.

In the case of two immiscible fluids, surface tension forces act at the interfaces separating the two fluids, with the strength directly defined by the interface shape. In addition, if the two fluids have different density and viscosity, these physical quantities will have a jump in values across the interfaces. Any method designed to perform multiphase flow calculations must therefore include an accurate description of the moving and deforming interfaces, and a strategy for how to treat discontinuous physical quantities and singular surface tension forces.

Background

Since the movement and deformation of the interfaces separating the two fluids are not known beforehand, it is desirable to use a method where the background mesh does not have to adjust to these changes. Most methods designed for multiphase flow calculations are therefore based on algorithms where the background mesh is kept fixed, and the internal boundaries are represented by supplying and continuously updating some additional information.

The Marker in Cell (MAC) method [3] was an early method of this type, where a fixed number of discrete Lagrangian particles inserted are advected by the local flow. The distribution of these particles identifies the regions occupied by a certain fluid. In volume-tracking or Volume of Fluid (VOF) methods, a fractional volume function is defined to indicate the volume fraction of a certain fluid in each grid cell. An update on such methods, together with a comparison between some of them is presented in [7].

A different approach is to explicitly represent each interface with a discrete data structure, which is updated continuously to track the movement of the interface. This idea was given by Peskin [6], as he applied his immersed boundary method to calculations of blood flow in the heart. Unverdi and Tryggvason [13] applied this idea in their front-tracking method designed to simulate the motion of bubbles in a surrounding fluid.

By embedding the information of the interface locations into a function defined on the background grid, the need for a separate data structure is eliminated. The interfaces separating two fluids A and B can be represented simply as the zero level sets of a continuous function, designed to be of one sign in fluid A, and of opposite sign in fluid B. The advection of the interface is replaced by an advection of this function, and the dimensionality of the advection problem is increased with one. This is the basic idea of the level-set method, first introduced by Osher and Sethian [5]. This method has been further developed for use in many different applications, and its application to multiphase flow calculations has been described by Sussman et al. in [10] and [9]. An important property for the quality of the calculations is that the level-set function is a signed distance function, at least in regions close to any interface, carrying information about the closest distance to an interface. The level-set function is initialized as such, but as it is advected by the flow, this property will not be retained, and reinitialization is applied [10].

The level-set function can represent an arbitrary (limited by resolution only) number of bubble or drop interfaces. When merging or splitting of bubbles and drops take place, this topology change is only seen as a change in the values of this function, causing a different pattern for the zero level sets. This is the main advantage of the level-set methodology compared to the front-tracking methodology. The front-tracking method cannot be used to handle topology changes without explicit treatment of the connection and splitting of interface data structures.

The present work

In this paper, a numerical method based on a level-set formulation for incompressible two-dimensional multiphase flow is presented. The method is discretized using a finite element technique and is designed to handle surface tension forces acting at the interfaces separating the two fluids, as well as the density and viscosity jumps that in general occur across these interfaces.

In the differential formulation of the Navier-Stokes equations, the location of the singular surface tension forces are represented by a Dirac delta function. Both the front-tracking method in [13] and the level-set method in [10] was discretized using finite difference techniques, and thereby based on this formulation of the equations. In both cases, surface tension forces are smoothed by the use of a mollified delta function, as described by Peskin [6]. No such smoothing and explicit discretization of the delta function is needed with our finite element discretization. The weak formulation of the equations includes the singular surface tension forces through the evaluation of a well-defined line integral along the interface (the analogue in three dimensions would be a surface integral). Further, the differential form of the equations includes derivatives of the discontinuous viscosity. In the weak formulation, by using Green's formula, this derivative has been moved over to the test function. It is also advantageous to be able to use variable spatial resolution. The finite element technique is also well suited to perform simulations on domains of various geometrical shapes.

The finite-element integrals contain the discontinuous density and viscosity. With this motivation, the errors associated with the evaluation of integrals of discontinuous functions have been analyzed. In the aproach we have chosen, the discontinuous function is replaced by a smooth approximation before the integrals are evaluated. The integration error will consist of two parts, the analytical error made when replacing a discontinuous function by a smooth approximation, and the numerical error from the integration of this smooth function. These errors were analyzed in [11], and it was shown that vanishing moments of a certain error function are needed to obtain a small analytical error. The regularity of the smooth approximation was shown to be critical for the numerical error.

The outline of the paper is as follows: We start by formulating the problem in Sect. 2, continued by a brief discussion of the discretization of the Navier-Stokes equations. Section 3 describes the level-set method and its discretization. In the end of this section, the analysis of the errors associated with integration of discontinuous functions, performed in [11], is briefly reviewed in connection to the discussion concerning evaluation of integrals containing discontinuous physical quantities. In Sect. 5, a simulation including topology changes is presented.

2 The Navier–Stokes equations

In this presentation, we assume that we have two different fluids, fluid A and fluid B. The density and viscosity at a fixed time are given by

$$(\rho(\boldsymbol{x}), \mu(\boldsymbol{x})) = \begin{cases} (\rho_A, \mu_A) & \text{for } \boldsymbol{x} \text{ in fluid A,} \\ (\rho_B, \mu_B) & \text{for } \boldsymbol{x} \text{ in fluid B.} \end{cases} \tag{1}$$

In general $\rho_A \neq \rho_B$ and $\mu_A \neq \mu_B$, so that $\rho(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ are discontinuous at each interface separating fluid A and B. Refer to Fig. 1 for an example of a configuration of the two fluids A and B.

The equations describing this immiscible multiphase flow are essentially the Navier–Stokes equations for incompressible flow. The contribution of the surface tension forces, denoted here by f, is in addition to the gravity forces added as a source term. The equations can be written

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \nabla\cdot\left(\mu\ (\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)\right) + \boldsymbol{f} + \rho\boldsymbol{g} \tag{2}$$

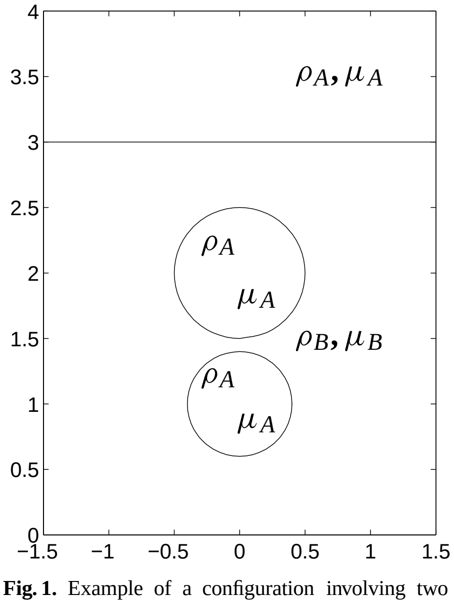


Fig. 1. Example of a configuration involving two fluids A and B, with density and viscosity $(\rho_A,\ \mu_A)$ and $(\rho_B,\ \mu_B)$

in $\Omega \subset \mathbb{R}^2$, together with the divergence-free constraint and boundary conditions,

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \subset \mathbb{R}^2, \tag{3}$$

$$\boldsymbol{u} = \upsilon \qquad \text{on } \partial\Omega, \tag{4}$$

and some appropriate initial condition $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$, where $u(x, t)$ denotes the velocity field. The pressure field is denoted by $p(x, t)$. Buoyancy effects arises from the source term $\rho\boldsymbol{g}$, where the gravitational force $\boldsymbol{g}$ is multiplied by the discontinuous density $\rho(\boldsymbol{x})$, defined in (1).

We here denote the union of all interfaces separating the two fluids by γ. In general, γ will consist of several separate segments, where each segment can either be closed or emerge from the boundary. For simplicity of description, we often refer to γ as one single segment.

The localized surface tension force f is given by

$$\boldsymbol{f} = \sigma\kappa\hat{\boldsymbol{n}}\delta_\gamma, \tag{5}$$

Here, δ_γ is a measure of Dirac delta function type with support on γ. Its action on any smooth test function φ is given by

$$\int_\Omega \delta_\gamma \, \varphi \, d\Omega = \int_\gamma \varphi \, d\gamma, \tag{6}$$

where the last quantity denotes the line integral of φ along the interface γ. This part of the expression determines the localization of the surface tension forces. The product of the other quantities, where $\sigma \in \mathbb{R}$ is the surface tension coefficient, $\kappa \in \mathbb{R}$ is the (local) curvature and $\hat{\boldsymbol{n}} \in \mathbb{R}^2$ is a normal vector to the interface γ, determines the strength and the direction of the forces. At any point along an interface, the direction of this force is towards the local center of curvature.

Proceeding to give a weak formulation of the equations, we introduce the spaces

$$\mathsf{V} = \left\{ \boldsymbol{v} \in H^1(\Omega)^2 : \boldsymbol{v} = \{v_i\}_{i=1}^2, \; v_i \in H^1(\Omega), \right\}, \tag{7}$$

$$\Pi = \{q \in L^2(\Omega) : \int_\Omega q \, d\boldsymbol{x} = 0\}, \tag{8}$$

where

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) : \frac{\partial v}{\partial x_i} \in L^2(\Omega), \; i = 1, 2 \right\}. \tag{9}$$

We define the subspaces

$$\mathsf{V}_\nu = \{\boldsymbol{v} \in \mathsf{V} : \boldsymbol{v} = \nu \text{ on } \partial\Omega\} \tag{10}$$

and

$$\mathsf{V}_0 = \{\boldsymbol{v} \in \mathsf{V} : \boldsymbol{v} = 0 \text{ on } \partial\Omega\}, \tag{11}$$

where ν is given by (4). Multiplying equation (2) by $\boldsymbol{v} \in \mathsf{V}_0$, (3) by $q \in \Pi$, and integrating over the domain using integration by parts, the following variational formulation of (2)–(4) is obtained:

Find $\boldsymbol{u}(\boldsymbol{x}, t) \in \mathsf{V}_\nu$ and $p(\boldsymbol{x}, t) \in \Pi$ such that $\forall t \in [0, T]$

$$m\left(\rho, \frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right) + \tilde{a}(\mu, \boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) + c(\rho, \boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) = f_\gamma(\boldsymbol{v}) + m(\rho, \boldsymbol{g}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathsf{V}_0 \tag{12}$$

and

$$b(\boldsymbol{u}, q) = 0 \qquad \forall q \in \Pi. \tag{13}$$

The form $\tilde{a}(\mu, \cdot, \cdot)$ is defined as

$$\tilde{a}(\mu, \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \mu \left\{ \operatorname{tr}(\nabla \boldsymbol{u}^T : \nabla \boldsymbol{v}) + \operatorname{tr}(\nabla \boldsymbol{u} : \nabla \boldsymbol{v}) \right\} d\boldsymbol{x} \tag{14}$$

where $\operatorname{tr}(\nabla \boldsymbol{u}^T : \nabla \boldsymbol{v})$ denotes the trace of the dyad product of $\nabla \boldsymbol{u}^T$ and $\nabla \boldsymbol{v}$, and similarly for $\operatorname{tr}(\nabla \boldsymbol{u} : \nabla \boldsymbol{v})$. Further, the forms $b(\cdot, \cdot)$, $c(\rho, \cdot, \cdot, \cdot)$ and $m(\rho, \cdot, \cdot)$ are defined by

$$b(\boldsymbol{v}, q) = -\int_\Omega (\nabla \cdot \boldsymbol{v}) q \, d\boldsymbol{x}, \tag{15}$$

$$c(\rho, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \int_\Omega \rho(\boldsymbol{u} \cdot \nabla \boldsymbol{v}) \cdot \boldsymbol{w} \, d\boldsymbol{x}, \tag{16}$$

$$m(\rho, \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \rho \, \boldsymbol{u} \cdot \boldsymbol{v} \, d\boldsymbol{x}. \tag{17}$$

According to definition (6), the interfacial force term $f_\gamma(\boldsymbol{v})$ evaluates as the following line integral along the interface γ,

$$f_\gamma(\boldsymbol{v}) = \sigma \int_\gamma \kappa \hat{\boldsymbol{n}} \cdot \boldsymbol{v} \, d\gamma, \tag{18}$$

The weak formulation of the equations therefore does not include δ_γ, and no explicit discretization of this two-dimensional delta function is required. The evaluation of the line integral (18) will be discussed in some detail in Sect. 3.

Discretization of the Navier–Stokes equations

Piecewise quadratic polynomials have been used for the spatial discretization. We here let W_h denote the space of piecewise polynomials of degree two on a triangular mesh, and let V_h denote the subspace of W_h with the additional constraint that any member of V_h vanishes at the boundary. Further, denote by $\mathsf{V}_h = V_h \times V_h$. For the evaluation of the integrals in the variational formulation, each element is mapped onto a "reference" element τ, where a 13-point quadrature formula (cf. Strang and Fix [8], Sect. 4.3) is used.

The particular problem to solve in each time step is determined by the time stepping scheme used, and the treatment of the incompressibility constraint. We have used a time-stepping scheme which includes a combination of implicit and explicit terms and an iteration over the nonlinearity. The discretization of the time derivative is given by the backward difference formula:

$$\frac{\partial \boldsymbol{u}}{\partial t} = \frac{3\boldsymbol{u}^{n+1} - 4\boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\Delta t} + O((\Delta t)^2), \tag{19}$$

resulting in an implicit approximation. Introduce the notation

$$G^n(\rho^{n+1}, \boldsymbol{v}) = \frac{2}{\Delta t} m(\rho^{n+1}, \boldsymbol{u}^n, \boldsymbol{v}) - \frac{1}{2\Delta t} m(\rho^{n+1}, \boldsymbol{u}^{n-1}, \boldsymbol{v}), \tag{20}$$

and

$$d(\rho, \mu, \boldsymbol{u}, \boldsymbol{v}) = \frac{3}{2\,\Delta t} m(\rho, \boldsymbol{u}, \boldsymbol{v}) + \tilde{a}(\mu, \boldsymbol{u}, \boldsymbol{v}). \tag{21}$$

Using this notation, the time stepping scheme for the discretization of the variational formulation (12) reads: Find the solution at time t^{n+1}, $\boldsymbol{u}^{n+1} \in \mathsf{V}_h$ such that

$$\begin{aligned} &d(\rho^{n+1}, \mu^{n+1}, \boldsymbol{u}^{n+1}, \boldsymbol{v}) + b(\boldsymbol{v}, p^{n+1}) + c(\rho^{n+1}, \boldsymbol{u}^{n+1}, \boldsymbol{u}^{n+1}, \boldsymbol{v}) \\ &= G^n(\rho^{n+1}, \boldsymbol{v}) + 2 f_{\gamma^n}(\boldsymbol{v}) - f_{\gamma^{n-1}}(\boldsymbol{v}) + 2m(\rho^n, \boldsymbol{g}, \boldsymbol{v}) \\ &\quad - m(\rho^{n-1}, \boldsymbol{g}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathsf{V}_h. \end{aligned} \tag{22}$$

The indecies denotes at which time level each term is defined. A fixed point iteration is introduced to resolve the nonlinearities. After each iteration, the density and viscosity fields are updated by computing a new temporary bubble position. The iteration is terminated when the difference between two consecutive velocity fields is smaller than a chosen tolerance. See [11] for more details.

The resulting problem to solve in each iteration is a Stokes problem. An iterated penalty method (described by Brenner and Scott [2]) is used to enforce the divergence-free constraint. With this method, the problem is reformulated as an iterative procedure, where the divergence of the velocity field is removed as the pressure is obtained through iterative updates. The pressure is not solved for explicitly, but can be computed from $\boldsymbol{w} \in \mathsf{V}_h$ where $\boldsymbol{w}$ is obtained from the penalty iterations. The number of unknowns in the system of equations to solve is therefore determined by the dimension of the approximation space of the velocity. The two velocity components are coupled.

The linear algebraic system obtained from the discretization is symmetric, and can be solved using a conjugate gradient method. To obtain convergence of the solution in a feasible number of iterations, the system matrix with $\rho(\boldsymbol{x}, t)$ and $\mu(\boldsymbol{x}, t)$ at one instant in time is assembled, factored and used as a preconditioner. It is kept as a preconditioner until the number of iterations needed for convergence exceeds a certain limit. At this point, the process is repeated.

3 The level-set method

In order to complete the problem formulation initiated in the previous section, we need to define how the interfaces are represented and how their evolution are determined. Further, it must be defined how to determine which fluid region that contains a point $\boldsymbol{x}$, such that $\rho(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ can be defined from (1). Finally, the evaluation of the contribution of the surface tension forces in the variational formulation, as given by (18), needs to be discussed.

In the level-set method, the interfaces are represented as the zero level sets of a continuous function, designed to be of positive sign in fluid A, and of negative sign in fluid B. The level-set function ϕ is initialized as a signed distance function, carrying information about the closest distance to any interface separating the two fluids.

To determine the evolution of the interfaces, the level-set function ϕ is simply advected by the flow, when the velocity field $\boldsymbol{u}$ is known. The advection is given by

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0. \tag{23}$$

The property of ϕ being a distance function is not preserved during advection. It has however been shown [10] that it is critical that ϕ remain a distance function in regions close to any interface, in order to obtain acceptable conservation of mass. A reinitialization procedure is applied as part of the calculations in order to restore this property of the level-set function in regions close to a zero level set.

The following equation was introduced by Sussman et al. [10]:

$$\frac{\partial \psi}{\partial \tau} = S(\psi_0)\ (1 - |\nabla \psi|) \tag{24}$$

$$\psi(\boldsymbol{x}, 0) = \psi_0(\boldsymbol{x}), \tag{25}$$

where $S(t)$ is a sign function, given by

$$S(t) = \begin{cases} -1 & \text{for } t < 0, \\ 0 & \text{for } t = 0, \\ 1 & \text{for } t > 0, \end{cases} \tag{26}$$

Let $\psi_0 = \phi(t^*)$ be the level set function obtained in any time step. It will be somewhat distorted from a distance function. By solving equation (24) to steady state, a distance function ψ that has the same zero level set as ψ_0 is obtained. Alternatively, equation (24) can be written as

$$\frac{\partial \psi}{\partial \tau} + \boldsymbol{w} \cdot \nabla \psi = S(\psi_0), \qquad \boldsymbol{w} = S(\psi_0) \frac{\nabla \psi}{|\nabla \psi|}. \tag{27}$$

That is, $\boldsymbol{w}$ is a unit vector, always pointing away from the interface, and information is propagated out from the zero level sets with a speed of one. In practice, the reinitialization need not to be done until a steady state is reached, since it is only of interest that we obtain a distance function in the neighborhood of the zero level set.

The equations (23) and (27) are discretized using piecewise quadratic polynomials, as the Navier–Stokes equations. In order to stabilize the equations, the streamline diffusion method has been used. After each advection step, we perform M reinitialization "time"-steps, in artificial time τ. Including the streamline diffusion modification, formulating an explicit first order method in time, this reads: Given $\boldsymbol{u} \in \mathsf{V}_h$ and $\phi^{n-1} = \phi(t_{n-1}) \in W_h$, find $\tilde{\phi}^n \in W_h$ such that

$$\left(\frac{\tilde{\phi}^n - \phi^{n-1}}{\Delta t}, v \right) + \left(\boldsymbol{u} \cdot \nabla \phi^{n-1}, v + \delta\, \boldsymbol{u} \cdot \nabla v \right) = 0 \quad \forall v \in W_h. \tag{28}$$

Set $\psi^0 = \psi_0 = \tilde{\phi}^n$, and for $m = 1, ..M$, solve

$$\begin{aligned} &\left(\frac{\psi^m - \psi^{m-1}}{\Delta \tau}, v \right) + \left(\epsilon \nabla \psi^m, \nabla v \right) = \\ &\quad - \left(\mathbf{w}^{m-1} \cdot \nabla \psi^{m-1}, v + \delta\, \mathbf{w}^{m-1} \cdot \nabla v \right) \\ &\quad + \left(S_\alpha(\psi), v + \delta\, \mathbf{w}^{m-1} \cdot \nabla v \right) \quad \forall v \in W_h. \end{aligned} \tag{29}$$

Set $\phi^n = \psi^M$.

The parameter δ is chosen as

$$\delta = \frac{1}{2\sqrt{(\Delta t)^{-2} + |\boldsymbol{u}|^2 h^{-2}}} = \frac{1}{2\sqrt{(\Delta t)^{-2} + |J^{-1}\boldsymbol{u}|^2}} \tag{30}$$

where J^{-1} is the inverse Jacobian for the mapping from the reference element to the element in physical coordinates.

For numerical purposes, the sign function $S(\psi)$ has been replaced by a smooth approximation $S_\alpha(\psi)$ given by

$$S_\alpha(\psi) = \frac{\psi}{\sqrt{\psi^2 + \alpha^2}}. \tag{31}$$

The parameter α is on the order of the grid size. Because of this, $\boldsymbol{w}$, defined as $\boldsymbol{w} = S_\alpha(\psi_0)\nabla\psi/|\nabla\psi|$ will in practice not be a unit vector inside the smoothed zone of the sign function, where its magnitude will be given by $|S_\alpha(\psi_0)| < 1$.

In the reinitialization equation (29), extra numerical diffusion has been added ($\epsilon > 0$). This is needed to stabilize the calculations, since the streamline diffusion modification gives an insufficient diffusion effect close to the zero contour, where $S_\alpha(\psi_0)$ is small, and where $\boldsymbol{w}$ therefore is small in magnitude. This modification however negatively affects the conservation of the area fractions of fluid A and B, defined by the positions of the zero level sets.

Natural boundary conditions are imposed for the advection equation. Outgoing characteristics of the reinitialization equation keep the inflow boundaries free from disturbances. In the cases where $\boldsymbol{w}$ does not point outwards at the boundaries, a small modification to $\boldsymbol{w}$ is added close to the boundary, to ensure that information is propagated out of the domain.

The most time consuming part of the calculations is the solution of the Navier–Stokes equations. If higher resolution of ϕ is wanted, in order to resolve small scales better in a merging process or to increase the quality of the curvature calculations, it yields much less extra work to only increase the resolution in the advection and reinitialization calculations while retaining the same resolution in the solution for the velocity field, compared to increasing the resolution for the full problem.

In addition to the mesh on which we solve the Navier–Stokes equations, and thereby define the velocity field, we define a refined mesh, which is obtained by regular subdivision of the first mesh, i.e. by splitting each element into four sub-elements. The advection and reinitialization of the level set function ϕ is performed on this refined mesh. This refinement of the ϕ calculations also yields a more accurate evaluation of the force term $f_\gamma(\boldsymbol{v})$.

Evaluating the interfacial force term $f_\gamma(\boldsymbol{v})$

For evaluation of the interfacial force term in (12),

$$f_\gamma(\boldsymbol{v}) = \sigma \int_\gamma \kappa \hat{\boldsymbol{n}} \cdot \boldsymbol{v}\, d\gamma, \tag{32}$$

the segments of the interface γ are defined from the level-set function ϕ. The evaluation is a local process, performed element by element in the finite element mesh. An element will yield a non-zero contribution to this term only if some part of a zero level set of $\phi(\boldsymbol{x})$ (i.e a segment of γ) is intersecting the element. Each element is divided into four sub-elements, and the zero level set of ϕ is defined by a linear approximation on this sub-element scale.

The curvature κ and normal vectors $\hat{\boldsymbol{n}}$ can be calculated as

$$\hat{\boldsymbol{n}} = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa = -\nabla\cdot\hat{\boldsymbol{n}}. \tag{33}$$

The unit normal vector $\hat{\boldsymbol{n}}$ always point into fluid A (where $\phi > 0$). The surface tension coefficient σ is a positive coefficient, and the direction of the surface tension force is determined by the sign of κ along the interface γ. When ϕ is an exact distance distance function, it holds that $|\nabla\phi| = 1$.

As ϕ is advected, the discontinuities of its gradient will create some high frequency dispersion errors of a small magnitude. All oscillating components of the error will be magnified when derivatives are calculated, a fact that complicates the curvature calculations. To avoid this magnification, the high frequencies in ϕ is filtered out, and (33) is applied to that filtered function. We calculate

$$(\tilde{\phi}, v) + (\epsilon\nabla\tilde{\phi}, \nabla v) = (\phi, v) \quad \forall v \in W_h, \tag{34}$$

and from here,

$$\hat{\boldsymbol{n}} = \frac{\nabla\tilde{\phi}}{|\nabla\tilde{\phi}|}, \quad \kappa = -\nabla\cdot\hat{\boldsymbol{n}}. \tag{35}$$

This procedure effectively filters out high frequencies from ϕ to create $\tilde{\phi}$. The function $\tilde{\phi}$ is only used in this intermediate step in the curvature calculations, and is not used elsewhere.

Discontinuous density and viscosity

The density and viscosity fields are easily defined in terms of the level set function ϕ, since ϕ is of different sign in the two fluids. The characteristic function $H(\phi(\boldsymbol{x}))$ is 1 in fluid A and 0 in fluid B, and we can define

$$\rho(\boldsymbol{x}) = \rho_B + (\rho_A - \rho_B)\ \ H(\phi(\boldsymbol{x})), \tag{36}$$

$$\mu(\boldsymbol{x}) = \mu_B + (\mu_A - \mu_B)\ H(\phi(\boldsymbol{x})), \tag{37}$$

where $H(t)$ is the Heaviside function,

$$H(t) = \begin{cases} 0 & \text{for } t < 0, \\ 1/2 & \text{for } t = 0, \\ 1 & \text{for } t > 0, \end{cases} \tag{38}$$

The variational formulation (12) contains some integrals with the discontinuous density or viscosity as a factor of the integrand. For the integration of a function over a triangulated domain, the integral is written as a sum of integrals over each triangle. The quadrature errors for a general smooth function is kept small, using the 13-point quadrature rule (cf. Strang and Fix [8], Sect. 4.3) on each triangle. Discontinuous functions are however not well approximated by polynomials of any order and the quadrature error will be large if special care is not taken.

There are two main approaches for the evaluation of integrals of discontinuous functions: *i)* Keeping the characteristic function discontinuous, altering the quadrature procedure, and *ii)* to replace the characteristic function by a smooth approximation. In [11], both these approaches were studied. Here, some results for the second approach are briefly reviewed.

All functions discontinuous across the curve described by $\phi = 0$ can be written as the sum of a continuous part and a discontinuous part given by $F(\boldsymbol{x}) = G(\boldsymbol{x})H(\phi(\boldsymbol{x}))$, where $G(\boldsymbol{x})$ is a smooth function.

We replace the Heaviside function $H(\phi)$ by a smooth approximation $H_w(\phi)$, given by

$$H_w(\phi) = \begin{cases} 1 & \phi > w \\ \nu(\phi/w) & |\phi| \le w \\ 0 & \phi < -w \end{cases} \tag{39}$$

where $\nu(\xi)$ is a smooth transition function such that $\nu(-1) = 0$ and $\nu(1) = 1$.

The total error in the integration of $H(\phi(\boldsymbol{x}))\, G(\boldsymbol{x})$ can be divided in two parts, the analytical error $E_{w,G}$ made when replacing $H(\phi(\boldsymbol{x}))$ by its smooth approximation $H_w(\phi(\boldsymbol{x}))$, and the numerical error $E_{\mathsf{quad},G}$ made in the integration of $H_w(\phi(\boldsymbol{x}))$. Let quad$(F(\boldsymbol{x}))$ denote the approximation to the integral $\int_\Omega F(\boldsymbol{x})\, d\boldsymbol{x}$ obtained by summation of element integrals evaluated according to the quadrature rule. The total error in the integration of $H(\phi(\boldsymbol{x}))\, G(\boldsymbol{x})$ can be written

$$E_{\mathsf{tot},G} = E_{w,G} + E_{\mathsf{quad},G}, \tag{40}$$

with

$$E_{w,G} = \int_\Omega H(\phi(\boldsymbol{x}))\, G(\boldsymbol{x})\, d\boldsymbol{x} - \int_\Omega H_w(\phi(\boldsymbol{x}))\, G(\boldsymbol{x})\, d\boldsymbol{x}$$

$$E_{\mathsf{quad},G} = \int_\Omega H_w(\phi(\boldsymbol{x}))\, G(\boldsymbol{x})\, d\boldsymbol{x} - \mathrm{quad}(H_w(\phi(\boldsymbol{x}))\, G(\boldsymbol{x}))$$

In general, the numerical error is large for w small, and decreases with w, while the analytical error increases as w gets larger. Both $E_{w,G}$ and $E_{\mathsf{quad},G}$ depend on the particular choice of the transition function $\nu(\xi)$.

The analytical error $E_{w,G}$ is obtained by integrating the error function $E(\boldsymbol{x}) = (H(\phi(\boldsymbol{x})) - H_w(\phi(\boldsymbol{x})))\, G(\boldsymbol{x})$ over Ω. Since $E(\boldsymbol{x}) = 0$ for $\boldsymbol{x}$ such that $|\phi(\boldsymbol{x})| > w$, we parameterize the region where $|\phi(\boldsymbol{x})| \le w$ to perform the integration of $E(\boldsymbol{x})$. Assume that the zero contour of ϕ can be parameterized by $\gamma = (x(s), y(s))$, where $s \in [0, 2\pi]$, and further that $w \max_s |\kappa(s)| < 1$, where $\kappa(s)$ is the curvature of γ. Then the integral over $E(\boldsymbol{x})$ can be transformed to an integral in the parameters (s, t), and the analysis can be performed in these coordinates.

We define the error function $E(t) = H(t) - H_w(t)$ and introduce what we henceforth call the moments of $E(t)$,

$$M_\alpha(E(t)) = \int_{-w}^{w} E(t)\, t^\alpha\, dt = w^{\alpha+1} \left\{ \frac{1}{\alpha+1} - \int_{-1}^{1} \nu(\xi)\, \xi^\alpha\, d\xi \right\}, \tag{41}$$

where $\nu(\xi)$ is the transition function in the definition of $H_w(\phi)$ (39).

One can show that the analytical error $E_{w,G}$ is proportional to w^β, where β is related to the number of vanishing moments of the error function. Choosing the transition function $\nu(\xi)$ such that the moments of the error function, $M_\alpha(E(t))$, evaluates as zero for $\alpha = 0, \ldots, m$, yields a leading term for the analytical error proportional to w^β, with $\beta = m + 2$. If $\nu(\xi)$ is chosen as $\nu(\xi) = 1/2(1 + \tilde{\nu}(\xi))$, where $\tilde{\nu}(\xi)$ is an odd function, all even moments of the error function will vanish in addition to the m first moments. In this case we have $\beta = 2\lfloor \frac{m+1}{2} \rfloor + 2$, and the next term in the error expansion will be proportional to $w^{\beta+2}$. For the details of this analysis, with complete theorems and proofs, see [11].

The coefficient in the error expression depends on derivatives of the function $G(\boldsymbol{x})$, and this analysis shows further that if $G(\boldsymbol{x})$ is constant, any approximation of the Heaviside function with at least two vanishing moments of the error function will introduce no analytical error. The numerical error is in this case the only source of error. Assuming that w/h, the ratio of the width of the transition zone to the size of the quadrature triangles, is not too small, this error is predicted to show the following dependence:

$$E_{\mathrm{quad}}^{m,k} \sim \frac{h^{k+2}}{w^{k+1}}, \tag{42}$$

k being the number of continuous derivatives of $H_w(t)$, simply given by the number of vanishing derivatives of $\nu^{m,k}(\xi)$ at $\xi = \pm 1$. This is valid assuming that $k < n$, where n is the order of the quadrature rule used. This can be verified by numerical experiments, see [11].

In conclusion, the choice of the transition function $\nu(\xi)$ determines the order of the analytical error (in powers of the width of the transition zone w) by the number of vanishing error moments of the corresponding error function, and the order of the numerical error by its number of vanishing derivatives at $\xi = \pm 1$. Choosing $\nu(\xi)$ as a polynomial, we have proven [11] that the lowest degree polynomial, obeying $\nu(-1) = 0$, $\nu(1) = 1$, with k vanishing derivatives at $\xi = \pm 1$, defined to yield m vanishing moments of the corresponding error function, exists and is uniquely determined. It is of degree $r = 2\lfloor (m+1)/2 \rfloor + 2k + 1$. Further, we have that $\nu(\xi) = 1/2 + p(\xi)$, where $p(\xi)$ is a polynomial of degree r, containing only odd powers of ξ.

The transition function used in the calculations in this paper is the fifth order polynomial

$$\nu(\xi) = \frac{1}{2} + \frac{1}{32}\,(45\,\xi - 50\,\xi^3 + 21\xi^5). \tag{43}$$

The corresponding Heaviside approximation $H_w(t)$ has one continuous derivate ($k = 1$), and two vanishing moments of its error function ($m = 2$).

4 The algorithm in summary

Below, we present a short summary of the algorithm in order to emphasize the overall structure.

Initially, assume that the level-set function ϕ, is known either from initial conditions or the previous time step. The steps are then as follows:

i) The density and viscosity fields (36) are defined with the characteristic function $H(\phi(\boldsymbol{x}))$ replaced by a smooth approximation $H_w(\phi(\boldsymbol{x}))$ (39), (43). The contribution of the surface tension forces $\boldsymbol{f}_{\gamma}(\boldsymbol{v})$ is evaluated using the level-set function ϕ, using a high frequency filtering in the curvature calculations (35).

ii) Solve the time discretized Navier–Stokes equations through an iterative procedure, to obtain the velocity $\boldsymbol{u}$ at the next time level. The divergence-free constraint is enforced using an iterative penalty method. The resulting symmetric system of linear equations is solved using a preconditioned conjugate gradient method. The contributions from *i)* are needed in these calculations.

iii) Advect the level set function ϕ by the velocity field $\boldsymbol{u}$ (28). Do one or more reinitialization step as defined in (29), to restore the distance function property in the region close to the zero level set.

iv) This completes one time step, repeat *i)–iii)* to advance further.

Some comments regarding *ii)*: The solution to the Navier–Stokes equations is in each time step obtained by performing fixed-point iterations, until convergence is reached. After each fixed-point iteration, a temporary advection of the level set function ϕ needs to be done, in order to update the density and viscosity fields included in the left hand side of the equation.

The basic tools for using high-order finite elements are available in the library `Albert` [1]. Many of the features used here, for example the streamline diffusion modification and evaluation of the interfacial force term are however not available in this library.

5 Numerical simulations

The runs we have performed includes the rise of buoyant bubbles. The non-dimensional parameters used to characterize the problems are, in addition to the viscosity and density ratios, μ_B/μ_A and ρ_B/ρ_A, the Morton number M and the Eötvös number Eo. These numbers are defined as

$$M = \frac{g\mu_B^4}{\rho_B\sigma^3}, \quad Eo = \frac{\rho_B g d^2}{\sigma}. \tag{44}$$

Here, μ_B and ρ_B denote the viscosity and density of the outer fluid, respectively, g is the gravitational constant and σ is the surface tension coefficient. $d = 2\sqrt{A/\pi}$, where A is the area of the bubble.

In the case of a single buoyant bubble, the simulations show good agreement with results from the front-tracking method presented in [12]. The level-set method does not conserve the mass (or area, since the density inside the bubble is constant) as well as the front-tracking method. In a typical computation, the decrease compared to the initial area for the level-set method is 1.0%, compared to 0.01% for the front-tracking method, see [11]. The reason for this is the numerical diffusion present in the level-set calculations. The main part of this artificial viscosity comes from the stabilization of the reinitialization procedure. The artificial diffusion needed to stabilize the equations decreases as the spatial resolution is increased, which results in improved mass conservation with reduction in mesh size. The front-tracking method is simple and accurate for the case with a single buoyant bubble. It is however not well suited for simulations including topology changes, since this cannot be done without explicit treatment of the connection and splitting of interface data structures.

We now present a level-set based simulation including topology changes. In this run, we start with two bubbles of fluid A immersed into fluid B, with a layer of fluid A on top of fluid B. The initial configuration is as in Fig. 1, with the domain extended up to $y = 6.0$. A contour plot of the initial level-set function is shown in Fig. 2.

The diameter of the largest bubble (the upper one) is normalized to 1. The diameter of the smaller bubble is 0.8. The Morton number and Eötvös numbers given for the flow are based on the largest bubble.

The fluid is quiescent at $t = 0$. The bubbles will rise in the middle of the domain, and we need the highest resolution in this part of the region, as well as close to the surface. We use the irregular mesh shown in Fig. 3 for this simulation.

We present the results from a run with $Mo = 0.1$ and $Eo = 10.0$ with density ratio $\rho_B/\rho_A = 100$ and viscosity ratio $\mu_B/\mu_A = 2$. The numerical parameters are as follows: $\Delta t = 5 \times 10^{-4}$. In each advection time step one reinitialization step with time step $\Delta\tau = 0.01$ is performed. The diffusion parameter in this procedure is set to $\epsilon = 4 \times 10^{-3}$. The smoothing parameter in the Heaviside approximation (39) is set to 0.05, and the smoothing parameter for the sign function in (31), is set to 0.1.

We can study the buoyancy induced motion of the two fluids that are quiescent initially. The two bubbles (of fluid A) will start to rise since they are lighter than the surround-

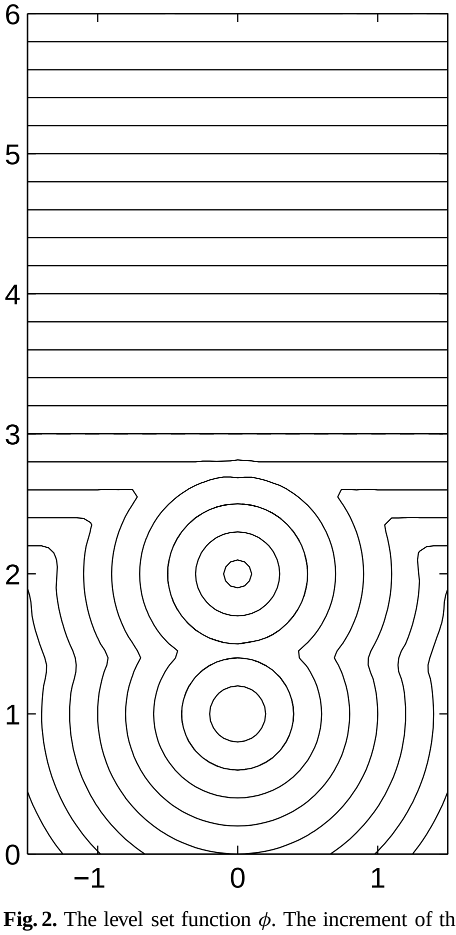


Fig. 2. The level set function ϕ. The increment of the contours is 0.2

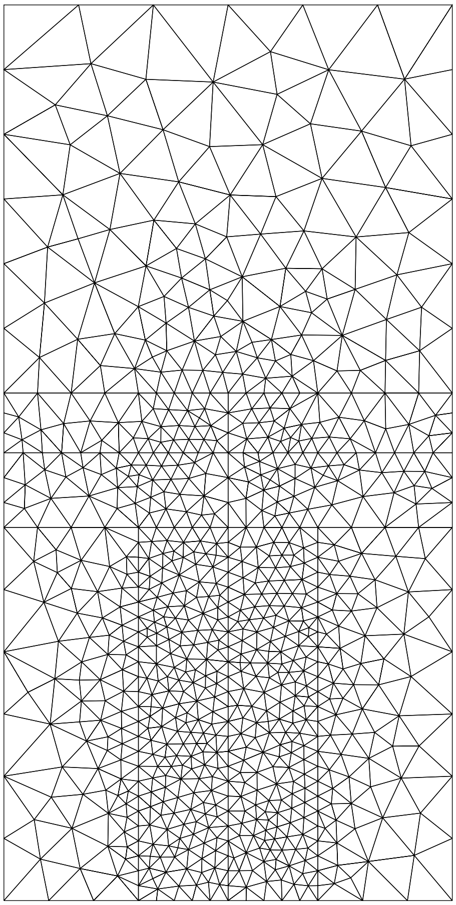


Fig. 3. The mesh used in the simulations. 1566 elements, with 6 nodes in each element

ing fluid. The smaller bubble travels in the wake of the upper, larger bubble, and will rise faster. When it catches up with the larger bubble, the two bubbles will merge. This is shown in Fig. 4. The plots do not cover the top part of the domain. The merged bubble is deforming as it moves closer to the surface. It pushes the surface upwards and the drainage of fluid B from the region between the two interfaces starts.

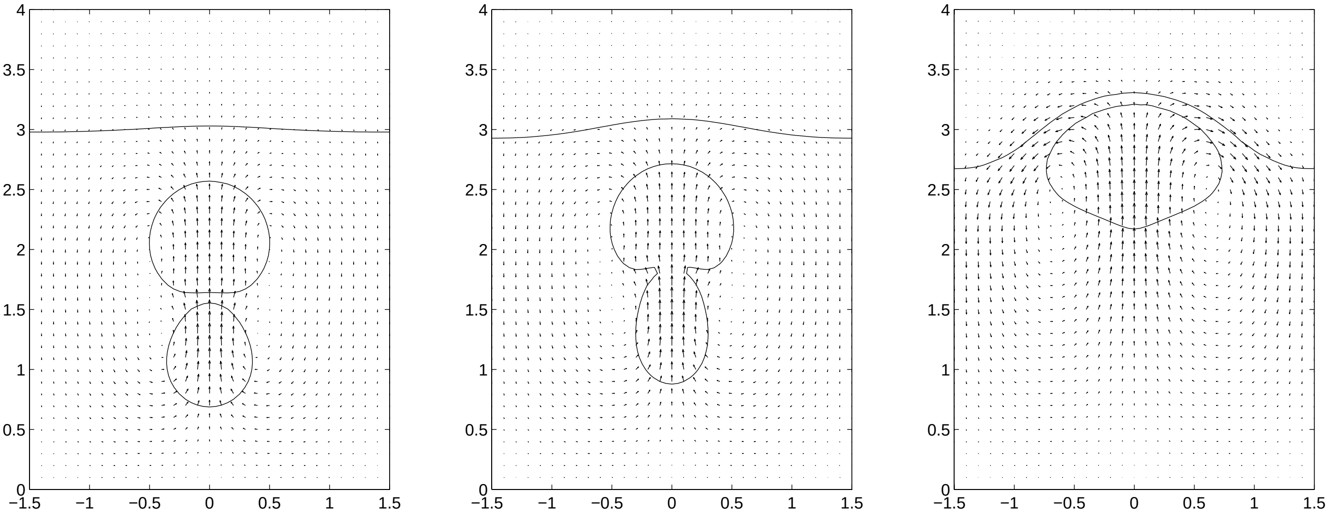
Finally, the filament between the two interfaces gets so thin that the bubble merge with the surface. This is shown in Fig. 5.

Two thin filaments of fluid A are pointing into fluid B after the merge. Local high velocities develop here to smooth the surface out, as can be seen in Fig. 5b. In this process, the surface gets pushed up in the middle from the recirculation of fluid. Note that the flow above the surface have changed direction in Fig. 5c, compared to Fig. 5b, so that the surface is pushed down to a flat surface again. We can note that the flow is slowing down. The final steady state will be a zero velocity field ($\boldsymbol{u} = 0$), with fluid A on top of fluid B, the two fluids separated by a flat surface.

The curvature calculations are complicated at the point where the two bubbles have merged (Fig. 5a), and are not very accurate at the interface along the two thin filaments. The curvature calculated from (35) is cut off at a maximum value of 15.0. This maximum value is motivated from the fact that structures with larger curvatures, i.e. with such small scale details, can not be represented with the resolution of the present mesh. The high frequencies of this cut curvature is thereafter filtered out equivalent to (34).

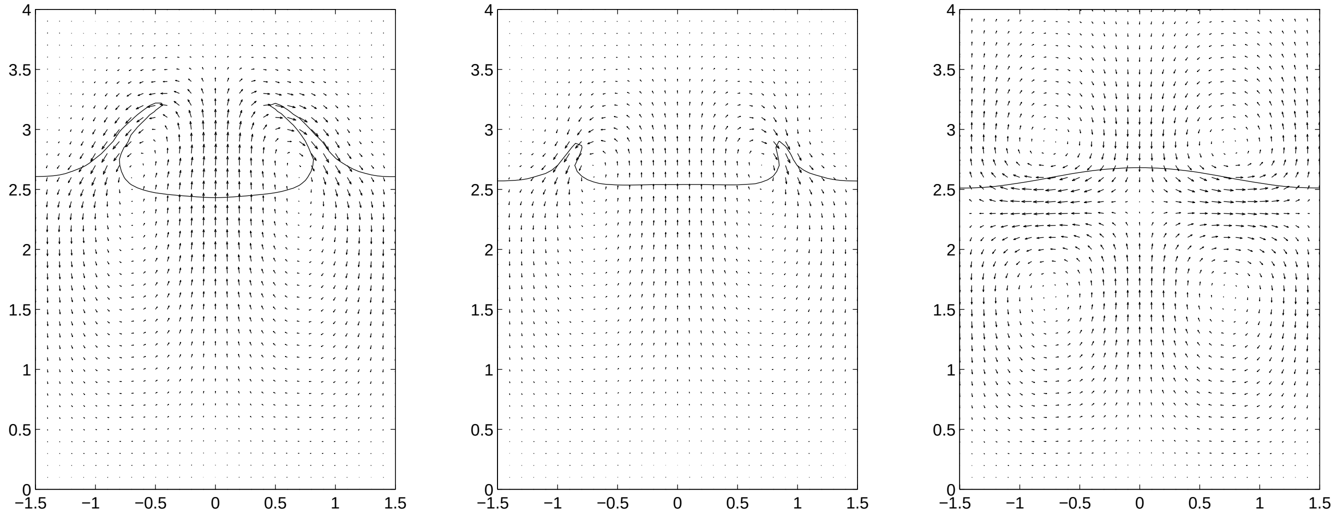
The area fractions of fluids A and B are not conserved during the simulation. The relative change in the area fraction of fluid A is plotted versus time in Fig. 6. The area fraction decreases at first, but increases some at later times. The area fraction of fluid A at time $t = 0.55$, the instant plotted in Fig. 5c, is 99.3% of the initial area fraction.

We have here shown the abilities of the level-set method in performing simulations where topology changes occurs. No specific treatment is needed when a merging takes place. The exact time at which merging will occur in a simulation for a fixed set of physical parameters will be however be affected by the resolution, i.e how small scales that can be resolved, and by the amount of artificial diffusion present in the calculations. This is however a process converging to a certain solution as the resolution is increased and the numerical diffusion is decreased.



a) $t = 0.05$. $Re_\infty = 5.52$. **b)** $t = 0.1$. $Re_\infty = 7.65$. **c)** $t = 0.3$. $Re_\infty = 5.07$.

Fig. 4. The two bubbles merge. The interfaces plotted together with the corresponding velocity field



a) $t = 0.3625$. $Re_\infty = 3.84$. **b)** $t = 0.40$. $Re_\infty = 5.064$. **c)** $t = 0.55$. $Re_\infty = 0.86$.

Fig. 5. After the last merge. The interface plotted together with the corresponding velocity field

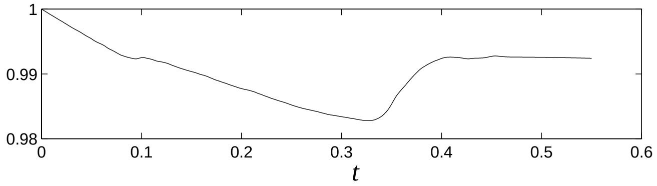


Fig. 6. The relative change in area fraction of fluid *A* plotted versus the non-dimensional time *t*

References

1. The Albert system is available in the compressed `tar` file fec.tar.gz which can be obtained from `http://www.hpc.uh.edu/~babak/fec.tar.gz`.
2. Brenner, S.C., Scott, L.R.: The Mathematical Theory of Finite Element Methods. Springer 1994
3. Harlow, F.H., Welch, J.E.: Volume-Tracking Methods for Interfacial Flow Calculations. Phys. Fluids 8: 2182 (1965)
4. Johnson, C.: Numerical Solution of Partial Differential Equations by the Finite Element Method. Lund, Sweden and Cambridge: Cambridge University Press, England 1987
5. Osher, S., Sethian, J.A.: Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. J. Comput. Phys. 79: 12–49 (1988)
6. Peskin, C.S.: Numerical Analysis of Blood Flow in the Heart. J. Comput. Phys. 25: 220–252 (1977)
7. Rudman, M.: Volume-Tracking Methods for Interfacial Flow Calculations. Int. J. Numer. Methods Fluids 24: 671–691 (1997)
8. Strang, G., Fix, G.J.: An Analysis of the Finite Element Method. Englewood Cliffs: Prentice Hall 1973. now published by Wellesley-Cambridge Press, Wellesley, MA; Cambridge, MA
9. Sussman, M., Fatemi, M., Smereka, P., Osher, S.: An Improved Level Method for Incompressible Two-Phase Flows. Computers & Fluids 27: 663–680 (1998)
10. Sussman, M., Smereka, P., Osher, S.: A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. J. Comput. Phys. 114: 146–159 (1994)
11. Tornberg, A-K.: A Finite Element Based Level Set Method for Multiphase Flow Simulations. Licentiate's thesis, Royal Institute of Technology, Department of Numerical Analysis and Computer Science, September 1998. ISBN 91-7170-317-9, TRITA-NA-9817
12. Tornberg, A-K., Metcalfe, R.W., Scott, R., Bagheri, B.: A Fluid Particle Motion Simulation Method. In Computational Science for the 21st Century, pp. 312–321. New York: Wiley 1997
13. Unverdi, S.O., Tryggvason, G.: A Front-Tracking Method for Viscous, Incompressible, Multi-Fluid Flows. J. Comput. Phys. 100: 25–37 (1992)