INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:659–672
Published online 28 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1147

A new implicit surface tension implementation for interfacial flows

S. Hysing*,†

Institute of Applied Mathematics (LS III), University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany

SUMMARY

A new implementation of surface tension effects in interfacial flow codes is proposed which is both fully implicit in space, that is the interface never has to be reconstructed, and also semi-implicit in time, with semi-implicit referring to the time integration of the surface tension forces. The main idea is to combine two previously separate techniques to yield a new expression for the capillary forces. The first is the continuum surface force (CSF) method, which is used to regularize the discontinuous surface tension force term. The regularization can be elegantly implemented with the use of distance functions, which makes the level set method a suitable choice for the interface-tracking algorithm. The second is to use a finite element discretization together with the Laplace–Beltrami operator, which enables simple reformulation of the surface tension term into its semi-implicit equivalent. The performance of the new method is benchmarked against standard explicit methods, where it is shown that the new method is significantly more robust for the chosen test problems when the time steps exceed the numerical capillary time step restriction. Some improvements are also found in the average number of nonlinear iterations and linear multigrid steps taken while solving the momentum equations. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: multiphase flow; implicit surface tension; finite elements; level set method

1. INTRODUCTION

Various approaches exist for tracking interfaces while simulating interfacial flow phenomena. These range from explicit arbitrary Lagrangian–Eulerian (ALE) moving mesh algorithms to fully implicit Eulerian fixed grid methods, with front tracking approaches usually falling somewhere in between. The Lagrangian advantage is the sharpness of the interface representation, with the disadvantage of the added complexity of a moving grid/ALE code, which except for rare cases is unable to handle complex merging and breaking of interfaces. The Eulerian approach is the opposite side of the coin, that is ease of implementation but at the cost of

*Correspondence to: S. Hysing, Institute of Applied Mathematics (LS III), University of Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany.
†E-mail: shuren.hysing@math.uni-dortmund.de

Received 29 August 2005
Revised 26 October 2005
Accepted 26 October 2005

a sharp interface representation. Unfortunately most Eulerian approaches use some form of interface tracking too where the interface location has to be found explicitly. Some examples of this is calculating curvature in volume of fluid (VOF) simulations and redistancing in level set codes.

Surface tension which is a major component of multiphase flow codes is almost exclusively implemented with explicit time integration. This has the drawback that the time step size is limited by both the grid size and coefficient of surface tension. This can potentially lead to prohibitively costly simulations even for problems where capillary effects dominate and we do not expect much distortion of the interface. Implicit and semi-implicit implementations do not have this time step restriction allowing for more efficient simulations.

In this paper we introduce a new method for interfacial flow calculations which combines an Eulerian approach with semi-implicit time integration of the surface tension forces. Section 2 presents the governing equations that we address and derives the new method. Section 3 briefly describes the numerical algorithm in which the model is implemented and evaluates it on three examples. Finally, Section 4 summarizes the evaluation of the new model and also gives a brief outlook.

2. MODELLING OF SURFACE TENSION EFFECTS

When modelling and simulating fluid flow with immiscible fluids, surface tension effects generally play a great role in defining the physical behaviour. In the case of incompressible flow we have to solve the Navier–Stokes equations:

$$\rho(\mathbf{x})\left(\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)=-\nabla p+\nabla\cdot(\mu(\mathbf{x})(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}}))+\rho(\mathbf{x})\mathbf{g}$$

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

in $\Omega\subset\mathbb{R}^d$, $\mathbf{x}\in\Omega$ for varying density $\rho(\mathbf{x})$ and viscosity $\mu(\mathbf{x})$ fields. Here $\mathbf{g}$ represents external forces such as gravity.

At the $d-1$-dimensional interface $\Gamma\subset\Omega$, separating the different fluids, the following boundary conditions apply:

$$[\mathbf{u}]|_\Gamma=0,\quad -[-p\mathbf{I}+\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})]|_\Gamma\cdot\hat{\mathbf{n}}=\sigma\kappa\hat{\mathbf{n}}$$

where $\hat{\mathbf{n}}$ denotes the interface normal and $[\mathrm{A}]|_\Gamma$ is the jump of property A across the interface. These conditions imply continuity of the velocity across the interface and also a jump in the normal stress proportional to the coefficient of surface tension σ and the curvature of the interface κ. The interface conditions may be rewritten as volumetric forces and then take the form

$$\mathbf{f}_{\mathrm{st}}=\sigma\kappa\hat{\mathbf{n}}\delta(\Gamma,\mathbf{x}) \tag{1}$$

where $\delta(\Gamma,\mathbf{x})$ is a Dirac delta function localizing the surface tension force to the interface between the different fluids. This immersed interface (or boundary) approach has its roots in the early work on blood flow by Peskin [1], and later its extensions to VOF calculations with surface tension forces by Brackbill *et al.* [2] who dubbed it the continuum surface force (CSF) method.

2.1. Semi-implicit time integration

When working with the finite element method we can employ partial integration to transform the explicit calculation of the curvature to the test function space. This methodology was introduced by Dziuk [3] and later applied to flow calculations in the ALE framework by Bänsch and coworkers [4–6], Sashikumaar and Tobiska [7], and Matthies [8]. Some work has also been done in the context of Eulerian fixed grids; with level sets by Groß *et al.* [9] and with front tracking by Minev and coworkers [10].

In order to understand this method we first need to introduce some definitions from differential geometry.

Definition 1
The tangential gradient of a function f, which is differentiable in an open neighbourhood of Γ, is defined by

$$\underline{\nabla} f(x) = \nabla f(x) - (\hat{\mathbf{n}}(x) \cdot \nabla f(x))\hat{\mathbf{n}}(x), \quad x \in \Gamma$$

Here, ∇ denotes the usual gradient in $\mathbb{R}^d$.

Definition 2
If f is two times differentiable in a neighbourhood of Γ, then we define the Laplace–Beltrami operator of f as

$$\underline{\Delta} f(x) = \underline{\nabla} \cdot (\underline{\nabla} f(x)), \quad x \in \Gamma \tag{2}$$

Lemma 1
A theorem of differential geometry now states that

$$\underline{\Delta} \mathrm{id}_\Gamma = \kappa \hat{\mathbf{n}} \tag{3}$$

where κ is the mean curvature and id_Γ is the identity mapping on Γ. For the proof we refer to basic textbooks in differential geometry such as Reference [11].

To transform the surface tension term (1) to its variational equivalent, which is the basis of a finite element discretization, we start by multiplying it with a suitably chosen test function space $\mathbf{v}$ and integrate over Ω which yields

$$\mathbf{f}_{\mathrm{st}} = \int_\Omega \sigma \kappa \hat{\mathbf{n}} \cdot \mathbf{v} \delta(\Gamma, \mathbf{x}) \, \mathrm{d}\mathbf{x} = \int_\Gamma \sigma \kappa \hat{\mathbf{n}} \cdot \mathbf{v} \, \mathrm{d}\Gamma \tag{4}$$

Here we have also embedded the delta function in the integral to simplify the expressions. Using relations (2) and (3) with Equation (4) and partial integration now gives us the following:

$$\begin{aligned} \mathbf{f}_{\mathrm{st}} &= \int_\Gamma \sigma \kappa \hat{\mathbf{n}} \cdot \mathbf{v} \, \mathrm{d}\Gamma = \int_\Gamma \sigma (\underline{\Delta} \mathrm{id}_\Gamma) \cdot \mathbf{v} \, \mathrm{d}\Gamma \\ &= -\int_\Gamma \sigma \underline{\nabla} \mathrm{id}_\Gamma \cdot \underline{\nabla} \mathbf{v} \, \mathrm{d}\Gamma + \int_\gamma \sigma \partial_\gamma \mathrm{id}_\Gamma \cdot \mathbf{v} \, \mathrm{d}\gamma \end{aligned} \tag{5}$$

where the boundary term $\partial_\gamma \mathrm{id}_\Gamma = \hat{\mathbf{n}}_\gamma$ is acting on the tangent line γ, given by the intersection between the interface Γ and boundary of Ω, in the direction tangential to the interface. This term will only appear if our integration path does not posses a closed shape and $\mathbf{v}$ is non-vanishing on γ.

Time integration. The most common way of discretizing the surface tension forces in time is explicit integration. The surface tension forces are evaluated for the interface at the previous time step and added to the right-hand side as a source term, that is

$$\mathbf{f}_{\mathrm{st}} = \int_{\Gamma^n} \sigma \kappa^n \hat{\mathbf{n}}^n \cdot \mathbf{v} \, \mathrm{d}\Gamma \tag{6}$$

$$= -\int_{\Gamma^n} \sigma \underline{\nabla}(\mathrm{id}_\Gamma)^n \cdot \underline{\nabla}\mathbf{v} \, \mathrm{d}\Gamma \tag{7}$$

where a superscript n denotes the old time level (with boundary terms omitted). This will impose a limit on the time step by introducing a numerical capillary time scale

$$\Delta t_{\mathrm{num}}^{(\mathrm{ca})} < \sqrt{\frac{\langle \rho \rangle h^3}{2\pi\sigma}} \tag{8}$$

where h is the mesh size and $\langle \rho \rangle$ is the average fluid density at the interface [2]. As can be seen this constraint is rather restrictive for high mesh densities (scaling like $h^{3/2}$) and also large coefficient of surface tension (scaling like $\sigma^{-1/2}$).

Following the work of Bänsch [4, 5] we can lift this restriction by using a semi-implicit time discretization. This is accomplished by writing the new interface position as a function of the old position

$$(\mathrm{id}_\Gamma)^{n+1} = (\mathrm{id}_\Gamma)^n + \Delta t^{n+1} \mathbf{u}^{n+1} \tag{9}$$

where $\Delta t^{n+1} = t^{n+1} - t^n$ is the time step and $\mathbf{u}^{n+1}$ is the velocity field at the new time level. This results in the following representation of the surface tension effects:

$$\mathbf{f}_{\mathrm{st}} = -\int_{\Gamma^n} \sigma \underline{\nabla}(\mathrm{id}_\Gamma)^n \cdot \underline{\nabla}\mathbf{v} \, \mathrm{d}\Gamma - \Delta t^{n+1} \int_{\Gamma^n} \sigma \underline{\nabla}\mathbf{u}^{n+1} \cdot \underline{\nabla}\mathbf{v} \, \mathrm{d}\Gamma \tag{10}$$

The new term compared to Equation (7) is linear with respect to the velocity at time level $n+1$ and can thus simply be assembled as a positive definite contribution to the iteration matrix. This approach is also a little bit simpler compared to the earlier efforts by Hochstein and Williams [12], which essentially required the normals and curvature to be written implicitly, that is

$$\hat{\mathbf{n}}^{n+1} \approx \hat{\mathbf{n}}^n + \Delta t \left(\frac{\partial \hat{\mathbf{n}}}{\partial t} \right)^n; \quad \kappa^{n+1} \approx \kappa^n + \Delta t \left(\frac{\partial \kappa}{\partial t} \right) \approx \kappa^n - \Delta t \nabla \cdot \left(\frac{\partial \hat{\mathbf{n}}}{\partial t} \right)^n$$

The clear advantage the semi-implicit discretization (10) has over a pure explicit one is that the additional term represents a diffusion operator working in the tangential direction of Γ. This results in a more physical implementation of capillary effects since an increased coefficient of surface tension now generates more interface diffusion, that is a stiffer system, instead of a larger destabilizing source term.

2.2. Fully implicit surface tension force

Having arrived at the semi-implicit expression in time for the surface tension force we would now like to combine it with the CSF framework, which would enable us to work fully implicit in space while retaining the stabilizing effect. To do this we go back to Equations (4) and (5) but now keep the delta function in the expressions

$$\mathbf{f}_{\text{st}} = \int_\Omega \sigma\kappa\hat{\mathbf{n}} \cdot \mathbf{v}\delta(\Gamma,\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sigma(\underline{\Delta}\mathrm{id}_\Gamma)\cdot(\mathbf{v}\delta(\Gamma,\mathbf{x}))\,\mathrm{d}\mathbf{x}$$

$$= -\int_\Omega \sigma\underline{\nabla}\mathrm{id}_\Gamma \cdot \underline{\nabla}(\mathbf{v}\delta(\Gamma,\mathbf{x}))\,\mathrm{d}\mathbf{x} = -\int_\Omega \sigma\underline{\nabla}\mathrm{id}_\Gamma \cdot \underline{\nabla}\mathbf{v}\delta(\Gamma,\mathbf{x})\,\mathrm{d}\mathbf{x}$$

Applying the semi-implicit time integration (9) now gives

$$\mathbf{f}_{\text{st}} = -\int_\Omega \sigma\underline{\nabla}(\mathrm{id}_\Gamma)^n \cdot \underline{\nabla}\mathbf{v}\delta(\Gamma^n,\mathbf{x})\,\mathrm{d}\mathbf{x} - \Delta t^{n+1}\int_\Omega \sigma\underline{\nabla}\mathbf{u}^{n+1}\cdot\underline{\nabla}\mathbf{v}\delta(\Gamma^n,\mathbf{x})\,\mathrm{d}\mathbf{x}$$

The final step is to substitute the singular Dirac delta function δ with its regularized counterpart δ_ε, after which we arrive at the following expression for the surface tension force:

$$\mathbf{f}_{\text{st}} = \int_\Omega \sigma\delta_\varepsilon(\Gamma^n,\mathbf{x})\ \underline{\nabla}(\widetilde{\mathrm{id}}_\Gamma)^n\cdot\underline{\nabla}\mathbf{v}\,\mathrm{d}\mathbf{x} + \Delta t\int_\Omega \sigma\delta_\varepsilon(\Gamma^n,\mathbf{x})\underline{\nabla}\mathbf{u}\cdot\underline{\nabla}\mathbf{v}\,\mathrm{d}\mathbf{x} \tag{11}$$

where $\widetilde{\mathrm{id}}_\Gamma$ is the extension of the interface over the support of δ_ε with width 2ε.

Assuming that it is possible to find an implicit way to construct the regularized delta function in Equation (11), we have then achieved a fully implicit formulation for the surface tension force, which should not be bounded by the capillary time step restriction (8).

2.3. Regularization of the Dirac delta function

The surface tension force integral in Equation (4) does not need to be evaluated explicitly over the interface itself. Of course this approach will be the most accurate one, however a major drawback is that the interface has to be explicitly found, which can be quite a difficult and imprecise task for higher-order schemes, and especially in three dimensions. By regularization of the singular Dirac delta function we have the possibility of evaluating the singular terms implicitly.

The Dirac delta function is defined as

$$\int_{\mathbb{R}^d}\delta(\Gamma,\mathbf{x})f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Gamma f(\mathbf{X}(S))\,\mathrm{d}S$$

where $\mathbf{X}(S)\in\Gamma$. To regularize this function in higher dimensions we can follow two approaches, the first is to use the method of Peskin [13] and construct δ_ε by multiplication of d one-dimensional regularized delta functions

$$\delta_\varepsilon(\Gamma,\mathbf{x}) = \int_\Gamma \prod_{k=1}^{d}\delta_{\varepsilon_k}(x^{(k)} - X^{(k)}(S))\,\mathrm{d}S$$

This approach however does not let us escape from the explicit localization of the interface since $\mathbf{X}(S)=(X^{(1)}(S),\ldots,X^{(d)}(S))$ are points on the interface Γ. Fortunately there is another path open to us, and that is to construct $\delta_{\boldsymbol{\varepsilon}}$ from a distance function

$$\delta_{\boldsymbol{\varepsilon}}(\Gamma,\mathbf{x})=\delta_{\varepsilon}(\mathrm{dist}(\Gamma,\mathbf{x}))$$

where $\mathrm{dist}(\Gamma,\mathbf{x})$ gives the minimum distance from $\mathbf{x}$ to Γ. The regularized continuous delta function δ_{ε} is defined as

$$\delta_{\varepsilon}(x)=\begin{cases}\dfrac{1}{\varepsilon}\varphi_m(x/\varepsilon) & |x|\leqslant\varepsilon=mh\\ 0 & |x|>\varepsilon=mh\end{cases}$$

where h is the mesh spacing which together with the constant m defines the support ε of the regularized delta function. Some of the most common choices for the kernel function φ are the linear hat function

$$\varphi(\xi)=1-|\xi|$$

the commonly used cosine function introduced by Peskin

$$\varphi(\xi)=\tfrac{1}{2}(1+\cos(\pi\xi))$$

and polynomials of various degrees of which one example is

$$\varphi(\xi)=\tfrac{35}{32}(1-3\xi^2+3\xi^4-\xi^6) \tag{12}$$

The task is now to construct an appropriate distance function, which can be done explicitly. A more attractive alternative however is to use the level set method to track the interface.

2.4. Level set method

The main idea is to assign $\Gamma(t)$ as the zero level set embedded in a higher-dimensional function ϕ, that is,

$$\Gamma(t)=\{\mathbf{x}\in\mathbb{R}^d\mid\phi(\mathbf{x},t)=0\}$$

The level set function ϕ is generally described (or at least initialized) as a signed distance function d from the interface

$$\phi=d(\Gamma,\mathbf{x})=\begin{cases}\mathrm{dist}(\Gamma,\mathbf{x}), & \mathbf{x}\in\Omega_1\\ 0, & \mathbf{x}\in\Gamma\\ -\mathrm{dist}(\Gamma,\mathbf{x}), & \mathbf{x}\in\Omega_2\end{cases}$$

The evolution of ϕ, and thus also implicitly the interface, can be posed as a general transport problem

$$\frac{\partial\phi}{\partial t}+F|\nabla\phi|=0$$

where F is a speed function dictating the rate of change of Γ in the normal direction. F will only depend on the velocity field $\mathbf{u}$ for the considered multiphase flow applications.

3. NUMERICAL EXAMPLES

In this section we present the results from computational simulations highlighting the performance of the new surface tension implementation variant (11), which we label CSF-LBI. The new method is compared to the standard explicit method labelled CSF (6).

The tests were performed with the finite element flow solver package FEATFLOW [14] extended to incorporate interfacial flow with the level set method. The flow variables were discretized with non-conforming $\tilde{\mathbb{Q}}_1\mathbb{Q}_0$ basis functions and solved with the discrete projection method. This decouples the velocity from the pressure and allows for efficient solutions via multigrid. The momentum equations were also stabilized with streamline diffusion (SUPG) to handle convection dominated situations.

The level set solver employed a continuous $\mathbb{Q}_1$ approximation with FEM-TVD [15] to stabilize the pure transport equation. Normals and curvature were reconstructed from the level set function with gradient recovery (ZZ-technique [16]), which leads to better approximation of geometrical quantities. After each time step the level set field was reinitialized with the fast marching method [17]. Height correction (as described in Reference [18]) was also applied to ensure mass conservation. To regularize the delta function the polynomial (12) was used with $m = 1.5$.

For the time integration a second-order Crank–Nicolson scheme was used. The dependent variables were solved sequentially after each other in each time step according to the following scheme.

1. Solve the momentum equations with surface tension.
2. Solve the pressure Poisson equation.
3. Update the pressure and velocity.
4. Solve the level set equation.
5. Reinitialize the level set field.
6. Apply mass correction to the level set field.

3.1. Static bubble

This test case models a perfectly stationary circular bubble at equilibrium. According to the Laplace–Young law the pressure inside the bubble is equal to $p_{\text{in}} = p_{\text{out}} + \sigma/r$, where r is the radius of the bubble. Since everything is stationary we should ideally have a zero velocity field but due to certain imbalances in our numerical method spurious velocity currents will be generated.

To have a comparison we follow the configuration given in Reference [18], which is a bubble with radius $r = 0.25$ positioned in the centre of a unit square. The coefficient of surface tension and all viscosities were set to unity while densities were given a magnitude of 10^4. This corresponds to a Laplace number of $La = (2r)\sigma\rho\mu^{-2} = 5 \times 10^3$. A fixed time step of $\Delta t = 0.01$ was used, and computations were performed until $t = 125$.

Tables I and II show the error of the dimensionless velocity for the different methods in l^∞ and l^1 norms defined as $\max_i |\mathbf{u}_i\mu/\sigma|$ and $\frac{1}{N}\sum_{i=1}^{N} |\mathbf{u}_i\mu/\sigma|$, respectively, where N is the number of nodes. The results with $\mathbb{P}_k\mathbb{P}_1$ discretization are reprinted from Reference [18]. From the comparisons we can conclude that the errors are of the same order or even a little bit better than those given by Smolianski. Also the new CSF-LBI method proved to be marginally more accurate than the standard CSF method.

Table I. The error and convergence rate (EOC) in the discrete l^∞ norm for the non-dimensional velocity $\mathbf{u}\mu/\sigma$.

1/h	$\mathbb{P}_1\mathbb{P}_1$	$\mathbb{P}_1^b\mathbb{P}_1$	$\mathbb{P}_2\mathbb{P}_1$	$\tilde{\mathbb{Q}}_1\mathbb{Q}_0^{\text{CSF}}$	$\tilde{\mathbb{Q}}_1\mathbb{Q}_0^{\text{CSF-LBI}}$
20	1.4×10^{-2}	1.5×10^{-2}	2.2×10^{-2}	7.4×10^{-3}	6.9×10^{-3}
40	9.1×10^{-3}	9.2×10^{-3}	1.3×10^{-2}	3.9×10^{-3}	3.7×10^{-3}
80	5.0×10^{-3}	4.2×10^{-3}	9.7×10^{-3}	2.0×10^{-3}	1.8×10^{-3}
160				9.8×10^{-4}	8.1×10^{-4}
EOC $\approx$	0.8	0.9	0.6	1.0	1.0

Table II. The error and convergence rate (EOC) in the discrete l^1 norm for the non-dimensional velocity $\mathbf{u}\mu/\sigma$.

1/h	$\mathbb{P}_1\mathbb{P}_1$	$\mathbb{P}_1^b\mathbb{P}_1$	$\mathbb{P}_2\mathbb{P}_1$	$\tilde{\mathbb{Q}}_1\mathbb{Q}_0^{\text{CSF}}$	$\tilde{\mathbb{Q}}_1\mathbb{Q}_0^{\text{CSF-LBI}}$
20	6.9×10^{-4}	8.6×10^{-4}	1.1×10^{-3}	6.7×10^{-4}	5.8×10^{-4}
40	1.8×10^{-4}	2.3×10^{-4}	3.5×10^{-4}	1.9×10^{-4}	1.6×10^{-4}
80	4.7×10^{-5}	5.9×10^{-5}	1.0×10^{-4}	5.2×10^{-5}	4.1×10^{-5}
160				1.4×10^{-5}	1.0×10^{-5}
EOC $\approx$	1.9	1.9	1.7	1.9	2.0

Table III. Absolute (top) and relative (bottom) errors in the Laplace–Young law.

1/h	$\mathbb{P}_k\mathbb{P}_1$	$\tilde{\mathbb{Q}}_1\mathbb{Q}_0^{\text{CSF}}$	$\tilde{\mathbb{Q}}_1\mathbb{Q}_0^{\text{CSF-LBI}}$
$\lvert p_{\text{in}} - p_{\text{out}} - \sigma/r\rvert$			
20	2.0×10^{-2}	4.8×10^{-2}	6.1×10^{-3}
40	2.0×10^{-3}	1.2×10^{-2}	3.5×10^{-3}
80	6.0×10^{-4}	3.1×10^{-3}	1.0×10^{-3}
160		7.8×10^{-4}	2.7×10^{-4}
$\dfrac{\lvert p_{\text{in}} - p_{\text{out}} - \sigma/r\rvert}{(\sigma/r)}100\%$			
20	0.5%	1.2%	0.15%
40	0.05%	0.31%	0.088%
80	0.015%	0.078%	0.026%
160		0.020%	0.0066%

Table III shows how well the pressure field fulfilled the Laplace–Young law in both absolute and relative error norms. Overall we can see that the $\tilde{\mathbb{Q}}_1\mathbb{Q}_0$ approach with the standard CSF method does not perform equally well to the $\mathbb{P}_k\mathbb{P}_1$ approach, while the new CSF-LBI method however gave results on a comparable level. The overall better pressure approximation of the $\mathbb{P}_k\mathbb{P}_1$ approximation could possibly be attributed to the higher order of the pressure space. Figure 1 shows pressure cut-lines at $y = 0.5$ for various levels of grid refinement and as can be seen the pressure approximation is quite sharp for both CSF methods even on coarse grids.

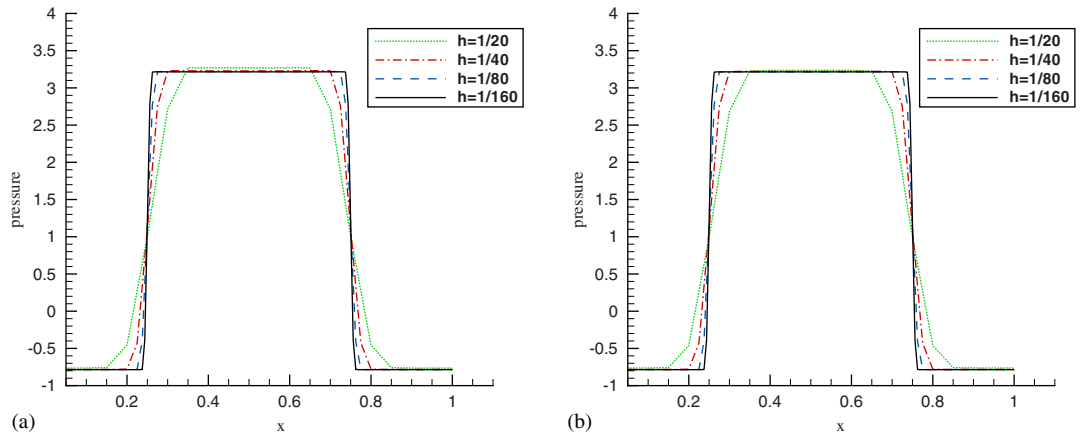

Figure 1. Pressure cut-line ($y = 0.5$) for four different mesh sizes: (a) CSF; and (b) CSF-LBI.

Table IV. Capillary time step restriction for the standard CSF method applied to the oscillating bubble example.

Mesh level	L1	L2	L3	L4	L5	L6	L7
$\Delta t_{\text{num}}^{(\text{ca})}$	11.3	4.0	1.4	0.5	0.17	0.06	0.02

3.2. *Oscillating bubble*

In this second more dynamic example we continue with the configuration used in the previous example, but now initially perturb the circle to an elliptical shape by scaling the semi-axes a factor 1.25 in the x-direction and 0.8 in the y-direction. The ellipse or bubble consists of the same fluid as in the surrounding unit square cavity. The fluid has a density of 10^4, viscosity 1, and coefficient of surface tension equal to 0.1. Simulations for various levels of refinement of the initial 5×5 mesh were performed with a fixed time step $\Delta t = 5$ until $t = 1000$. At the final time the bubble is expected to have reached an equilibrium state that is a stable circular shape. Theoretically, the capillary time step restriction (8) is already exceeded on the second mesh refinement (level 3) as can be seen in Table IV, and instabilities can thus be expected to appear on this and finer meshes.

Figure 2 shows the results for the standard CSF method at six representative times for levels 3–6. As is apparent numerical oscillations start to appear and pollute the solution as the mesh is refined. The new CSF-LBI method however proved to be very stable, and as can be seen from Figure 3 only the very finest levels (at 80–250 times $\Delta t_{\text{num}}^{(\text{ca})}$) show the onset of oscillations.

3.3. *Rising bubble*

The final test example concerns a bubble rising in a heavier fluid. The bubble is given an initial radius of $r_0 = 0.2$ and is placed at $[0.5; 0.5]$ in a rectangular domain with dimensions

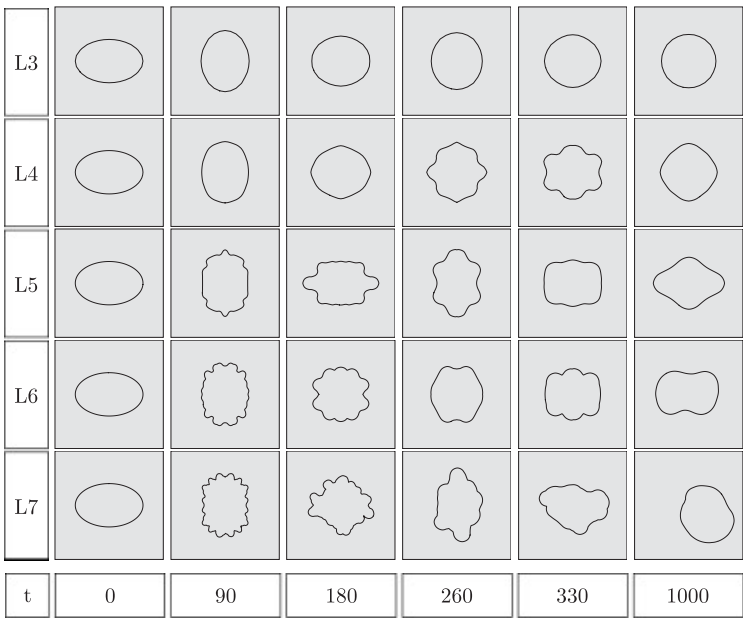

Figure 2. Evolution of an oscillating bubble with the standard CSF method. Refinement levels 3–7 with $\Delta t = 5$.

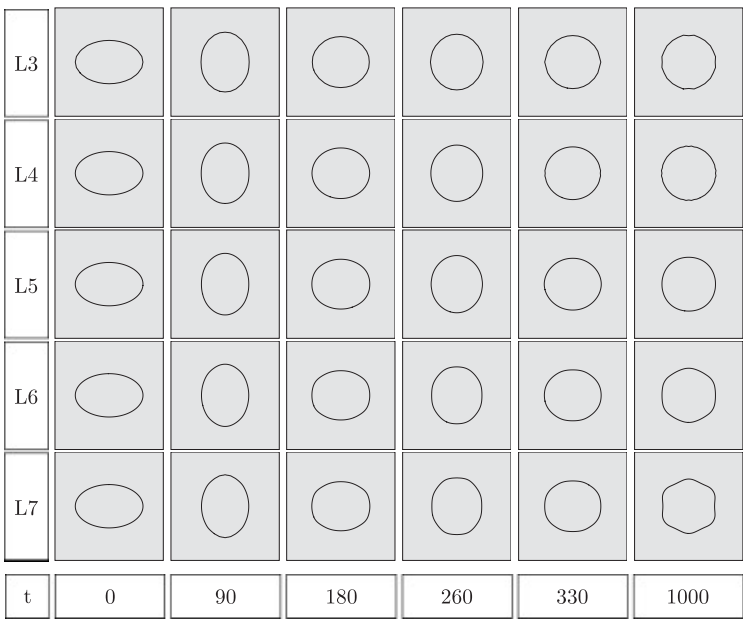

Figure 3. Evolution of an oscillating bubble with the new CSF-LBI method. Refinement levels 3–7 with $\Delta t = 5$.

Table V. Physical parameters used in the rising bubble example.

ρ_1 (liquid)	10^4
ρ_2 (gas)	10^3
μ_1 (liquid)	1
μ_2 (gas)	1
g_y	-8×10^{-4}
σ	0.5

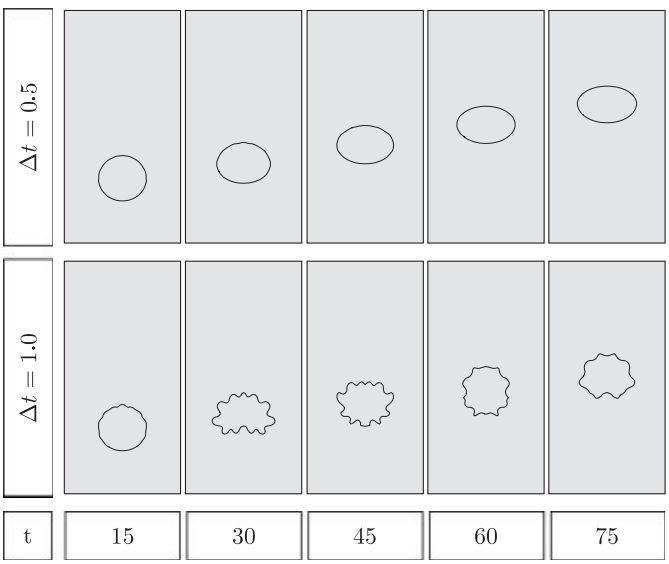

Figure 4. Evolution of a rising bubble with the standard CSF method with time steps $\Delta t = 0.5$ and $\Delta t = 1.0$.

$[0;1] \times [0;2]$. No-slip conditions are applied on the horizontal walls and slip conditions on the vertical ones. Initially, both fluids are at rest having a velocity of zero everywhere.

The physical parameters for the simulation are listed in Table V and correspond to a Reynolds number $Re = (2r_0)^{3/2} g_y^{1/2} \rho_1 \mu_1^{-1} = 71.6$ and Eötvös number $Eo = 4\rho_1 g_y r_0^2 \sigma^{-1} = 2.56$. According to Clift *et al.* [19] such a bubble is expected to assume an ellipsoidal shape. This is valid for fully three-dimensional bubbles but not necessarily for the considered two-dimensional one, it should however give an indication of the final shape that might be found. The computations were performed on a single 80×160 mesh where the time step Δt was varied between 0.0125 and 4 to test the stabilizing capabilities of the new surface tension implementation method. From (8) the estimate of the capillary time step restriction is $\Delta t_{\text{num}}^{(\text{ca})} = 0.056$ indicating possible problems for the larger time step sizes.

Figure 4 shows the results for the standard CSF method at time steps $\Delta t = 0.5$ and $\Delta t = 1.0$, where it can be seen that severe oscillations pollute the solution at the larger time step. Further increases in time step size resulted in a complete breakdown of the algorithm. It is interesting to note that it was still possible to perform the simulation although the capillary time step

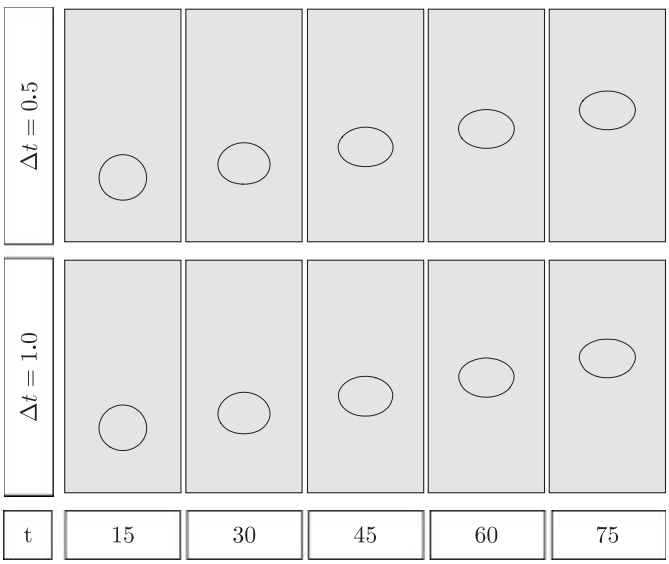

Figure 5. Evolution of a rising bubble with the new CSF-LBI method with time steps $\Delta t = 0.5$ and $\Delta t = 1.0$.

Table VI. Averaged number of nonlinear iterations (ANNL), linear multigrid steps for velocity U (AMGU) and pressure P (AMGP), respectively, for the rising bubble test case with time steps $\Delta t = 0.5$ and $\Delta t = 0.25$.

	Δt=0.5		Δt=0.25	
	CSF	CSF-LBI	CSF	CSF-LBI
ANNL	4.1	3.6	3	2.6
AMGU	5.2	4.8	3	2.7
AMGP	5.3	5.2	4.6	4.6

restriction was exceeded by almost a factor of 10. Figure 5 in contrast shows the results for the new CSF-LBI method where both time steps yielded the same qualitative solution. The method did in fact even work quite well up to $\Delta t = 2.0$, after which there also appeared too much distortion in the interface contour. Table VI lists the averaged number of nonlinear iterations and linear multigrid steps for velocity and pressure at time steps $\Delta t = 0.5$ and $\Delta t = 0.25$. The new CSF-LBI method resulted in a slight decrease in the number of nonlinear iterations and multigrid steps while solving the momentum equations, the multigrid steps for solving the Pressure Poisson equation were however unaffected.

4. CONCLUSIONS AND OUTLOOK

A fully implicit, with respect to space, Eulerian method to implement surface tension effects has been presented which also is semi-implicit in time. The main idea of the new imple-

mentation is to combine the continuum surface force (CSF) method with a Laplace–Beltrami operator working on the variational form of the capillary force. The CSF method ensures implicitness in space with respect to the interface location and the Laplace–Beltrami operator enables the surface tension terms to be formulated semi-implicitly in time. The advantage of using semi-implicit instead of explicit time integration of the surface tension forces is that it potentially allows for larger time steps with respect to the numerical capillary time step restriction imposed otherwise.

Efficient implementation of the new method is made possible by using the implicitness inherent in the level set methodology together with a finite element discretization in space. Since existence of a distance function is a natural result of working with the level set method it is particularly suitable for use in CSF formulations, that is in the construction of approximations to Heaviside and Dirac delta functions. Another advantage of working with the level set method is that geometrical quantities are globally available and can easily be recovered implicitly. The finite element method, on the other hand, naturally gives us access to the variational formulation of the equations enabling the subsequent application of the Laplace–Beltrami operator on the surface tension term.

Three numerical examples were presented to validate the new CSF-LBI method; a static bubble following the Laplace–Young law, an oscillating bubble, and finally a rising bubble. The static bubble test case showed that the new CSF-LBI method gives comparable or even marginally better accuracy than the standard CSF method. The following two cases, which tested different time step sizes with respect to mesh refinement, showed that the new method clearly has a significant stabilizing effect on lessening the capillary time step restriction. This effect was most dramatic for the second purely capillary driven case where the maximum possible time step was roughly two magnitudes larger than allowed for a purely explicit implementation. In the case of the rising bubble the standard explicit method worked remarkably well allowing for 10 times larger time step compared to the capillary time step restriction. The new CSF-LBI method however proved a little bit better being able to use a 20 times larger time step. For this case it was also shown that the average number of nonlinear iterations and multigrid steps needed in the solution of the momentum equations were somewhat decreased with the new implementation method.

The future application of the presented CSF-LBI method to interfacial flows with significant topology change (breakup and coalescence) and differing flow conditions will be investigated rigorously in forthcoming publications. Only then will it be possible to fully evaluate the performance over a wide range of flow regimes, and to find eventual drawbacks of the new method. Other interesting aspects to examine is the efficiency when coupled with mesh deformation/adaption algorithms and also higher-order finite element discretizations. The final step would be to move to three-dimensional space which does not pose any additional conceptual complexities or problems due to the fully Eulerian approach.

REFERENCES

1. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**: 220–252.
2. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1997; **100**:335–354.
3. Dziuk G. An algorithm for evolutionary surfaces. *Numerische Mathematik* 1991; **58**(6):603–611.
4. Bänsch E. Finite element discretization of the Navier–Stokes equations with a free capillary surface. *Numerische Mathematik* 2001; **88**(2):203–235.

5. Bänsch E. Numerical methods for the instationary Navier–Stokes equations with a free capillary surface. *Habilitation Thesis*, Universität Freiburg, 1998.
6. Bänsch E, Berg CP, Ohlhoff A. Uniaxial extensional flows in liquid bridges. *Journal of Fluid Mechanics* 2004; **521**:353–379.
7. Sashikumaar G, Tobiska L. Finite element simulation of a droplet impinging a horizontal surface. *Proceedings of Algoritmy* 2005; 1–11.
8. Matthies G. Finite element methods for free boundary value problems with capillary surfaces. *Ph.D. Thesis*, Shaker Verlag, 2002.
9. Groß S, Reichelt V, Reusken A. A finite element based level set method for two-phase incompressible flows. *IGPM-Report RWTH Aachen* 2004; **243**.
10. Minev PD, Chen T, Nandakumar K. A finite element technique for multifluid incompressible flow using Eulerian grids. *Journal of Computational Physics* 2003; **187**:255–273.
11. Gallot S, Hulin D, Lafontaine J. *Riemannian Geometry*. Springer: Berlin, 2004.
12. Hochstein JI, Williams TL. An implicit surface tension model. *AIAA Meeting Papers* 1996; **599**.
13. Peskin CS. The immersed interface method. *Acta Numerica* 2002; **11**:479–517.
14. Turek S, Becker C. FEATFLOW finite element software for the incompressible Navier–Stokes equations. *User Manual Release 1.2*, Dortmund, 1999.
15. Kuzmin D, Turek S. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. *Journal of Computational Physics* 2004; **198**:131–158.
16. Zienkiewicz OC, Zhu JZ. The superconvergence patch recovery and a posteriori error estimates. Part 1: the recovery technique. *International Journal for Numerical Methods in Engineering* 1992; **33**:1331–1364.
17. Hysing S, Turek S. The Eikonal equation: numerical efficiency vs. algorithmic complexity on quadrilateral grids. *Proceedings of Algoritmy* 2005; 22–31.
18. Smolianski A. Numerical modeling of two-fluid interfacial flows. *Ph.D. Thesis*, University of Jyväskylä, 2001.
19. Clift R, Grace JR, Weber ME. *Bubbles, Drops and Particles*. Academic Press: New York, 1978.

