Level set calculations for incompressible two-phase flows on a dynamically adaptive grid

Yana Di [*] Ruo Li [†] Tao Tang [‡] Pingwen Zhang[§]

April 24, 2006

Abstract

We present a coupled moving mesh and level set method for computing incompressible two-phase flow with surface tension. This work extends a recent work of Di et al. [SIAM J. Sci. Comput., 26 (2005), pp. 1036-1056] where a moving mesh strategy was proposed to solve the incompressible Navier-Stokes equations. With the involvement of the level set function and the curvature of the interface, some subtle issues in the moving mesh scheme, in particular the solution interpolation from the old mesh to the new mesh and the choice of monitor functions, require careful considerations. In this work, a simple monitor function is proposed that involves both the level set function and its curvature. The purpose for designing the coupled moving mesh and level set method is to achieve higher resolution for the free surface by using a minimum amount of additional expense. Numerical experiments for air bubbles and water drops are presented to demonstrate the effectiveness of the proposed scheme.

KEYWORDS: Level set method, moving mesh methods, two-phase flows.

Dedicated to Dr. Xudong Liu for friendship.

1 Introduction

In this paper, we will describe a moving mesh level set approach for computing incompressible two-phase flow with surface tension. The flow we consider has discontinuous density

[*] School of Mathematical Sciences, Peking University, 100871, Beijing, People's Republic of China. Email: yndi@math.pku.edu.cn

[†] School of Mathematical Sciences, Peking University, 100871, Beijing, People's Republic of China. Email: rli@math.pku.edu.cn

[‡] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong. Email: ttang@math.hkbu.edu.hk.

[§] School of Mathematical Sciences, Peking University, 100871, Beijing, People's Republic of China. Email: pzhang@math.pku.edu.cn.

and viscosity, and is characterized by large density and viscosity ratios at the free surface, e.g. air and water. The main existing computational methods used to solve incompressible two-phase flow problems include front-tracking methods, boundary integral methods, volume-of-fluid methods, and level set methods; see, e.g., [4, 23, 24, 28]. Our goal is to use the ideas in the level set approach [25] and in the moving mesh method for incompressible flow simulations of [11] to compute incompressible two-phase flow with surface tension.

The level set method was devised by Osher and Sethian [25] as a simple and versatile method for computing and analyzing the motion of an interface in two or three space dimensions. Assume that Γ bounds a (possibly multiply connected) region Ω. The goal is to compute the subsequent motion of Γ under a velocity field $\mathbf{u}$. This velocity can depend on position, time, the geometry of the interface, and the external physics. The interface is captured for later time as the zero level set of a smooth (at least Lipschitz continuous) function $\phi(\mathbf{x}, t)$; i.e., $\Gamma(t) = \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\}$. This deceptively trivial statement is of great significance for numerical computation, primarily because it can naturally capture the change of topology in the fluid interface, such as merging and breaking. Furthermore, the level set method can be generalized easily to three dimensions.

One of the purposes of this work is to apply the level set method on a dynamically moving grid. The basic idea of moving mesh method is to construct a transformation from a logical domain (or called computational domain) to the physical domain. A fixed mesh is given on the logical domain, and the transformation is realized by solving moving mesh PDEs or minimization problems for a mesh functional. By connecting points in the physical space corresponding to discrete points in the parameter space, the physical domain can be covered with a computational mesh suitable for the solution of finite difference/element equations; see, e.g., [1, 2, 7, 18, 35]. The key ingredients of the moving mesh methods include:

- **Mesh equations.** The mesh equations determine a one-to-one mapping from a regular domain in a parameter space to an irregularly shaped domain in physical space. By connecting points in the physical space corresponding to discrete points in the parameter space, the physical domain can be covered with a computational mesh suitable for the solution of finite difference/element equations. Choosing suitable mesh equations and solving them efficiently are very crucial for an effective moving mesh method;
- **Monitor function.** A monitor function is used to guide the mesh redistribution. It may depend on the solution arclength (in 1D), curvature, and a posteriori errors. In practice, local (spatial) smoothing of the monitor function is necessary, see, e.g., [7, 8];
- **Interpolations.** If the mesh equations are time-dependent and are solved simultaneously with the given differential equations, then interpolation of dependent variables from the old mesh to the new mesh is unnecessary. Otherwise, some kind of interpolation is required to pass the solution information on the old mesh to the newly

generated mesh.

In [13], Dvinsky suggests that harmonic function theory may provide a general framework for developing useful mesh generators. Unlike most other generalizations which add terms or functionals to the basic Winslow grid generator [34], Dvinsky's approach uses a single functional to accomplish the adaptive mapping. The critical points of this functional are *harmonic maps.* Motivated by the work of Dvinsky, a moving mesh finite element strategy based on harmonic mapping was proposed and studied by the authors in [16, 17]. The key idea of this strategy is to construct the harmonic map between the physical space and a parameter space by an iteration procedure. The idea of iterative mesh generations was used before, e.g., in spectral methods with co-ordinate transformations [33, 19]. The overall scheme proposed in [16] contains two parts: a solution algorithm and a mesh generation algorithm. These two parts are independent in the sense that the change of the physical PDEs will affect the first part only. In [11], a moving mesh scheme for solving the incompressible Navier-Stokes equations in the primitive variables formulation is developed using the moving mesh finite element strategy of [16]. The main effort in [11] is to design a divergence-free interpolation which is very essential for the incompressible problems. By some careful analysis, it is concluded that the interpolation can be implemented by solving a linearized inviscid Navier-Stokes-type equations.

The paper is organized as follows. In Section 2, the governing equations for the level set approach and the incompressible Navier-Stokes equations involving free surfaces will be described. The level set evolution scheme on a fixed grid will be described in Section 3. A complete moving mesh scheme designed for two-phase problems will be introduced in Section 4. A number of illustrative examples will be considered in Section 5. Some concluding remarks will be made in the final section.

2 Governing equations

The level set approach [25] has been used for computing moving boundaries which are singular or extremely thin with sharp gradients. A level set function $\phi(\mathbf{x},t)$ is defined to be a smooth function which is positive in the liquid and negative in the gas, with the zero level set of $\phi(\mathbf{x},t)$ representing the moving interface at time t.

When the underlying velocity field $\mathbf{u}$ is specified, the advection step involves the solution of the scalar transport equation for ϕ,

$$\phi_t + \mathbf{u}\cdot\nabla\phi = 0. \tag{2.1}$$

The unit normal on the interface, pointing from the gas into the liquid, and the curvature of the interface can be expressed in terms of $\phi(\mathbf{x},t)$:

$$\mathbf{n} = \left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0}, \quad \kappa = \nabla\cdot\left.\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right|_{\phi=0}. \tag{2.2}$$

Suppose that μ_g and μ_l are the viscosity for gas and liquid respectively, ρ_g and ρ_l are the density for gas and liquid respectively. Then continuous nondimensionalized viscosity and density can be defined as

$$\mu(\phi) = \mu_g/\mu_l + H(\phi)(1 - \mu_g/\mu_l), \quad \rho(\phi) = \rho_g/\rho_l + H(\phi)(1 - \rho_g/\rho_l),$$

where $H(\phi)$ is the Heaviside function defined by

$$H(\phi) = \begin{cases} 0, & \text{if} \quad \phi < 0, \\ \frac{1}{2}, & \text{if} \quad \phi = 0, \\ 1, & \text{if} \quad \phi > 0. \end{cases}$$

The governing equations for the two-dimensional incompressible Navier-Stokes equations separated by a free surface was derived in [10]:

$$\begin{aligned} &\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = \nabla \cdot (2\mu \mathbf{D}) - \nabla p + \sigma\kappa\delta(\phi)\nabla\phi + \rho\mathbf{g}, \\ &\nabla \cdot \mathbf{u} = 0, \end{aligned} \tag{2.3}$$

where $\mathbf{u} = (u, v)$ is the fluid velocity, $\rho = \rho(\mathbf{x}, t)$ is the fluid density, $\mu = \mu(\mathbf{x}, t)$ is the fluid viscosity, $\mathbf{D} = \frac{1}{2}[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^T]$ is the viscous stress tensor. The only body force being considered is the gravity denoted as $\mathbf{g}$. The surface tension term is considered to be a force concentrated on the interface. We denote σ as the surface tension coefficient and δ as the Dirac delta function. After a standard non-dimensionalization procedure, we have

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = \frac{1}{\mathsf{Re}}\nabla \cdot (2\mu\mathbf{D}) - \nabla p + \frac{1}{\mathsf{Fr}}\rho\mathbf{g} + \frac{1}{\mathsf{We}}\kappa\delta(\phi)\nabla\phi. \tag{2.4}$$

The dimensionless parameters used are Reynolds number, $\mathsf{Re} = \rho_l LU/\mu_l$, Froude number, $\mathsf{Fr} = U^2/gL$ and Weber number, $\mathsf{We} = \rho_l LU^2/\sigma$.

3 Level set evolution

The level set evolution equation (2.1) can be reformulated as

$$\begin{aligned} \phi_t + F(\nabla\phi, \mathbf{x})|\nabla\phi| &= f(\mathbf{x}), \\ F(\nabla\phi, \mathbf{x}) = \frac{\mathbf{u} \cdot \nabla\phi}{|\nabla\phi|}, &\quad f(\mathbf{x}) = 0. \end{aligned} \tag{3.1}$$

The above problem is solved by using the explicit positive coefficient scheme with triangular elements proposed by Barth and Sethian [3]. The algorithm can be summarized in Table 1.

We will give the interface a thickness as was done in the work of [28]. Numerically, we substitute the smoothed Heaviside function $H_\epsilon(\phi)$ for the sharp Heaviside function $H(\phi)$.

Table 1: The algorithm of the explicit positive coefficient scheme for (3.1).

```
1. Initialize $\phi_i^* = \omega_i = 0, i = 1, \cdots, |V|$.
2. For each triangle $T$, $i = 1, 2, 3$:
    $N_i(x) \in P_1,\ N_i(x_j) = \delta_{ij},\ j = 1, 2, 3,\ x \in T$,
    $F = \frac{1}{\mathrm{meas}(T)} \int_T \frac{\mathbf{u} \cdot \nabla\phi}{|\nabla\phi|} dx, \quad n_i = 2\mathrm{meas}(T)\nabla N_i, \quad \nabla\phi = \sum_{j=1}^{3} \nabla N_j \phi_j,$
    $K_i = \frac{F \nabla\phi \cdot n_i}{2|\nabla\phi|}, \quad \delta\phi = \sum_{l=1}^{3} K_l \phi_l, \quad \delta\phi_i = K_i^+ \sum_{l=1}^{3} K_l^- (\phi_i - \phi_l) / \sum_{l=1}^{3} K_l,$
    $\alpha_i = \max(0,\ \delta\phi_i/\delta\phi) / \sum_{l=1}^{3} \max(0,\ \delta\phi_l/\delta\phi), \quad \phi_i^* = \phi_i^* + \alpha\delta\phi, \quad \omega_i = \omega_i + \alpha_i \mathrm{meas}(T).$
3. For each $v_i \in V$,
    $\phi_i^{n+1} = \phi_i^n - \Delta t (\phi_i^*)^n / \omega_i^n$.
```

By giving the interface a thickness of 2ϵ, the smoothed Heaviside function is defined as

$$H_\epsilon(\phi) = \begin{cases} 0, & \phi < -\epsilon, \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} \sin\left(\pi\phi/\epsilon\right), & -\epsilon \le \phi \le \epsilon, \\ 1, & \phi > \epsilon. \end{cases} \tag{3.2}$$

In order to maintain $\phi(\mathbf{x}, t)$ as a distance function, we re-initialize a given level set function $\phi(\mathbf{x}, t)$ so that it is a distance function for $|\phi| < \epsilon$ without changing its zero level set. This is achieved (see [28]) by solving the following problem to steady state

$$\begin{aligned} d_\tau &= S(\phi)(1 - |\nabla d|), \quad \tau > 0, \\ d(\mathbf{x}, 0) &= \phi(\mathbf{x}, t), \end{aligned} \tag{3.3}$$

where S is the sign function and τ is an artificial time. It can be shown that the steady state solution of (3.3) is a distance function. Furthermore, $S(0) = 0$ implies that $d(\mathbf{x}, \tau)$ has the same zero level set as $\phi(\mathbf{x}, t)$. For numerical purposes it is useful to smooth the sign function using

$$S_\epsilon(\phi) = \frac{\phi}{\sqrt{\phi^2 + \epsilon^2}}.$$

Our discretization for approximating (3.3) is also based on the explicit positive coefficient scheme [3], which is outlined in Table 2.

We use an improvement to the re-distance step as described in [29]. The idea is based on the fact that the volume filled by each fluid must stay constant when the re-distance step is applied. In order to minimize volume variation, we project the current values of the level set function, denoted as $\tilde{d}_i^k$, onto new values, denoted as d_i^k, using

$$d_i^k = \tilde{d}_i^k + \lambda_i H'_\epsilon(d^0), \tag{3.4}$$

Table 2: The algorithm of the explicit positive coefficient scheme for (3.3).

```
1. Initialize $\phi_i^* = \omega_i = 0, i = 1, \cdots, |V|$.
2. For each triangle T, $i = 1, 2, 3$,
     $N_i(x) \in P_1,\ N_i(x_j) = \delta_{ij},\ j = 1,2,3,\ x \in T$,
     $F = \frac{1}{\mathrm{meas}(T)} \int_T S_\epsilon(\phi)dx, \quad n_i = 2\mathrm{meas}(T)\nabla N_i, \quad \nabla\phi = \sum_{j=1}^{3} \nabla N_j \phi_j$,
     $K_i = \frac{F\nabla\phi \cdot n_i}{2|\nabla\phi|}, \quad \delta\phi = \sum_{l=1}^{3} K_l\phi_l, \quad \delta\phi_i = K_i^+ \sum_{l=1}^{3} K_l^-(\phi_i - \phi_l) / \sum_{l=1}^{3} K_l^-)$,
     $\alpha_i = \max(0,\ \delta\phi_i/\delta\phi) / \sum_{l=1}^{3} \max(0,\ \delta\phi_l/\delta\phi)$,
     $\phi_i^* = \phi_i^* + \alpha(\delta\phi - F\mathrm{meas}(T)),\ \omega_i = \omega_i + \alpha_i\mathrm{meas}(T)$.
3. For each $v_i \in V$,
     $\phi_i^{n+1} = \phi_i^n - \Delta\tau(\phi_i^*)^n/\omega_i^n$.
```

where $H'_\epsilon(\phi)$ is the derivative for the smoothed Heaviside function $H_\epsilon(\phi)$ (3.2), and λ_i is given by

$$\lambda_i = \frac{-\int_{\Omega_i} H'_\epsilon(d^0)(\tilde{d}^k - d^0)dx}{\int_{\Omega_i} H'_\epsilon(d^0)^2 dx}. \tag{3.5}$$

4 Moving mesh method

4.1 Fixed mesh solution

Let $\mathbf{V} \subset H_0^1(\Omega)^2$ and $P \subset L_0^2(\Omega)$ be two finite element spaces. We use a standard mixed finite element approach to discretize the governing equation (2.4) together with the divergence free constraint. More precisely, we need to find a pair $(\mathbf{u}, p)$ in the space $\mathbf{V} \times P$ such that

$$\begin{aligned}
\frac{1}{\Delta t}(\rho^n u^{n+1}, \psi) + \frac{1}{\mathsf{Re}}[(2\mu^n u_x^{n+1}, \psi_x) + (\mu^n u_y^{n+1} + \mu^n v_x^{n+1}, \psi_y)] - (p^{n+1}, \psi_x) \\
= \frac{1}{\Delta t}(\rho^n u^n, \psi) - (\rho^n \mathbf{u}^n \cdot \nabla u^n, \psi) + (\frac{1}{\mathsf{We}}\kappa\delta\phi_x, \psi) \qquad (4.1a) \\
\frac{1}{\Delta t}(\rho^n v^{n+1}, \psi) + \frac{1}{\mathsf{Re}}[(\mu^n u_y^{n+1} + \mu^n v_x^{n+1}, \psi_x) + (2\mu^n v_y^{n+1}, \psi_y)] - (p^{n+1}, \psi_y) \\
= \frac{1}{\Delta t}(\rho^n v^n, \psi) - (\rho^n \mathbf{u}^n \cdot \nabla v^n, \psi) + (\frac{1}{\mathsf{We}}\kappa\delta\phi_y, \psi) - (\frac{1}{\mathsf{Fr}}\rho^n, \psi) \qquad (4.1b) \\
(q, u_x^{n+1} + v_y^{n+1}) = 0, \qquad (4.1c)
\end{aligned}$$

for any $\psi \in \mathbf{V}$ and $q \in P$. In other words, we use a very typical finite element discretization to solve the 2D incompressible Navier-Stokes equations.

4.2 Moving mesh strategy

Assume that at time $t = t_{n+1}$ a finite element solution $(\mathbf{u}_h^{n+1}, p_h^{n+1})$ and ϕ_h^{n+1} are obtained using the methods described in Section 4.1 and Section 3, respectively. These solutions are obtained in the old mesh, i.e., the mesh used at $t = t_n$. Now the question is how to obtain a new mesh $\mathcal{T}_h^{n+1}$ using $(\mathbf{u}_h^{n+1}, p_h^{n+1}, \phi_h^{n+1})$ and $\mathcal{T}_h^n$. Moreover, after obtaining the new mesh, the finite element solutions at $t = t_{n+1}$ available on $\mathcal{T}_h^n$ need to be redistributed on $\mathcal{T}_h^{n+1}$.

The outline of the mesh generation scheme is as follows.

• Step 1: minimization of a mesh functional. We redistribute interior domain and boundary points by solving the following optimization problem:

$$\begin{aligned} &\min \sum_k \int_\Omega G^{ij} \frac{\partial \xi^k}{\partial x^i} \frac{\partial \xi^k}{\partial x^j} d\vec{x} \\ &s.t. \quad \xi|_{\partial\Omega} = \xi_b \in K \end{aligned} \tag{4.2}$$

where the inverse of the matrix (G^{ij}) is called monitor function which is in general dependent on the solution $(\mathbf{u}_h^{n+1}, p_h^{n+1}, \phi_h^{n+1})$, K is a mapping set from the physical boundary $\partial\Omega$ to the computational boundary $\partial\Omega_c$, where the mapping keeps the geometrical character of the physical domain unchanged during the boundary grid redistribution. The minimization procedure is well described in Li et al. [17].

• Step 2: grid redistribution. Denote the initial (fixed uniform) mesh in the logical domain as $\mathcal{T}_c$ (with nodes $\mathcal{A}^{(0)}$), and the new logical mesh obtained by solving (4.2) as $\mathcal{T}_c^*$ (with nodes $\mathcal{A}^*$). Their difference

$$\delta\mathcal{A} = \mathcal{A}^{(0)} - \mathcal{A}^* \tag{4.3}$$

is used to determine the displacement δX_i in the physical domain. Then select a suitable ratio-parameter μ, and move the old mesh in the physical domain to a new one by using

$$X_i^{(n+1)} = X_i^{(n)} + \mu\delta X_i. \tag{4.4}$$

Again, the details of this grid redistribution procedure can be found in Li et al. [17].

• Step 3: solution interpolation. After obtaining the new mesh X_i^{n+1}, the finite element solutions $(\mathbf{u}_h^{n+1}, p_h^{n+1}, \phi_h^{n+1})$ at the mesh X_i^n need to be interpolated to the new mesh. Here we follow a procedure proposed in [11] where a moving mesh strategy was proposed to solve the incompressible Navier-Stokes equations. The basic idea is to maintain the solution surface of $\mathbf{u_h}$ on Ω unchanged. To this end, we consider a homotopy between $u_h(t_n)$ (the solution defined on the finite element space $V_h(t_n)$ at $t = t_n$) and $u_h(t_{n+1})$ (the solution defined on the finite element space $V_h(t_{n+1})$ at $t = t_{n+1}$), i.e., $\forall v_h(x;t) \in V_h(t)$

$$\left(\frac{\partial u_h}{\partial t}, v_h(x;t) \right) =< L_h(u_h), v_h(x;t) >, \tag{4.5}$$

where L_h is the corresponding spatial differential operator. By also adding the level set function, the desired solution interpolation can be realized by solving the system

$$\rho\left(\frac{\partial \mathbf{u}}{\partial \tau}-\nabla_{\vec{x}}\mathbf{u}\cdot\delta\vec{x}\right)=-\nabla p, \tag{4.6a}$$

$$\nabla_{\vec{x}}\cdot\mathbf{u}=0, \tag{4.6b}$$

$$\frac{\partial \phi}{\partial \tau}-\nabla_{\vec{x}}\phi\cdot\delta\vec{x}=0. \tag{4.6c}$$

We now give some necessary details for Step 1 above. We assume that ξ maps a linear boundary segment on $\partial\Omega$ to a linear segment on $\partial\Omega_c$. Such assumption leads the constraint in (4.2) to a linear system. More precisely, we define the energy for a map $\xi=\xi(\mathbf{x})$ as

$$E\left(\xi\right)=\sum_k\int_\Omega G^{ij}\frac{\partial\xi^k}{\partial x^i}\frac{\partial\xi^k}{\partial x^j}dx, \tag{4.7}$$

It can be demonstrated that the extreme of the functional (4.7) is a harmonic mapping in the interior of Ω. Moreover, solving problem (4.2) is equivalent to solving a linear constrained optimization problem

$$\begin{aligned}\min_{\xi}\quad & E\left(\xi\right)\\ s.t.\quad & \xi\left(\partial\Omega\right)=\partial\Omega_c.\end{aligned} \tag{4.8}$$

We will demonstrate that the above problem leads to a linear system. Use piecewise linear discretion, $\xi_h^k=\xi_i^k\lambda^i$, where a standard summation convention is used. Let Ξ_{inner}^k and Ξ_{bound}^k be the interior and boundary part of $\Xi^k=\left(\xi_i^k\right)_{1\le i\le N}$, respectively, and denote

$$H=\left(H^{ij}\right),\quad H^{ij}=\int_\Omega G^{\alpha\beta}\frac{\partial\lambda^i}{\partial x^\alpha}\frac{\partial\lambda^j}{\partial x^\beta}dx,\quad 1\le i,j\le N. \tag{4.9}$$

We further split the matrices H into the following form:

$$H=\left(\begin{array}{cc}H_{11} & H_{12}\\ H_{21} & H_{22}\end{array}\right),$$

where the subscripts 1 and 2 denote the rows and columns corresponding to the interior and boundary nodes, respectively. Assume that the constraint leads to a linear system of the form $\sum_k A_k\Xi^k=b$, or equivalently $\sum_k A_{k,inner}\Xi_{inner}^k+A_{k,bound}\Xi_{bound}^k=b$.

With the above preparation, the optimization problem (4.2) is equivalent to

$$\begin{aligned}\min_{\Xi}\quad & \sum_k\left\{\Xi^{k,T}H\Xi^k\right\}\\ s.t.\quad & \textstyle\sum_k A_k\Xi^k=b.\end{aligned} \tag{4.10}$$

Observe that $A_{k,inner}=0$. We then have

$$\begin{aligned}\min_{\Xi}\quad & \sum_k\left\{\Xi_{inner}^{k,T}H_{11}\Xi_{inner}^k+2\Xi_{bound}^{k,T}H_{21}\Xi_{inner}^k+\Xi_{bound}^{k,T}H_{22}\Xi_{bound}^k\right\}\\ s.t.\quad & \textstyle\sum_k A_{k,bound}\Xi_{bound}^k=b.\end{aligned} \tag{4.11}$$

4.3 Monitor function

It is very important to choose an appropriate monitor function, otherwise the adaptive effect can not be realized no matter how good a moving mesh algorithm is. For problems with free interfaces, the singularity often occurs around the interface where more grid points are required. Away from the interface, it is hoped that the grids are as uniform as possible. Hence, it is essential that the monitor function should be inversely proportional to the values of the level set function. It is also noted that the moving interface can develop corners, cusps, and undergo topological changes. To cluster more points around these places, some information on κ, the curvature of the level set function as defined in (2.2), should be included. To this end, the following monitor function is proposed and used:

$$G = \frac{1 + \beta|\kappa|/\max|\kappa|}{1 + \alpha\,|\phi|/\max|\phi|}, \tag{4.12}$$

where β and α are user-defined positive constants to determine the density of the mesh. Roughly speaking, more points will be moved to areas with large curvature or/and small level set values.

4.4 Numerical procedure

We describe the actual numerical discretization described in the previous sections. The outline of our scheme is as follows.

Given the velocity $\mathbf{u}^n$ and the level set function ϕ^n at time t^n. We are also given the density $\rho^n = \rho(\phi^n)$, viscosity $\mu^n = \mu(\phi^n)$ and Heaviside function $H^n = H(\phi^n)$. We discretize the problem using the following steps:

1. **Level set update for ϕ^{n+1}:**

   $$\phi_t + \mathbf{u}^n \cdot \nabla\phi = 0.$$

   using the explicit positive coefficient scheme as described in Section 3. We advance in time using third order TVD Runge-Kutta methods formula in [27].
2. **Re-distance step for ϕ^{n+1}.** We maintain the level set function ϕ as the signed normal distance to the free surface. Details of the resistance step are presented in Section 3.
3. **Solve the Navier-Stokes equations to get $\mathbf{u}^{n+1}$.** This is done by using of the mixed finite element methods as described in Section 4.1, by using the new level set function ϕ^{n+1}.
4. **Mesh motion and solution interpolation.** Compute the monitor function, move mesh and update the velocity and the level set function on the new mesh, as described in Section 4.2.

The last step above is in fact an iteration step, so it in general requires a few iterations at each time step (see, e.g., [11, 17]). However, for the numerical computations in this work, only one iteration is sufficient to obtain a satisfactory mesh at each time level except at the initial stage where the number of iterations depends on the degree of singularity of the initial data.

5 Numerical examples

In this section, we will apply our numerical method to several problems. The first one is the gas bubble bursting at a free surface, the second one is the merging of two fluid bubbles, and the last one is the impact of a water droplet on a pool. In the numerical examples, we use the following constants unless otherwise specified: $g = -9.8m/s^2$, $\sigma = 0.0728kg/s^2$, $\rho_{water} = 1000kg/m^3$, $\mu_{water} = 1.137 \times 10^{-3}kg/ms$, $\rho_{air} = 1.226kg/m^3$ and $\mu_{air} = 1.78\times10^{-5}kg/ms$. We consider flows in a unit square domain Ω and assume the flow is periodic in the x-direction. The boundary conditions at $y = 0$ and $y = 1$ are assumed to be no-slip and no-flow boundary conditions; i.e., $u = 0$, $v = 0$.

The purpose of our numerical experiments is to demonstrate that detailed inter-facial structures during a topological transition can be captured accurately with small number of grid points using our moving mesh method. For this reason, we have performed a series of resolution studies. The numerical solutions we present here are mostly carried out using a 80×80 grid. When we compare these solutions with the uniform 250×250 grid calculations, they are almost indistinguishable. Moreover, the parameters in the monitor function (4.12) are set to $\beta = 4$ and $\alpha = 10$ in our numerical experiments.

5.1 Gas bubble bursting at a free surface

We consider the problem of a gas bubble rising to the free surface of a liquid. Above the fluid, the air is same as that in the bubble. When a bubble bursts at a free surface, the surface tension rapidly pulls the rim where they intersect outward and downward. Eventually, a ring of fluid at the base of what was the bubble contracts to a point, throwing a plume of fluid upward in the form of a high-speed jet. Various aspects of this motion have been studied experimentally (see, e.g., [20]) and numerically (see, e.g., [5, 21, 12]). Previous numerical studies have used boundary integral methods [5, 6], VOF-type method [26], and the so-called marker-chain approach [12].

Fig. 1 illustrates the computational results of the jet formation in a 3mm bubble bursting at a free surface obtained on a 80^2 grid. The parameters used in Eq. (2.4) are $\mathsf{Re} = 6649$, $\mathsf{Fr} = 1$, and $\mathsf{We} = 44$. It is found that the the moving mesh results obtained with a 80^2 grid and a 250^2 grid are graphically indistinguishable, indicating that the moving mesh scheme with a 80^2 grid can resolve the gas bubble problems. To see the moving mesh effect, we plot in Fig. 2 the corresponding mesh at the final time, $t = 1.0$. It is seen that

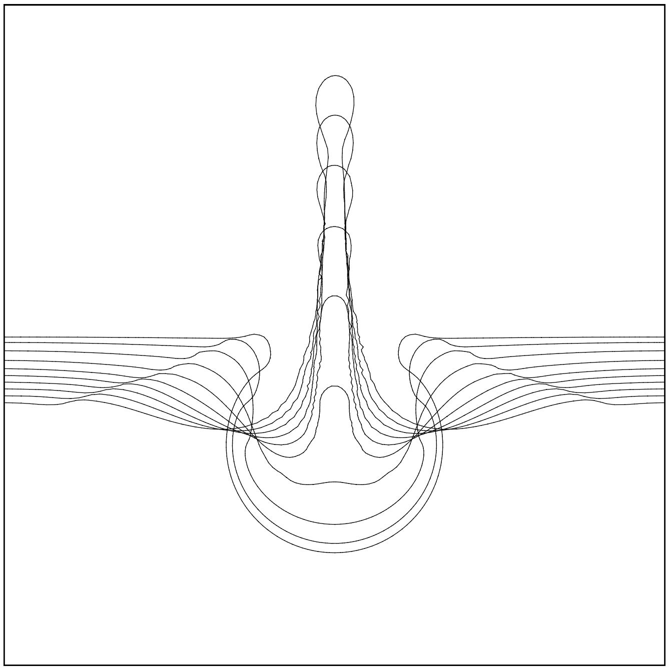

Figure 1: Time sequence of the jet formation in a $3mm$ bubble bursting at a free surface. Profiles are 0.1 apart, from $t = 0.1$ to 1.0. A 80^2 moving grid is used.

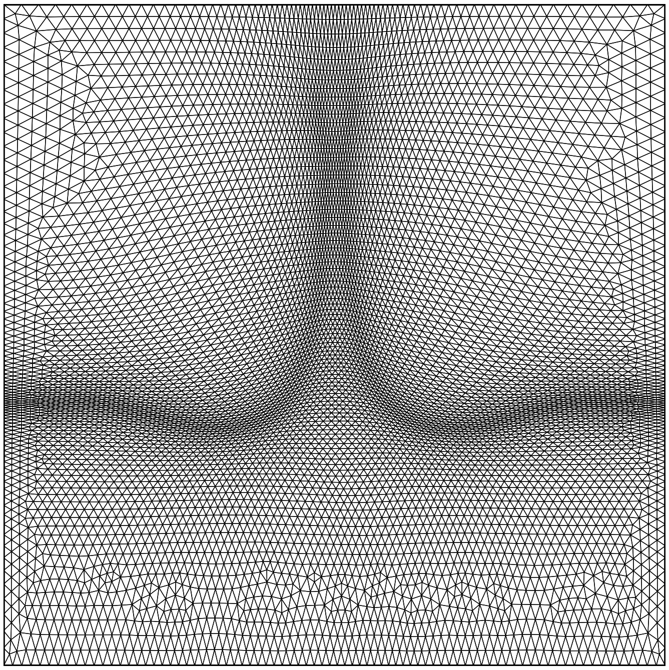

Figure 2: Gas bubble bursting at a free surface: the mesh at $t = 1.0$ at a 80^2 moving grid.

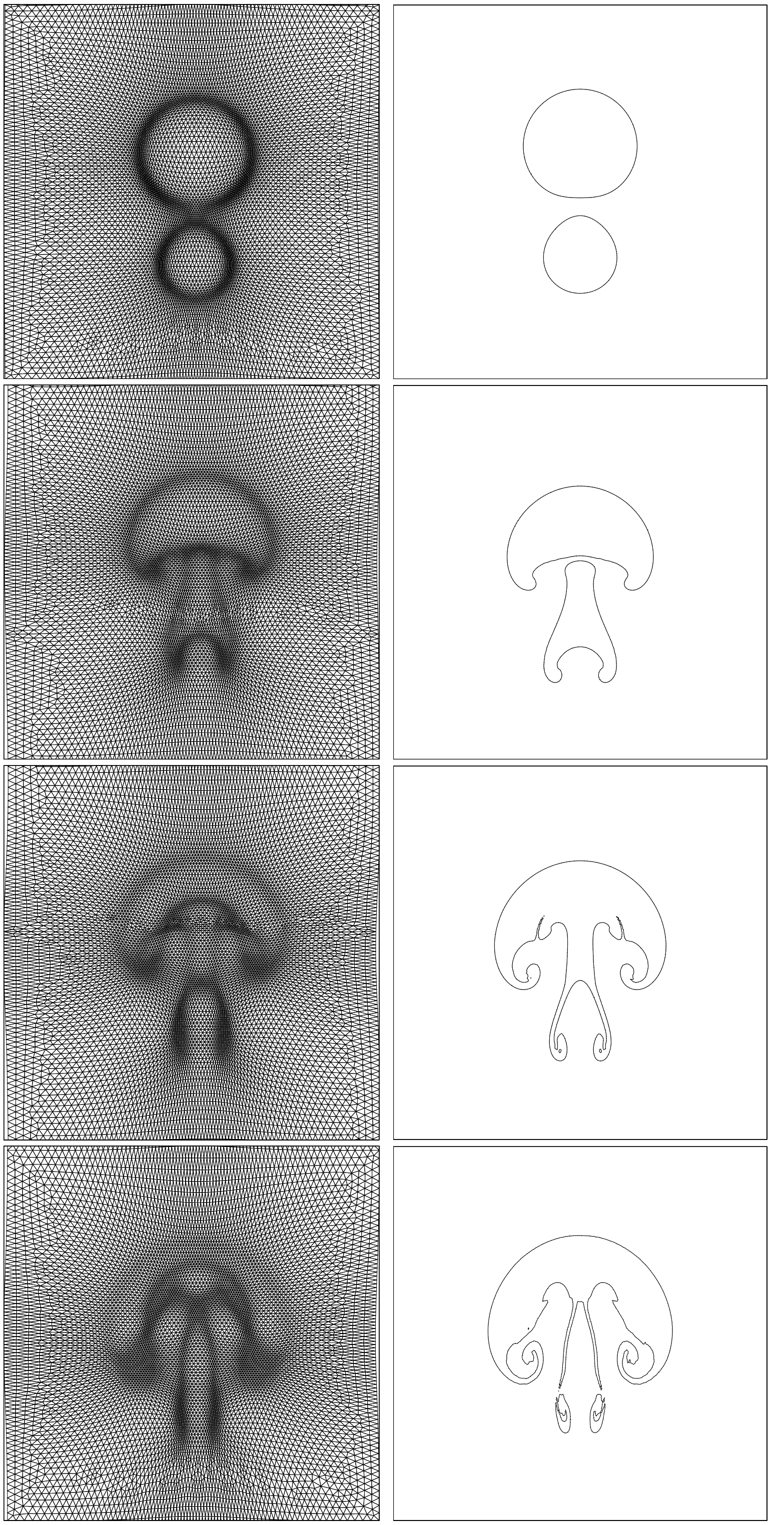

Figure 3: Merging of two bubbles in moving 80×80 mesh. The density ratio between the bubbles and the background is $1 : 10$, and $\mu_l = 0.0005, \mu_g = 0.00025, \sigma = 0$. $t = 0.1, 0.3, 0.4, 0.5$.

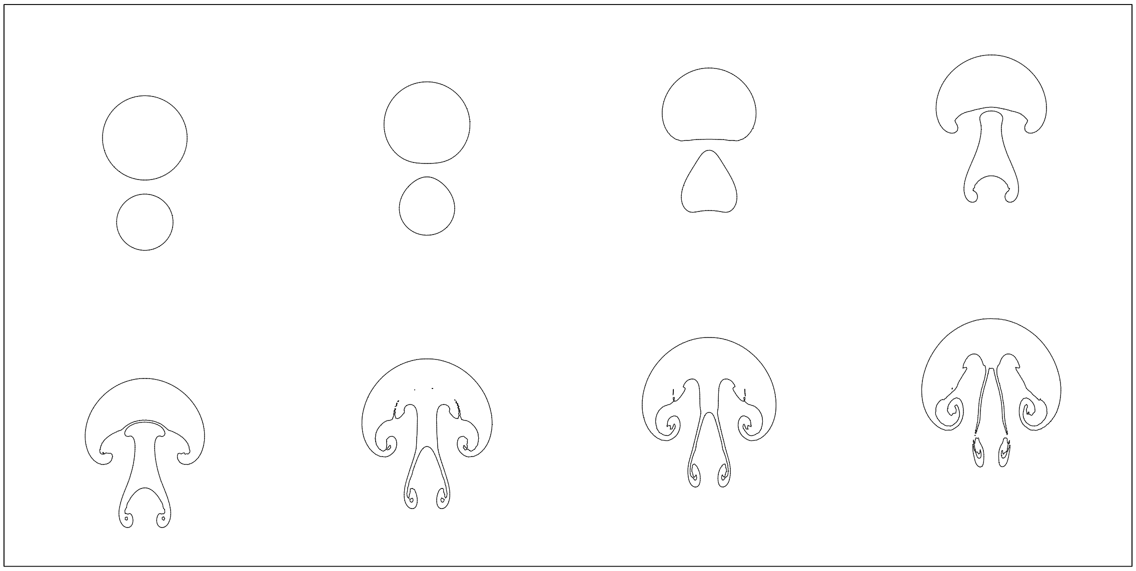

Figure 4: Merging of two bubbles in moving 80×80 mesh. The density ratio between the bubbles and the background is $1:10$, and $\mu_l = 0.0005, \mu_g = 0.00025, \sigma = 0$. $t = 0, 0.1, 0.2, 0.3$ for the first row, and $t = 0.35, 0.4, 0.45$ and 0.5 for the second row.

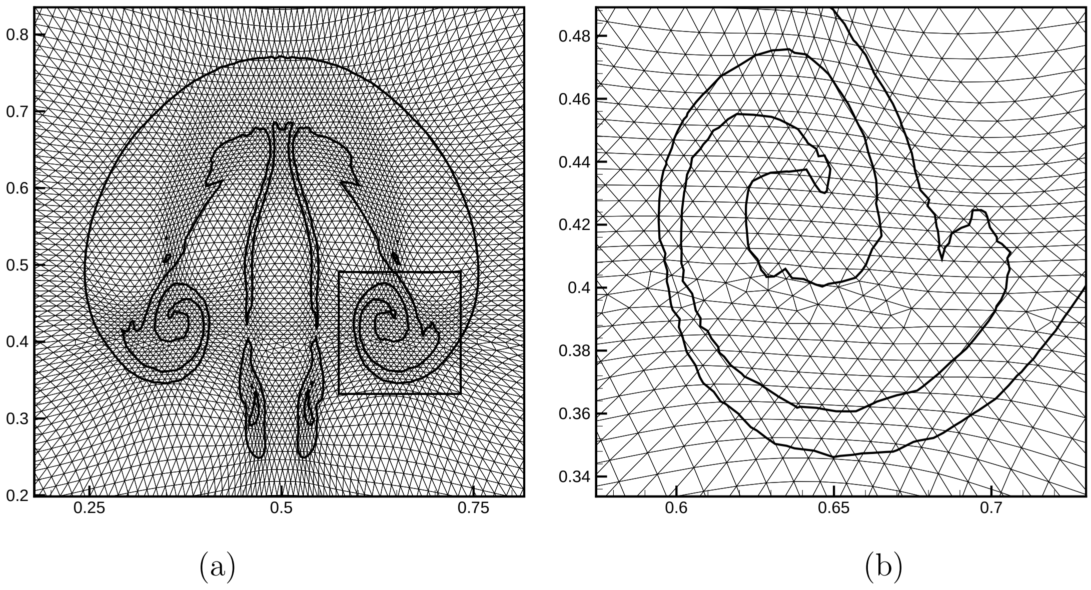

Figure 5: (a): Close-up of the adaptive mesh (80×80) around the roll of two bubbles merging at $t = 0.55$. (b): The boxed region in the left figure is magnified in the right figure.

more grid points are clustered around the interface where the level set function is zero.

5.2 Merging two bubbles with the same density

In this subsection, we compute the interaction of two fluid bubbles of the same density under the influence of gravity. The fluid is set at rest initially. The viscosity for the fluid inside and outside the two bubbles is equal to $\mu = 0.00025$ and 0.0005, respectively. The surface tension is set to be zero. The initial positions of the two bubbles correspond to two circles, with the lower one centered at $(0.5, 0.35)$ with radius 0.1 and the upper one centered at $(0.5, 0.65)$ with radius 0.15. We take the density inside and outside the two bubbles to be 1 and 10, respectively.

In Fig. 3, we plot the numerical solutions together with the corresponding meshes at $t = 0.1, 0.3, 0.4$ and 0.5, obtained by using a 80^2 grid. The desired effect of the mesh adaptivity can be clearly seen in this figure. The time sequence of the merging of the two bubbles is plotted in Fig. 4. In [10], the level set method together with a second-order projection scheme was used to study the merging of two bubbles with the above parameters. The overall agreement between our coarse mesh results and the fine mesh results of [10] (where a 256^2 uniform grid is employed) is very satisfactory.

`Monitor functions.` Fig. 5 shows a close-up of the dynamically adaptive mesh of the two bubbles merging at $t = 0.55$. On the one hand the level set function used in the monitor makes the adaptive mesh to follow closely the interface dynamics; on the other hand, the curvature term helps to move the grid points to the regions with complex structures, i.e., very singular corners and cusps. As a result, the interface is effectively resolved even when it becomes extremely localized and singular.

`Time steps for moving mesh method.` The time steps used for computing this problem are found interesting. In general, moving mesh approach has to use smaller time steps compared with the uniform mesh approach when both approaches have the same smallest mesh scale (the smallest circumradius of the triangles), see, e.g., [30, 31]. However, an exception is observed in this example. In Fig. 6, the dotted line is the time step on a 250^2 uniform grid and the solid line is the one on a 80^2 moving grid. The mesh scale of both grids are all about $3.5 * 10^{-3}$. However, it is observed from Fig. 6 that the time step for the moving mesh is larger than that for the uniform mesh. The possible reason for this is as follows. For the bubble merging problem, the time step is dominated by the CFL condition for the convection term, so the time step is proportional to the value of $\min_j\{\Delta x_j/|u|_j\}$, where Δx_j and u_j are representative mesh size and velocity in the j-th cell. In this problem, the area with the largest velocity is near the exterior interface of the big bubble where the mesh size is relatively large, while the area with the smallest mesh is in the interior interface where the velocity is not too large. Consequently, we end up with larger ration $\min_j\{\Delta x_j/|u|_j\}$ for the moving mesh methods. Consequently, with the same mesh scale the time step used for the moving mesh methods is larger than that of the uniform computation.

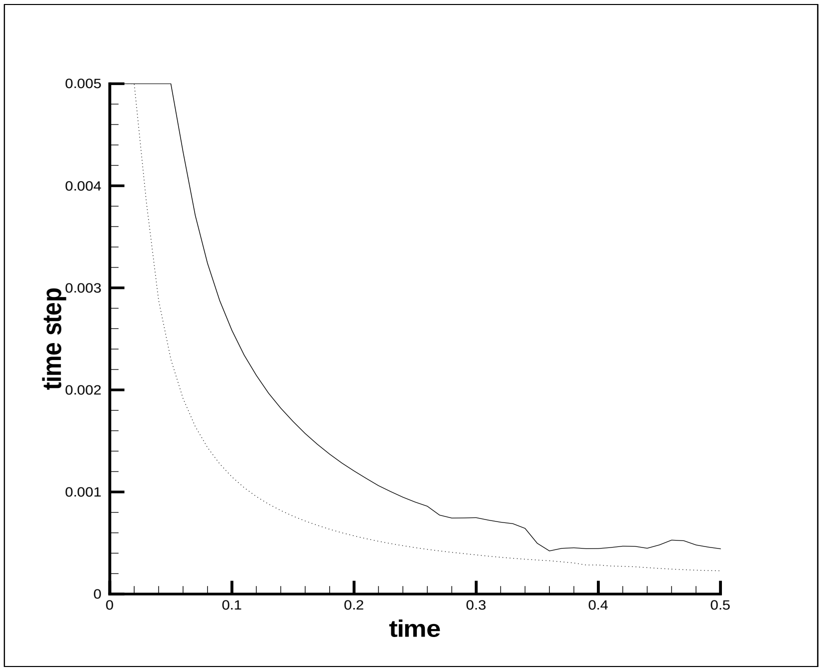

Figure 6: Time steps used for the bubble merging problems: dotted line is the time step on a 250^2 uniform mesh and the solid line is on a 80^2 moving mesh; both meshes have same value of the smallest mesh scale.

5.3 Water droplet impact

Liquid drop impact on liquid and solid surfaces has important engineering and biological implications. Ink-jet printing and the dispersal of fungal spores are examples of two processes which are governed by drop impact dynamics. We compute the impact of a water droplet on a pool of water along with the "splash" that comes afterward. In our computations, we use dimensionless parameters based on the impact velocity U and the radius of the drop R. In Fig. 7, we show results using $R = 1mm$ and $U = 4.0m/s$. For that we have $\mathsf{Re} = 28144$, $\mathsf{Fr} = 204$, and $\mathsf{We} = 1760$. The dimensionless impact velocity is 1; we accelerate the drop with a fictitious gravitational force term $1/\mathsf{Fr} = 4$ for a total dimensionless time 0.25. At dimensionless time $t = 0.25$, the drop will be traveling with dimensionless speed of 1 and begin to merge into the pool.

The study of drop impact on liquid surfaces has a long history, see, [15, 14, 22] and references therein. In [15, 14], Harlow and Shannon studied drop impact on a liquid layer using the MAC method. Their calculations were very inspiring. Sussman etc [29] studied this problem using an adaptive level set method with projection method.

In Fig. 7, we plot the evolution of the water drop at time $t = 0.25, 0.3, 0.46, 0.67$, using a 100^2 moving grid. To test the accuracy, the problem is also computed on an 250^2 uniform grid. It is found that the finer mesh results are graphically indistinguishable with the moving mesh results on the coarse grid.

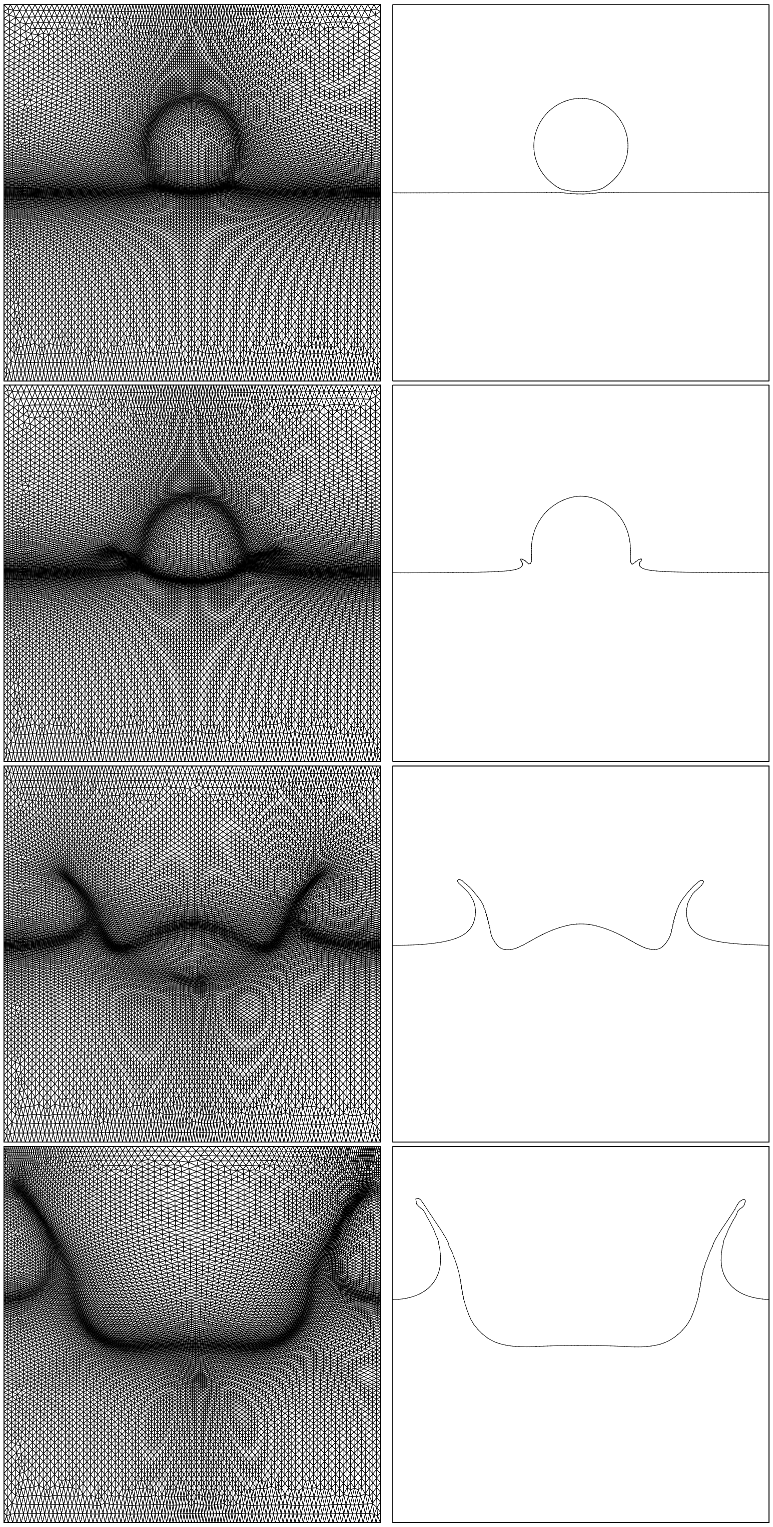

Figure 7: Impact of water drop with an 100^2 moving grid. The parameters used are $\mathsf{Re} = 28144, \mathsf{Fr} = 204, \mathsf{We} = 1760, \rho_g/\rho_l = 1/816, \mu_g/\mu_l = 1$; $t = 0.25, 0.3, 0.46, 0.67$.

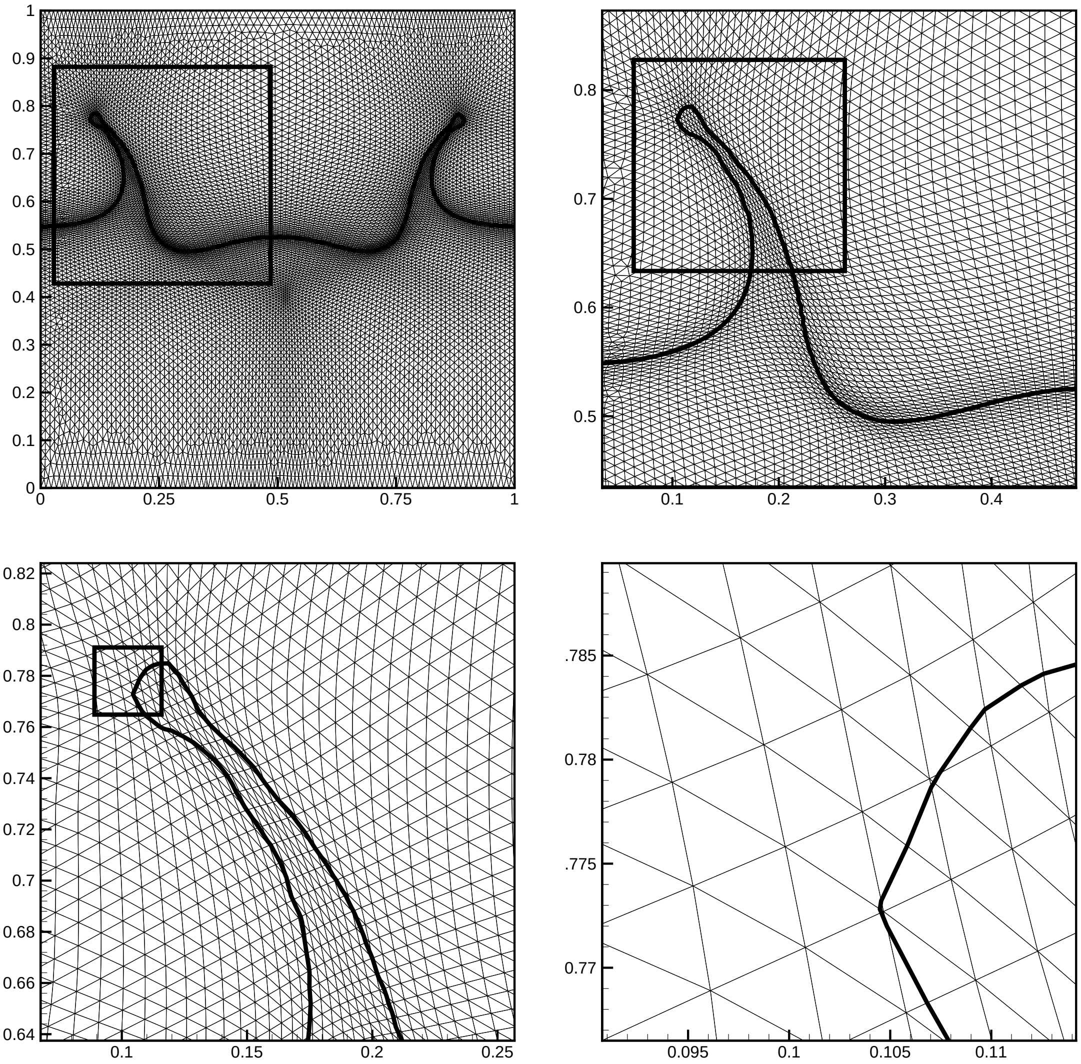

Figure 8: Computational mesh from the simulation of droplet impact shown in Fig. 7 at time $t = 0.6$. Each of the first three figures has a boxed region that is magnified in the next figure (left to right, top to bottom).

6 Concluding remarks

In this work, the level set method and the moving mesh technique is combined to form an efficient algorithm in solving the incompressible two-phase flow problems. To show the combining approach works, two issues are considered. First, we demonstrate that the moving mesh results on coarse grids are comparable with the uniform mesh results on finer grids. Secondly, it is demonstrated that to obtain the same resolution the moving mesh level set approach can save the internal memories and reduce the overall CPU time. Moreover, the codes related to our moving mesh level set method are simpler than those

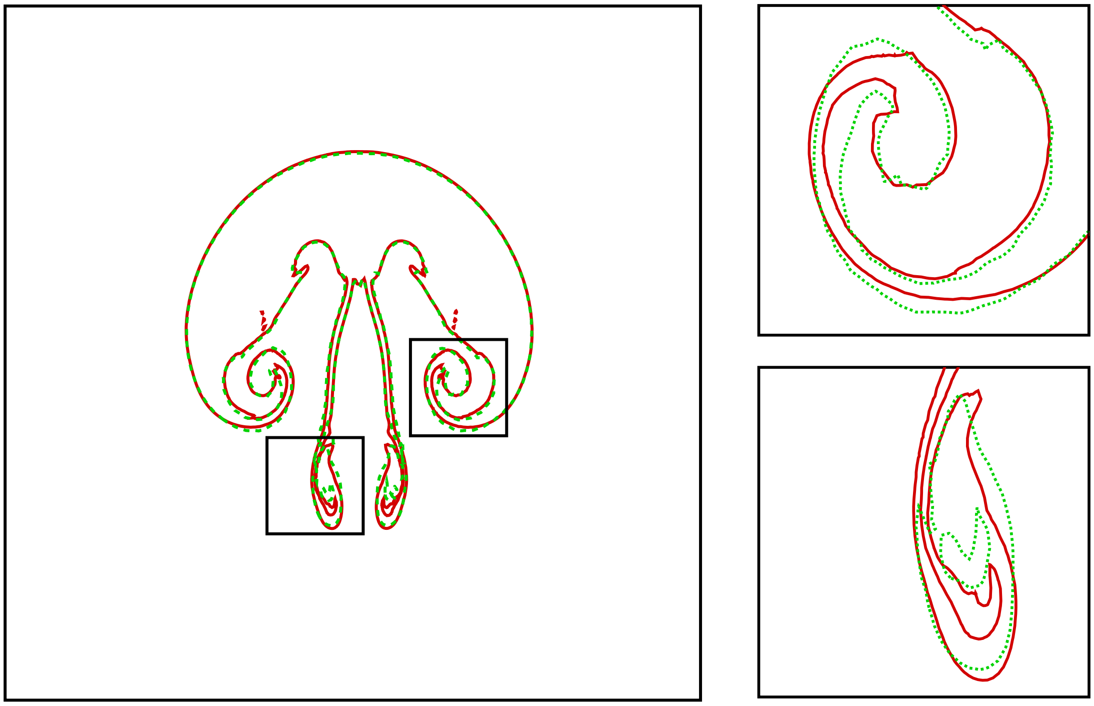

Figure 9: The comparison for simulations of merging of two bubbles between moving 80 × 80 mesh with $\epsilon = 0.01$ (solid line) and moving 160×160 mesh with $\epsilon = 0.005$ (dashed line). The two boxed regions in the left figure are magnified in the right two figures. The density ratio between the bubbles and the background is 1 : 10, and $\mu_l = 0.0005$, $\mu_g = 0.00025$, $\sigma = 0$, $t = 0.5$.

for the h-type adaptive level set approach developed in [29].

We close this work by discussing two issues related to the moving mesh level set computations performed in this work.

<u>`The size of the smoothing parameter.`</u> The first issue is about the size of the smoothing parameter ϵ which is used to regularize the delta function. When a topological change takes place in a free interface, it signals the formation of a singularity. In that case, it is expected that the numerical smoothing parameter would have a strong effect for the fine structure of the interface. Usually we relate this parameter and the grid size parameter h by the relation $\epsilon = ch$. For example, $\epsilon = 1.5h$ and $\epsilon = 2.5h$ are used in [28] and [10], respectively. In [29], $\epsilon = 3h$ is used for the gas bubble problem and $\epsilon = 2h$ for the water drop problem. In our computations, it was found that different inter-facial thickness ϵ produces similar qualitative results; but they may induce different detail information during the topological transition. This is demonstrated in Fig. 9 where numerical results for merging of two bubbles are plotted using a 80^2 resolution and $\epsilon = 0.01$ (solid line) and a 160^2 resolution and $\epsilon = 0.005$ (dashed line). The two boxed regions in the left figure are magnified in the right two figures. Note that the fine structure of two interfaces are quite

different. However, the general structures produced by the two pair of parameters are very similar. Since we are interested in capturing the fine scale structure of the physical solution and the comparison among the results with different meshes, we fix a small enough value of ϵ directly (just like in [9]).

`Memory savings and speed-up.` The second issue is about the real gaining of using the moving mesh level set method. For the gas bubble problem considered in Section 5.1, by comparing the moving mesh computation on a 80^2 grid and the uniform mesh computation on a 250^2 grid (both yield comparable resolution) we found that the memory savings for using the moving mesh method is 1:10 and the speed-up is about 4.3. For the water droplet computations considered in Section 5.3, the memory savings for using the moving mesh method is 1:6.5 and the speed-up is about 2.5. It is found that the speed-up is less than half of the internal memory savings. This phenomena was also observed in [29] where an adaptive projection method is used to solve the two-phase flow problems. For computing the water drop problem (Section 5.3, [29]), it was found that the internal memory savings for using an adaptive grid was 1:4 while the speed-up was only 1.7.

Acknowledgment

The research of Di and Li was supported in part by the Joint Applied Mathematics Research Institute between Peking University and Hong Kong Baptist University. The research of Tang was supported in part by Hong Kong Research Grants Council, the International Research Team on Complex System of Chinese Academy of Sciences and an NSAF Grant (#10476032) of National Science Foundation of China. The research of Zhang was supported in part by the special funds for Major State Research Projects and National Science Foundation of China for Distinguished Young Scholars.

References

[1] A. Anderson and X. Zheng and V. Cristini, Adaptive unstructured volume remeshing - I: The method, *J. Comput. Phys.*, **208**, 616-625 (2005).

[2] G. Beckett and J.A. Mackenzie and M.L. Robertson, An r-adaptive finite element method for the solution of the two-dimensional phase-field equations, To appear in *Commun. Comput. Phys.*, 2006.

[3] T.J. Barth and J.A. Sethian, Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains, *J. Comput. Phys.*, **145**, 1-40 (1998).

[4] J.U. Brackbill, D.B. Kothe and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.*, **100**, 335 (1992).

[5] J.M. Boulton-Stone and J.R. Blake, Gas bubble bursting at a free surface, *J. Fluid Mech.*, **254**, 437-466 (1993).

[6] J.M. Boulton-Stone, The effects of surfactants on bursting gas bubbles, *J. Fluid Mech.*, **302**, 231 (1995).

[7] W.M. Cao, W.Z. Huang and R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, *J. Comput. Phys.*, **149** (1999), pp. 221–244.

[8] W.M. Cao, W.Z. Huang and R.D. Russell, An error indicator monitor function for an r-adaptive finite-element method, *J. Comput. Phys.*, **170** (2001), pp. 871-892.

[9] H.D. Ceniceros, and T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, *J. Comput. Phys.*, **172**, 609-639 (2001).

[10] Y.C. Chang, T.Y. Hou, B. Merriman and S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows *J. Comput. Phys.*, **124**, 449-464 (1996).

[11] Y. Di, R. Li, T. Tang and P.W. Zhang, Moving mesh finite element methods for the incompressible Navier-Stokes equations, *SIAM J. Sci. Comput.*, **26**, 1036-1056 (2005).

[12] L. Duchemin, S. Popinet, C. Josserand and S. Zaleski, Jet formation in bubbles bursting at a free surface, *Phys. Fluids*, **14**, 3000-3008 (2001).

[13] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, *J. Comput. Phys.*, **95**, 450-476 (1991).

[14] F.H. Harlow and J.P. Shannon, Distortion of a liquid drop, *Science*, **157**, 547-550 (1967).

[15] F.H. Harlow and J.P. Shannon, The splash of a liquid drop, *J. Appl. Phys.*, **38**, 3855-3866 (1967).

[16] R. Li, T. Tang and P.W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, *J. Comput. Phys.*, **170**, 562-588 (2001).

[17] R. Li, T. Tang and P.W. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, *J. Comput. Phys.*, **177**, 365-393 (2002).

[18] K. Lipnikov and M. Shashkov, The error-minimization-based strategy for moving mesh methods. *Commun. Comput. Phys.*, **1**, 53-80 (2006).

[19] W.B. Liu and T. Tang, Error analysis for a Galerkin-spectral method with coordinate transformation for solving singularly perturbed problems, *Appl. Numer. Math.*, **38**, 315-345 (2001).

[20] F. MacIntyre, Flow patterns in breaking bubbles. *J. Geophys. Res.*, **77**, p. 5211 (1972).

[21] H.Z. Oguz, The role of surface disturbance on the entrainment of bubbles, *J. Fluid Mech.*, **372**, 189-212 (1998).

[22] H.N. Oguz and A. Prosperetti, Bubble entrainment by the impact of drops on liquid surfaces, *J. Fluid Mech.*, **219**, 143-179 (1990).

[23] S.O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.*, **100**, 25 (1992).

[24] M.S. Longuet-Higgins and E.D. Cocklet, Deformation of steep surface waves on water I: A numerical method of computation, *Proc. R. Soc. Lond. A.*, **350**, 1 (1975).

[25] S. Osher, and J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, **79**, 12-49 (1988).

[26] M. Rudman, A volume-tracking method for incompressible multi-fluid flows with large density variations, *Int. J. Numer. Methods Fluids*, **28**, 357 (1998).

[27] C.W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, *J. Comput. Phys.*, **83**, 32-78 (1989).

[28] M. Sussman, P. Smereka, and S.J. Osher, A level set approach for computing solutions to incompressible tow-phase flows, *J. Comput. Phys.*, **114**, 146-159 (1994).

[29] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell and M. Welcome, An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, **148**, 81-124 (1999).

[30] Z.J. Tan, Z.R. Zhang, Y.Q. Huang and T. Tang, Moving mesh methods with locally varying time steps, *J. Comput. Phys.*, **200**, 347-367 (2004).

[31] H.Z. Tang and T. Tang, Moving mesh methods for one- and two-dimensional hyperbolic conservation laws, *SIAM J. Numer. Anal.*, **41**, 487-515 (2003).

[32] H.Z. Tang, T. Tang and P.W. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton-Jacobi equations in two- and three dimensions, *J. Comput. Phys.*, **188**, 543-572 (2003).

[33] T. Tang and M.R. Trummer, Boundary layer resolving pseudo-spectral methods for singular perturbation problems, *SIAM J. Sci. Comput.*, **17**, 430-438 (1996).

[34] A. Winslow, Numerical solution of the quasi–linear Poisson equation, *J. Comput. Phys.*, **1**, 149-172 (1967).

[35] P.A. Zegeling, On resistive MHD models with adaptive moving meshes, *J. Sci. Comput.*, **24**, 263-284 (2005).