

An adaptive phase field method for the mixture of two incompressible fluids

Zhengru Zhang^a, Huazhong Tang^{b,*}

^a *School of Mathematical Sciences, Beijing Normal University, Beijing 100875, PR China*

^b *LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, PR China*

Received 26 June 2006; received in revised form 24 October 2006; accepted 13 December 2006

Available online 21 March 2007

Abstract

This paper develops an adaptive moving mesh method to solve a phase field model for the mixture of two incompressible fluids. The projection method is implemented on a half-staggered, moving quadrilateral mesh to keep the velocity field divergence-free, and the conjugate gradient or multigrid method is employed to solve the discrete Poisson equations. The current algorithm is composed by two independent parts: evolution of the governing equations and mesh-redistribution. In the first part, the incompressible Navier–Stokes equations are solved on a fixed half-staggered mesh by the rotational incremental pressure-correction scheme, and the Allen–Cahn type of phase equation is approximated by a conservative, second-order accurate central difference scheme, where the Lagrangian multiplier is used to preserve the mass-conservation of the phase field. The second part is an iteration procedure. During the mesh redistribution, the phase field is remapped onto the newly resulting meshes by the high-resolution conservative interpolation, while the non-conservative interpolation algorithm is applied to the velocity field. The projection technique is used to obtain a divergence-free velocity field at the end of this part. The resultant numerical scheme is stable, mass conservative, highly efficient and fast, and capable of handling variable density and viscosity. Several numerical experiments are presented to demonstrate the efficiency and robustness of the proposed algorithm.

© 2007 Elsevier Ltd. All rights reserved.

1. Introduction

The hydrodynamics of the mixture of different fluids is playing an increasingly important role in many scientific and engineering applications. Among them one of the fundamental issues is the interfacial dynamics. The conventional model for the mixture consists of separate hydrodynamic systems of each component, together with the free interface that separates different fluids. In recent years, many researchers have studied the phase field approach in various fluid flows, see, e.g., [6,17,22,26]. In [22], a phase field model for the mixture of two incompressible fluids is presented based on an energetic variational formulation and solved by a Fourier-spectral method.

The phase field approach introduces a continuous transition region between the two bulk phases (e.g., solid/liquid). This transition region is defined in terms of an additional field variable (the phase field), which is formulated to represent the dynamical evolution of the phase-change interface. This is in contrast to methods which assume a sharp interface between phases. At the cost of solving an additional partial differential equation, the advantage of the phase field method is that the location of the interface does not have to be explicitly determined (or tracked) as part of the solution. However, the discrete phase field method for the interface dynamics should be of high-order accuracy as well as high-efficiency. Up to now, some numerical methods have been developed for the phase field models, for example, the discontinuous Galerkin method [35,36] and the Fourier-spectral method [22], etc. The adaptive mesh methods [28,2] have also been proved very effective for the phase field model. However, in

* Corresponding author. Tel.: +86 10 62757018; fax: +86 10 62751801.
E-mail addresses: zrzhang@bnu.edu.cn (Z. Zhang), hztang@math.pku.edu.cn (H. Tang).

those adaptive methods, the phase field model is not coupled with the Navier–Stokes equations.

Adaptive moving mesh methods have important applications in a variety of scientific and engineering areas such as solid and fluid dynamics, etc., where singular or nearly singular solutions are developed dynamically in fairly localized regions of shock waves, boundary layers, and detonation waves, etc. Numerically investigating these phenomena requires extremely fine meshes over a small portion of the physical domain to resolve the large solution variations. Successful implementation of an adaptive strategy can increase accuracy of the numerical approximations and decrease the computational cost. Up to now, there have been many important progresses in adaptive moving mesh methods for partial differential equations, including grid redistribution approach based on the variational principle of Winslow [34], Brackbill [3], Ren and Wang [25]; moving finite element methods of Miller and Miller [23], and Davis and Flaherty [7]; moving mesh PDEs methods of Russell et al. [4,16], and Ceniceros and Hou [5]; and moving mesh methods based on the harmonic mapping of Dvinsky [9], and Li et al. [19,20,8] as well as [40]. Computational costs of moving mesh methods can be efficiently saved with locally varying time steps [29]. To our knowledge, there is little or no work on adaptive moving mesh methods of the phase field model for the mixture of two incompressible fluids.

The main objective of this paper is to extend the adaptive moving mesh method developed in [30,31,39] to the Allen–Cahn type phase field model for the mixture of two incompressible fluids. The main difficulty in solving the phase field model for the mixture of two incompressible fluids is the divergence-free constraint of the velocity vector and the mass-conservation of the phase field variable. There are two popular approaches to deal with the divergence-free constraint in the incompressible Navier–Stokes equations. One is to introduce the stream function, see [14,10], but it is mainly limited to two dimensional case. The second method is to use the projection technique, see e.g., [33,22,12,13], which can be easily extended to high-dimensional problems. But both ways are known to have problem in setting up artificial boundary conditions. The Gauge method of E and Liu [11] and the Gauge–Uzawa method of Nochetto and Pyo [24] are a kind of projection methods but provide consistent boundary conditions. Besides those methods, the penalty and sequential regularization methods, see e.g., [21], are stabilization techniques for the divergence-free condition but do not require any artificial boundary condition, and their many (physical) properties such as approximation of the energy law can be relatively easy to obtain. The present work will focus on the projection method. In all computational examples the solution activates mostly inside the domain and thus artificial boundary conditions will not be very important.

The paper is organized as follows. Section 2 first introduces the Allen–Cahn phase field model for the mixture of two incompressible fluids. Section 3 is devoted to the implementation of the rotational incremental pressure-cor-

rection scheme for the incompressible Navier–Stokes equations on a fixed half-staggered quadrilateral mesh, and presents a conservative, second-order accurate central difference scheme for the Allen–Cahn type of phase equation with the Lagrangian multiplier. In Section 4, the iterative mesh redistribution is discussed. During the iterative mesh redistribution, the phase field and pressure is remapped onto the newly resulting meshes by the high-resolution conservative interpolation, while the non-conservative interpolation algorithm is applied to the velocity field. The projection method is used to obtain a divergence-free velocity at the end of the iterative mesh redistribution. Thus the iterative mesh redistribution is divergence-free for the velocity field and mass-conservative for the phase field. Full solution procedure will be outlined in Section 5, ending with several remarks. Numerical experiments are included in Section 6 to validate the robustness and efficiency of the proposed adaptive approach. Conclusions and future work are in Section 7.

2. A phase field model for the mixture of two incompressible fluids

Let Ω_p be a two-dimensional (physical) domain with the Cartesian Coordinate system $\mathbf{x} = (x, y)$, and assume that Ω_p is filled with two incompressible fluids separated by a free moving interface. One fluid is included in a bubble while the other is the ambient fluid. As in [22], a phase function $\phi(\mathbf{x}, t)$ is introduced to represent the interface at the time t by the set $\{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}$, and label the inside and the outside of the bubble as $\{\mathbf{x} : \phi(\mathbf{x}, t) > 0\}$ and $\{\mathbf{x} : \phi(\mathbf{x}, t) < 0\}$, respectively. The dynamics of the phase function $\phi(\mathbf{x}, t)$ can be relaxed (approximated) according to either Allen–Cahn or Cahn–Hilliard type of gradient flow, depending on the choice of different dissipative mechanisms. Since numerical treatment of the Allen–Cahn phase equation is simpler than that of the Cahn–Hilliard phase model which involves fourth-order differential operators, see [22], this work chooses to use the modified (fluid transported) Allen–Cahn phase equation

$$\begin{cases} \phi_t + \mathbf{u} \cdot \text{grad} \phi = \gamma(\Delta \phi - f(\phi) + \zeta(t)), \\ \frac{d}{dt} \int_{\Omega_p} \phi d\mathbf{x} = 0, \end{cases} \quad (2.1)$$

where grad and Δ are the gradient and Laplacian operators, respectively, $\zeta(t)$ is the Lagrange multiplier corresponding to the constant volume constraint in the last equation, see [37], \mathbf{u} is the fluid velocity, and the parameter γ denotes the elastic relaxation time. In Eq. (2.1), $f(\phi) = F'(\phi)$, where $F(\phi) = (\phi^2 - 1)^2/4\hat{\eta}^2$ is the usual double-well potential, the parameter $\hat{\eta}$ represents the capillary width (width of the mixing layer). As shown in [37], the phase field model (2.1) may be derived from a variation formulation of the elastic (mixing) energy as

$$\frac{D\phi}{Dt} = -\gamma \frac{\delta W}{\delta \phi} = \gamma(\Delta \phi - f(\phi)), \quad (2.2)$$

where the elastic (mixing) energy $W(\phi, \text{grad}\phi)$ is defined by

$$W(\phi, \text{grad}\phi) = \int_{\Omega_p} \left\{ \frac{1}{2} |\text{grad}\phi|^2 + F(\phi) \right\} d\mathbf{x}.$$

Here $\frac{D\phi}{Dt}$ is the material derivative $\frac{D\phi}{Dt} = \phi_t + (\mathbf{u} \cdot \text{grad})\phi$, and $\frac{\delta W}{\delta \phi}$ represents the variation of the energy W with respect to ϕ .

The system governing the mixture of two incompressible fluids with same density (which is taken to be 1) and same viscosity constants can be written as follows:

$$\mathbf{u}_t + (\mathbf{u} \cdot \text{grad})\mathbf{u} - \nu \Delta \mathbf{u} + \text{grad}p + \lambda \text{div}(\text{grad}\phi \otimes \text{grad}\phi) = \mathbf{g}(\mathbf{x}), \tag{2.3}$$

$$\text{div} \mathbf{u} = 0, \tag{2.4}$$

where \mathbf{g} is the external body force, p is the pressure, ν is the kinematic viscosity, λ corresponds to the surface tension, and the term $\text{grad}\phi \otimes \text{grad}\phi$ is the usual tensor product, i.e., $(\text{grad}\phi \otimes \text{grad}\phi)_{ij} = \text{grad}_i\phi \text{grad}_j\phi$. Since $\text{div}(\text{grad}\phi \otimes \text{grad}\phi) = \Delta\phi \text{grad}\phi + \frac{1}{2} \text{grad}|\text{grad}\phi|^2$, the momentum Eq. (2.3) can be simplified as follows:

$$\mathbf{u}_t + (\mathbf{u} \cdot \text{grad})\mathbf{u} - \nu \Delta \mathbf{u} + \text{grad}p = -\lambda \Delta\phi \text{grad}\phi + \mathbf{g}(\mathbf{x}), \tag{2.5}$$

here p has been redefined as $p := p + \frac{1}{2} \lambda |\text{grad}\phi|^2$.

The coupled system (2.1), (2.3) or (2.5), and (2.4) is supplemented with the initial conditions

$$\mathbf{u}|_{t=0} = \mathbf{u}_0(\mathbf{x}), \quad \phi|_{t=0} = \phi_0(\mathbf{x}), \quad \mathbf{x} \in \Omega_p, \tag{2.6}$$

and boundary conditions

$$\mathbf{u}|_{\partial\Omega_p} = 0, \quad \frac{\partial\phi}{\partial n} \Big|_{\partial\Omega_p} = 0, \tag{2.7}$$

where $\frac{\partial\phi}{\partial n} = \mathbf{n} \cdot \text{grad}\phi$, and \mathbf{n} is the outward unit normal vector on the edge of the domain Ω_p . From (2.1) and the above conditions, one derives easily

$$\zeta(t) = \frac{1}{|\Omega_p|} \int_{\Omega_p} f(\phi(\mathbf{x}, t)) d\mathbf{x},$$

where $|\Omega_p|$ denotes the area of the solution domain Ω_p .

Remark 2.1. As $\hat{\eta} \rightarrow 0$, the Cahn–Hilliard phase model and Allen–Cahn type of phase equation coupling with the Navier–Stokes equations (2.3) and (2.4) will tend to the same limit, see [22] as well as [1].

3. A projection method on a fixed half-staggered mesh

Our adaptive phase field method consists of two independent parts: the evolution of the governing equation and the mesh redistribution. The first part is discussed in this section, while the second part will be introduced in Section 4.

In the following, the incompressible Navier–Stokes equations (2.3) and (2.4) will be solved by using the rotational incremental pressure-correction scheme of Timmer-

mans et al. [32] on a fixed half-staggered mesh, and a conservative scheme is given for Eq. (2.1). The main advantage of the rotational incremental pressure-correction scheme is that it can overcome the difficulty caused by the artificial pressure Neumann boundary condition.

Give a partition of the physical domain Ω_p , $\{A_{j+\frac{1}{2},k+\frac{1}{2}}|j, k \in \mathbb{Z}\}$, a uniform partition of the computational or logical domain Ω_c is given with unit step sizes, i.e., $\Delta\xi = \Delta\eta = 1$, and a partition of the time interval $[0, T]$, $\{t_n = t_{n-1} + \Delta t_n | \Delta t_n > 0, n \in \mathbb{N}\}$, where $A_{j+\frac{1}{2},k+\frac{1}{2}}$ is a quadrangle with four corners $\mathbf{x}_{j,k}$, $\mathbf{x}_{j+1,k}$, $\mathbf{x}_{j+1,k+1}$, and $\mathbf{x}_{j,k+1}$. For convenience, define $I_0 := \{\mathbf{x}_{j+\frac{1}{2},k+\frac{1}{2}} \in \Omega_p, j, k \in \mathbb{Z}\}$ and $I_n := \{\mathbf{x}_{j,k} \in \Omega_p, j, k \in \mathbb{Z}\}$, where $\mathbf{x}_{j+\frac{1}{2},k+\frac{1}{2}}$ denotes the centroid coordinates of the quadrangle $A_{j+\frac{1}{2},k+\frac{1}{2}}$. Let the scalar variables, e.g., p and ϕ , be approximated at cell centers by their cell averages

$$U_{j+\frac{1}{2},k+\frac{1}{2}}(t) = \frac{1}{|A_{j+\frac{1}{2},k+\frac{1}{2}}|} \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(\mathbf{x}, t) d\mathbf{x}, \quad U = p \text{ or } \phi, \tag{3.1}$$

and the vectors such as the velocity \mathbf{u} be approximated at the cell corners, see Fig. 3.1, where $|A_{j+\frac{1}{2},k+\frac{1}{2}}|$ denotes the area of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$. Moreover, the notations grad_h , div_h , and Δ_h are used to denote the discretizations of the gradient, divergence, and Laplacian operators, respectively, but will be given later, where $\Delta_h = \text{div}_h \text{grad}_h$.

The discrete projection method for Eqs. (2.1)–(2.3) is described as follows:

Algorithm 1.

Step 0: Give initial data $\phi^n, \mathbf{u}^n, p^n$.

Step 1 (the fluid evolution step): Compute the intermediate velocity field $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v})^T$ on I_n by a semi-implicit scheme:

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t_n} - \nu \Delta_h \tilde{\mathbf{u}} = -(\mathbf{u}^n \cdot \text{grad}_h) \mathbf{u}^n - \text{grad}_h p^n - (\lambda \Delta_h \phi^n) \text{grad}_h \phi^n + \mathbf{g}(\mathbf{x}), \tag{3.2}$$

with homogeneous Dirichlet boundary conditions

$$\tilde{\mathbf{u}} = 0, \quad \text{on } \partial\Omega_p.$$

Step 2: (the projection step) Project the intermediate velocity field $\tilde{\mathbf{u}}$ onto the divergence-free vector space by the standard Helmholtz decomposition

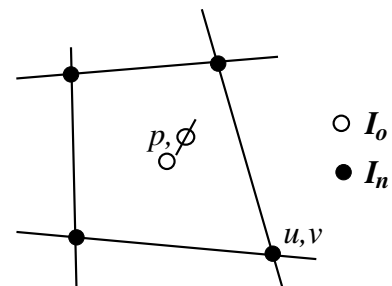


Fig. 3.1. Half-staggered mesh, see [12,13].

$$\begin{cases} \tilde{\mathbf{u}} = \mathbf{u}^{n+1} + \Delta t_n \text{grad}_h \psi^{n+1} & \text{on } I_n, \\ \text{div}_h \mathbf{u}^{n+1} = 0 & \text{on } I_0, \\ \mathbf{u}^{n+1} \cdot \mathbf{n} = 0 & \text{on } \partial\Omega_p, \end{cases} \quad (3.3)$$

and update the pressure p^{n+1} by

$$\psi^{n+1} = p^{n+1} - p^n + \nu \text{div}_h \tilde{\mathbf{u}} \quad \text{on } I_0. \quad (3.4)$$

The projection step is equivalent to solving a pressure Poisson equation together with the homogeneous Neumann boundary condition

$$\begin{cases} \Delta_h \psi = \frac{1}{\Delta t_n} \text{div}_h \tilde{\mathbf{u}}, & \text{on } I_0, \\ \text{grad}_h \psi \cdot \mathbf{n} = 0. \end{cases} \quad (3.5)$$

Step 3 (the phase evolution step): Compute the phase field ϕ^{n+1} by

$$\frac{\phi^{n+1} - \phi^n}{\Delta t_n} - \gamma \Delta_h \phi^{n+1} = -\text{div}_h(\mathbf{u}^{n+1} \phi^n) - \gamma f(\phi^n) + \gamma \zeta(t^n) \quad \text{on } I_0. \quad (3.6)$$

The detailed definitions of the discrete operators grad_h , div_h , and Δ_h used in the above projection method are presented as follows. The gradient, divergence, and Laplacian operators are first transferred into the logical domain Ω_c by the coordinate transformation $\mathbf{x} = \mathbf{x}(\xi)$ as follows:

$$\begin{aligned} \text{grad} \phi &= \frac{1}{J} (y_\eta \phi_\xi - y_\xi \phi_\eta, -x_\eta \phi_\xi + x_\xi \phi_\eta)^T \\ &\quad \text{(non-conservative)} \\ &= \frac{1}{J} ((y_\eta \phi)_\xi - (y_\xi \phi)_\eta, -(x_\eta \phi)_\xi + (x_\xi \phi)_\eta)^T, \\ &\quad \text{(conservative)} \end{aligned} \quad (3.7)$$

$$\begin{aligned} \text{div} \mathbf{u} &= \frac{1}{J} (y_\eta u_\xi - y_\xi u_\eta - x_\eta v_\xi + x_\xi v_\eta) \\ &\quad \text{(non-conservative)} \\ &= \frac{1}{J} ((y_\eta u)_\xi - (y_\xi u)_\eta - (x_\eta v)_\xi + (x_\xi v)_\eta), \\ &\quad \text{(conservative)} \end{aligned} \quad (3.8)$$

$$\begin{aligned} \Delta \psi &= \frac{1}{J} ((J^{-1} y_\eta^2 \psi_\xi)_\xi - (J^{-1} y_\xi y_\eta \psi_\eta)_\xi \\ &\quad - (J^{-1} y_\xi y_\eta \psi_\xi)_\eta + (J^{-1} y_\xi^2 \psi_\eta)_\eta \\ &\quad + (J^{-1} x_\eta^2 \psi_\xi)_\xi - (J^{-1} x_\xi x_\eta \psi_\eta)_\xi \\ &\quad - (J^{-1} x_\xi x_\eta \psi_\xi)_\eta + (J^{-1} x_\xi^2 \psi_\eta)_\eta), \end{aligned} \quad (3.9)$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian matrix of the coordinate transformation. Then all partial derivatives involved in (3.7) and (3.8) are approximated by using second-order accurate central finite difference scheme. For example, the non-conservative gradient of a scalar variable is approximated on I_n , see Fig. 3.2, by

$$\begin{aligned} \text{grad}_h \psi &= \frac{1}{J} \left((\widehat{y_\eta}) (\widehat{\psi_\xi}) - (\widehat{y_\xi}) (\widehat{\psi_\eta}), \right. \\ &\quad \left. - (\widehat{x_\eta}) (\widehat{\psi_\xi}) + (\widehat{x_\xi}) (\widehat{\psi_\eta}) \right)^T, \quad \text{on } I_n, \end{aligned} \quad (3.10)$$

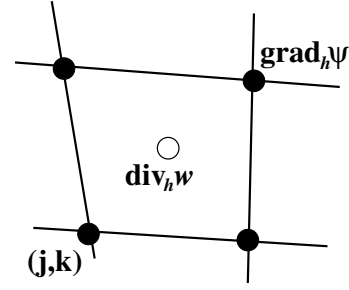


Fig. 3.2. Half-staggered mesh for div_h and grad_h .

where

$$\begin{aligned} (\widehat{\psi_\xi})_{j,k} &:= (\psi_{j+\frac{1}{2},k+\frac{1}{2}} - \psi_{j-\frac{1}{2},k+\frac{1}{2}} + \psi_{j+\frac{1}{2},k-\frac{1}{2}} - \psi_{j-\frac{1}{2},k-\frac{1}{2}})/2, \\ (\widehat{\psi_\eta})_{j,k} &:= (\psi_{j+\frac{1}{2},k+\frac{1}{2}} - \psi_{j+\frac{1}{2},k-\frac{1}{2}} + \psi_{j-\frac{1}{2},k+\frac{1}{2}} - \psi_{j-\frac{1}{2},k-\frac{1}{2}})/2, \\ (\widehat{Z_\xi})_{j,k} &:= (Z_{j+1,k} - Z_{j-1,k})/2, \quad (\widehat{Z_\eta})_{j,k} := (Z_{j,k+1} - Z_{j,k-1})/2, \\ \widehat{J}_{j,k} &= (\widehat{x_\xi} \widehat{y_\eta} - \widehat{x_\eta} \widehat{y_\xi})_{j,k}, \quad Z = x \text{ or } y. \end{aligned}$$

Similarly, the conservative divergence of the velocity vector is approximated on I_0 , see Fig. 3.2, by

$$\text{div}_h \mathbf{u} = \frac{1}{J} \left(\overline{(y_\eta u - x_\eta v)_\xi} + \overline{(-y_\xi u + x_\xi v)_\eta} \right) \quad \text{on } I_0, \quad (3.11)$$

where

$$\begin{aligned} \overline{(y_\eta u - x_\eta v)_\xi}_{j+\frac{1}{2},k+\frac{1}{2}} &:= \frac{1}{2} ((\widehat{y_\eta} u - \widehat{x_\eta} v)_{j+1,k} - (\widehat{y_\eta} u - \widehat{x_\eta} v)_{j-1,k}), \\ \overline{(-y_\xi u + x_\xi v)_\eta}_{j+\frac{1}{2},k+\frac{1}{2}} &:= \frac{1}{2} ((-\widehat{y_\xi} u + \widehat{x_\xi} v)_{j,k+1} - (-\widehat{y_\xi} u + \widehat{x_\xi} v)_{j,k-1}), \\ \widehat{J}_{j+\frac{1}{2},k+\frac{1}{2}} &= |A_{j+\frac{1}{2},k+\frac{1}{2}}|. \end{aligned}$$

From (3.10) and (3.11), a discrete Laplacian operator $\Delta_h = \text{div}_h \text{grad}_h$ is gotten. For the Poisson problems in Algorithm 1, such defined Laplacian operator $\Delta_h = \text{div}_h \text{grad}_h$ will give a mimetic discretization [27] and a linear algebraic system with a symmetric and semi-definite coefficient matrix. Many methods can be used to solve this linear system, for example, the algebraic multigrid (AMG), the conjugate gradient (CG) method, and the multilevel dissection method, see [13,12]. In this study, the AMG package [18] or the CG method have been used to solve this linear system.

If the mesh is uniformly rectangular, then the approximations (3.10) and (3.11) lead to a skewed five point scheme of the Poisson equation, see Fig. 3.3(a). The solution vector $\psi = (\dots, \psi_{j+\frac{1}{2},k+\frac{1}{2}}, \dots)^T$ of the Poisson equation has two degrees of freedom, in other words, $\psi + \psi_1 + c\psi_2$ is also its solution, where c is an arbitrary constant, ψ_1 is a constant vector, see Fig. 3.3(b), and ψ_2 is a so-called ‘‘checkerboard mode’’ vector, see Fig. 3.3(c). However, neither ψ_1 nor ψ_2 will affect $\text{grad}_h \psi$, see (3.10).

Finally, a few remarks on the above algorithm are given.

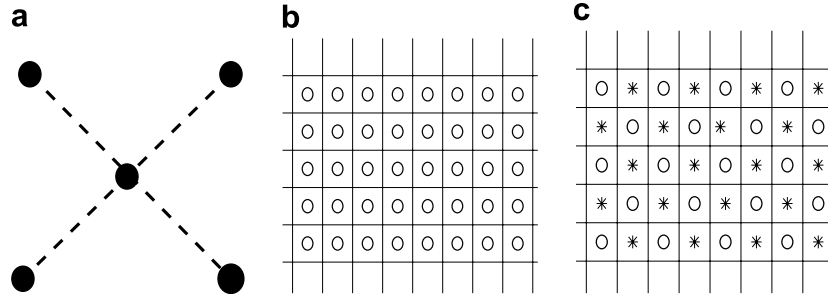


Fig. 3.3. Stencil of the skewed scheme and profile of the vector ψ_i , $i = 1, 2$. Symbols “○” and “*” denote two arbitrary constants. (a) Skewed scheme; (b) ψ_1 ; (c) ψ_2 .

Remark 3.1. In this work, $\text{grad}\psi$ and $\text{div}\tilde{u}$ in *step 2* of Algorithm 1 are discretized by using the formulae (3.10) and (3.11), respectively, so that the approximate velocity is exactly divergence free in the discrete sense. Such discrete approach will also be used to project the velocity vector onto a divergence-free space at the end of the iterative mesh redistribution, see Section 4.3.

Remark 3.2. The gradient operator in the momentum Eq. (3.2) may be either non-conservative or conservative. Moreover, the Laplacian operators in (3.2) and (3.6) may also be a second-order accurate finite difference approximation of (3.9). Such resulting linear algebra system of the Poisson equation will be symmetric and definite, and may be solved by using the multigrid method [18,38]. The tolerance for the l^2 norm of the residual is set to be 10^{-8} and the average iterative number 3–5.

Remark 3.3. To preserve the mass-conservation of the phase field ϕ , i.e.,

$$\sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1}| \phi_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^n| \phi_{j+\frac{1}{2},k+\frac{1}{2}}^n,$$

the discrete operators in (3.6) should be conservative, and corresponding Lagrangian multiplier becomes

$$\zeta_{j+\frac{1}{2},k+\frac{1}{2}}^n = \frac{1}{|\Omega_p^n|} \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^n| f(\phi_{j+\frac{1}{2},k+\frac{1}{2}}^n),$$

where $|A_{j+\frac{1}{2},k+\frac{1}{2}}^n|$ is the area of the control cell $A_{j+\frac{1}{2},k+\frac{1}{2}}$, and $|\Omega_p^n| = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^n|$.

4. Adaptive mesh redistribution

This section introduces our adaptive mesh redistribution briefly. The readers are referred to [30] for detailed descriptions.

4.1. Iterative mesh redistribution

Let $\mathbf{x} = (x, y)$ and $\boldsymbol{\xi} = (\xi, \eta)$ denote the physical and logical or computational coordinates, respectively. A one-to-one coordinate transformation from the logical or com-

putational domain Ω_c to the physical domain Ω_p is denoted by

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \Omega_c. \quad (4.1)$$

Our attention is limited to the case of that the physical domain Ω_p is convex and the map (4.1) is to find the minimizer of the following functional [5,30,31]:

$$\tilde{E}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^2 \int_{\Omega_c} (\tilde{\nabla} \mathbf{x}_i)^T G_i \tilde{\nabla} \mathbf{x}_i d\xi, \quad (4.2)$$

where $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^T$, and G_i ($i = 1, 2$) are given symmetric positive definite matrices called monitor functions. In general, the monitor functions depend on the solution or its derivatives of the underlying governing equations. The simplest choice of the monitor functions is $G_i = \omega I$, $i = 1, 2$, see [34], where I denotes the identity matrix and ω is a positive weight function. More terms can be added to the above functional to control other aspects of the mesh such as orthogonality and alignment with a given vector field, see e.g., [3,15].

Using Winslow’s choice, one deduces the Euler–Lagrange equations of the functional (4.2) to

$$\tilde{\nabla} \cdot (\omega \tilde{\nabla} \mathbf{x}) = 0. \quad (4.3)$$

In this study, $\omega = \omega(\text{grad}\phi)$. Eq. (4.3) is first discretized by the central difference scheme and then solved by the Gauss–Seidel (GS) iteration, that is to say,

$$\begin{aligned} &\omega_{j+\frac{1}{2},k}^{[m]} (\mathbf{x}_{j+1,k}^{[m]} - \mathbf{x}_{j,k}^{[m+1]}) - \omega_{j-\frac{1}{2},k}^{[m]} (\mathbf{x}_{j,k}^{[m+1]} - \mathbf{x}_{j-1,k}^{[m+1]}) \\ &+ \omega_{j,k+\frac{1}{2}}^{[m]} (\mathbf{x}_{j,k+1}^{[m]} - \mathbf{x}_{j,k}^{[m+1]}) - \omega_{j,k-\frac{1}{2}}^{[m]} (\mathbf{x}_{j,k}^{[m+1]} - \mathbf{x}_{j,k-1}^{[m+1]}) = 0, \end{aligned} \quad (4.4)$$

for $m = 0, 1, \dots$, where $\omega_{j+\frac{1}{2},k} = \frac{1}{2}(\omega_{j+1,k} + \omega_{j,k})$, $\omega_{j,k+\frac{1}{2}} = \frac{1}{2}(\omega_{j,k+1} + \omega_{j,k})$. The GS iteration is continued until $\|\mathbf{x}^{[m]} - \mathbf{x}^{[m+1]}\| < \epsilon$ or $m < \mu$, where μ is a given small integer. In practice, a few iterations (say $\mu = 3-5$) are required at each time level, so the cost for generating new mesh is not too expensive. In order to obtain a smooth mesh distribution, the low-pass filter

$$\begin{aligned} \omega_{j,k} &\leftarrow \frac{4}{16} \omega_{j,k} + \frac{2}{16} (\omega_{j+1,k} + \omega_{j-1,k} + \omega_{j,k+1} + \omega_{j,k-1}) \\ &+ \frac{1}{16} (\omega_{j-1,k-1} + \omega_{j-1,k+1} + \omega_{j+1,k-1} + \omega_{j+1,k+1}) \end{aligned}$$

is applied to the discrete monitor function, which is usually carried out 3–5 times at each GS iterative step.

4.2. Interpolation of the cell-averaged variables

After each GS iterative step, the approximate solutions need to be remapped onto the newly resulted mesh $\{\mathbf{x}_{j,k}^{[m+1]}\}$ from the old mesh $\{\mathbf{x}_{j,k}^{[m]}\}$. The remapping procedure of the cell-averaged variables such as the phase field ϕ and p can be conducted by using the conservative interpolation technique proposed by Tang and Tang [30], which is

$$\begin{aligned} \left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]}\right| U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]} &= \left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}\right| U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]} \\ &\quad - \left((c_n^2 U^{[m]})_{j+1,k+\frac{1}{2}} + (c_n^4 U^{[m]})_{j,k+\frac{1}{2}} \right) \\ &\quad - \left((c_n^3 U^{[m]})_{j+\frac{1}{2},k+1} + (c_n^1 U^{[m]})_{j+\frac{1}{2},k} \right), \end{aligned} \tag{4.5}$$

where $U = \phi$ or p , $|A_{j+\frac{1}{2},k+\frac{1}{2}}|$ means area of the corresponding control cell, and $c_n^l := c^x n_x^l + c^y n_y^l$ with the mesh velocity $(c^x, c^y) = (x^{[m]} - x^{[m+1]}, y^{[m]} - y^{[m+1]})$ and the normal outward vector $\mathbf{n}^l = (n_x^l, n_y^l)$, defined as follows:

$$\begin{aligned} c_n^1 &= \frac{1}{2} (c_{j,k}^x + c_{j+1,k}^x) (y_{j+1,k} - y_{j,k}) \\ &\quad - \frac{1}{2} (c_{j,k}^y + c_{j+1,k}^y) (x_{j+1,k} - x_{j,k}), \\ c_n^2 &= \frac{1}{2} (c_{j+1,k}^x + c_{j+1,k+1}^x) (y_{j+1,k+1} - y_{j+1,k}) \\ &\quad - \frac{1}{2} (c_{j+1,k}^y + c_{j+1,k+1}^y) (x_{j+1,k+1} - x_{j+1,k}), \\ c_n^3 &= \frac{1}{2} (c_{j+1,k+1}^x + c_{j,k+1}^x) (y_{j,k+1} - y_{j+1,k+1}) \\ &\quad - \frac{1}{2} (c_{j+1,k+1}^y + c_{j,k+1}^y) (x_{j,k+1} - x_{j+1,k+1}), \\ c_n^4 &= \frac{1}{2} (c_{j,k+1}^x + c_{j,k}^x) (y_{j,k} - y_{j,k+1}) \\ &\quad - \frac{1}{2} (c_{j,k+1}^y + c_{j,k}^y) (x_{j,k} - x_{j,k+1}), \end{aligned}$$

and the fluxes $(c_n^l U)_{j+r,k+\frac{1}{2}}$ and $(c_n^l U)_{j+\frac{1}{2},k+s}$, $r, s = 0$ or 1 , denote the values of $c_n^l U$ on the corresponding surface of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$, where $l = 1, 2, 3, 4$. The fluxes will be approximated by using an upwind scheme. For example, the term $(c_n^2 U)_{j+1,k+\frac{1}{2}}$ may be approximated by

$$\begin{aligned} (c_n^2 U)_{j+1,k+\frac{1}{2}} &= \frac{c_n^2}{2} \left(U_{j+\frac{3}{2},k+\frac{1}{2}} + U_{j+\frac{1}{2},k+\frac{1}{2}} \right) \\ &\quad - \frac{|c_n^2|}{2} \left(U_{j+\frac{3}{2},k+\frac{1}{2}} - U_{j+\frac{1}{2},k+\frac{1}{2}} \right). \end{aligned} \tag{4.6}$$

The above approximation is only first order accurate in space. In order to avoid large numerical dissipation, the reconstruction technique will be used, see [30], and (4.6) is replaced by

$$\begin{aligned} (c_n^2 U)_{j+1,k+\frac{1}{2}} &= \frac{c_n^2}{2} \left(U_{j+1+0,k+\frac{1}{2}} + U_{j+1-0,k+\frac{1}{2}} \right) \\ &\quad - \frac{|c_n^2|}{2} \left(U_{j+1+0,k+\frac{1}{2}} - U_{j+1-0,k+\frac{1}{2}} \right), \end{aligned} \tag{4.7}$$

with

$$\begin{aligned} U_{j+1+0,k+\frac{1}{2}} &= U_{j+\frac{3}{2},k+\frac{1}{2}} - \frac{1}{2} S_{j+\frac{3}{2},k+\frac{1}{2}}^\xi, \\ U_{j+1-0,k+\frac{1}{2}} &= U_{j+\frac{1}{2},k+\frac{1}{2}} + \frac{1}{2} S_{j+\frac{1}{2},k+\frac{1}{2}}^\xi. \end{aligned}$$

Here S^ξ is an approximation of $\partial U / \partial \xi$, taken by us as

$$S_{j+\frac{1}{2},k+\frac{1}{2}}^\xi = \text{vLL}(\Delta_\xi U_{j+\frac{1}{2},k}, \Delta_\xi U_{j-\frac{1}{2},k}),$$

where $\Delta_\xi U_{j-\frac{1}{2},k+\frac{1}{2}} = U_{j+\frac{1}{2},k+\frac{1}{2}} - U_{j-\frac{1}{2},k+\frac{1}{2}}$. The function $\text{vLL}(a, b)$ denotes van Leer’s limiter defined by

$$\text{vLL}(a, b) = (\text{sign}(a) + \text{sign}(b)) \frac{|ab|}{|a| + |b| + \varepsilon}, \tag{4.8}$$

where the parameter ε , $0 < \varepsilon \ll 1$, is used to avoid that the denominator becomes zero.

The formula (4.5) is obtained by using the classical perturbation theory, and satisfies the following mass-conservation property:

$$\sum_{j,k} \left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]}\right| U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]} = \sum_{j,k} \left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}\right| U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}, \quad U = \phi \text{ or } p. \tag{4.9}$$

Some further theoretical properties of this conservative interpolation can be found in [30].

4.3. Interpolation of the velocity variables

This subsection begins to remap the velocity variables onto the newly resulted mesh $\{\mathbf{x}_{j,k}^{[m+1]}\}$ from the old mesh $\{\mathbf{x}_{j,k}^{[m]}\}$. For the velocity vector $\mathbf{u} = (u, v)$, the high-resolution, nonconservative interpolation of Tang et al. [31] is employed, which is obtained by using Taylor’s expansion, i.e.,

$$\mathbf{u}(\mathbf{x}_{j,k}^{[m+1]}) \approx \mathbf{u}(\mathbf{x}_{j,k}^{[m]}) - (\mathbf{x}_{j,k}^{[m]} - \mathbf{x}_{j,k}^{[m+1]}) \cdot \text{grad} \mathbf{u}(\mathbf{x}_{j,k}^{[m]}). \tag{4.10}$$

Using the coordinate transformation $\xi = \xi(\mathbf{x})$ and a high-resolution Hamilton–Jacobi solver gives

$$\begin{aligned} \mathbf{u}_{j,k}^{[m+1]} &= \mathbf{u}_{j,k}^{[m]} - \frac{1}{2} \left(c_{j,k}^\xi \left(\mathbf{v}_{j+0,k}^{[m]} + \mathbf{v}_{j-0,k}^{[m]} \right) - |c_{j,k}^\xi| \left(\mathbf{v}_{j+0,k}^{[m]} - \mathbf{v}_{j-0,k}^{[m]} \right) \right) \\ &\quad - \frac{1}{2} \left(c_{j,k}^\eta \left(\mathbf{w}_{j,k+0}^{[m]} + \mathbf{w}_{j,k-0}^{[m]} \right) - |c_{j,k}^\eta| \left(\mathbf{w}_{j,k+0}^{[m]} - \mathbf{w}_{j,k-0}^{[m]} \right) \right), \end{aligned} \tag{4.11}$$

where

$$\begin{aligned} (c^\xi)_{j,k} &= \frac{1}{J_{j,k}} [x_\eta (y^{[m]} - y^{[m+1]}) - y_\eta (x^{[m]} - x^{[m+1]})]_{j,k}, \\ (c^\eta)_{j,k} &= \frac{1}{J_{j,k}} [y_\xi (x^{[m]} - x^{[m+1]}) - x_\xi (y^{[m]} - y^{[m+1]})]_{j,k}, \end{aligned}$$

and

$$\begin{aligned} \mathbf{v}_{j+0,k} &= \Delta_\xi \mathbf{u}_{j,k} - \frac{1}{2} \text{vLL}(\Delta_\xi \mathbf{u}_{j+1,k} - \Delta_\xi \mathbf{u}_{j,k}, \Delta_\xi \mathbf{u}_{j,k} - \Delta_\xi \mathbf{u}_{j-1,k}), \\ \mathbf{v}_{j-0,k} &= \Delta_\xi \mathbf{u}_{j-1,k} + \frac{1}{2} \text{vLL}(\Delta_\xi \mathbf{u}_{j,k} - \Delta_\xi \mathbf{u}_{j-1,k}, \Delta_\xi \mathbf{u}_{j-1,k} - \Delta_\xi \mathbf{u}_{j-2,k}), \\ \mathbf{w}_{j,k+0} &= \Delta_\eta \mathbf{u}_{j,k} - \frac{1}{2} \text{vLL}(\Delta_\eta \mathbf{u}_{j,k+1} - \Delta_\eta \mathbf{u}_{j,k}, \Delta_\eta \mathbf{u}_{j,k} - \Delta_\eta \mathbf{u}_{j,k-1}), \\ \mathbf{w}_{j,k-0} &= \Delta_\eta \mathbf{u}_{j,k-1} + \frac{1}{2} \text{vLL}(\Delta_\eta \mathbf{u}_{j,k} - \Delta_\eta \mathbf{u}_{j,k-1}, \Delta_\eta \mathbf{u}_{j,k-1} - \Delta_\eta \mathbf{u}_{j,k-2}), \end{aligned}$$

here $\Delta_\xi \mathbf{u}_{j,k} = \mathbf{u}_{j+1,k} - \mathbf{u}_{j,k}$, $\Delta_\eta \mathbf{u}_{j,k} = \mathbf{u}_{j,k+1} - \mathbf{u}_{j,k}$. The function $\text{vLL}(a, b)$ denotes van Leer’s limiter, see (4.8).

It is worth noting that the velocity $\mathbf{u}_{j,k}^{[m+1]}$ updated by (4.11) is not divergence-free generally. So $\mathbf{u}_{j,k}^{[m]}$ cannot be considered as the “initial” data of the incompressible Navier–Stokes equations, that is to say, one has to project $\mathbf{u}_{j,k}^{[m]}$ onto the divergence-free space before evolving the incompressible Navier–Stokes equations. Following the projection method for the incompressible Navier–Stokes equations, the standard Helmholtz decomposition is used:

$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}^{[m]} - \text{grad}_h \psi, \quad \text{div}_h \mathbf{u}^{n+1} = 0, \\ \mathbf{u}^{n+1} \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega_p. \end{cases} \quad (4.12)$$

to get the divergence-free velocity vector \mathbf{u}^{n+1} on the resultant adaptive mesh $\{\mathbf{x}_{j,k}^{[m]}\}$. It is equivalent to solving the Neumann boundary value problem of a Poisson equation:

$$\begin{cases} \Delta_h \psi = \text{div}_h \mathbf{u}^{[m]}, \\ \text{grad}_h \psi \cdot \mathbf{n} = 0, \quad \text{on } \partial\Omega_p. \end{cases} \quad (4.13)$$

Definitions of the above discrete operators are similar to those given in Section 3.

5. Solution procedure

Our solution procedure is composed by two independent parts: evolution of the governing equations and an iterative mesh redistribution. The first part is a divergence-free finite volume method, see Section 3. In each iteration of the second part, mesh points are first redistributed by the Gauss–Seidel method (4.4), and then ϕ and p are updated on the newly generated meshes by the conservative-interpolation formula (4.5), while the velocity vector \mathbf{u} is remapped by the non-conservative approach (4.11) and corrected finally by (4.12), see Section 4.3. The solution procedure can be illustrated by the following flowchart:

Algorithm 2.

Step 0: Give an initial adaptive mesh $\mathbf{x}_{j,k}^n$ based on the initial function, $n \geq 0$.

Step 1: Advance the solution one time step Δt_n by Algorithm 1 to get \mathbf{u}^{n+1} , p^{n+1} and ϕ^{n+1} .

Step 2: Set $\mathbf{x}_{j,k}^{[0]} := \mathbf{x}_{j,k}^n$, $\mathbf{u}_{j,k}^{[0]} := \mathbf{u}_{j,k}^{n+1}$, and $U_{j+\frac{1}{2},k+\frac{1}{2}}^{[0]} := U_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1}$, $U = \phi$ or p .

Step 3: For $m = 0, 1, 2, \dots, \mu - 1$, do the following:

(a) Solve the mesh redistributing equation (a generalized Laplacian equation) by one Gauss–Seidel iteration to get $\mathbf{x}^{[m+1]}$;

(b) Remap the approximate solutions on the new grid $\mathbf{x}^{[m+1]}$ to get $\mathbf{u}^{[m+1]}$, $p^{[m+1]}$ and $\phi^{[m+1]}$.

(c) Compute the monitor function $\omega^{[m+1]}$.

Step 4: Perform the projection method to get the divergence-free velocity field $\mathbf{u}_{j,k}^{n+1}$, and reset $\mathbf{x}_{j,k}^{n+1} := \mathbf{x}_{j,k}^{[m]}$ and $U_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} := U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}$, $U = \phi$ or p .

Step 5: If $t \geq T$, then save the result and stop. Otherwise, go to *Step 1* for the next time circle.

Remark 5.1. The conservative interpolation (4.5) and the non-conservative interpolation (4.11) are carried out after each GS iteration step, while the divergence free correction in (4.12) is only performed once after all μ GS iterations are finished. In our computations, the parameter μ is taken as 3–5 to get a satisfactory mesh redistribution.

Remark 5.2. Using Algorithm 2 can get a new set of mesh points and corresponding solutions, e.g., $\{\mathbf{x}^{n+1}, \phi^{n+1}, p^{n+1}, \mathbf{u}^{n+1}\}$, which satisfy

$$\sum_{j,k} \left| A_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} \right| \phi_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \sum_{j,k} \left| A_{j+\frac{1}{2},k+\frac{1}{2}}^n \right| \phi_{j+\frac{1}{2},k+\frac{1}{2}}^n, \quad (\text{div}_h \mathbf{u})_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = 0,$$

i.e., the phase variable is conservative and the velocity field is locally divergence free in the discrete sense.

6. Numerical experiments

In this section, the adaptive phase field algorithm, Algorithm 2, is applied to several two-dimensional problems on the mixture of two incompressible fluids. Throughout our computations, the physical parameters appeared in Section 2 are taken as

$$\hat{\eta} = 0.02, \quad \lambda = 0.1, \quad \nu = 0.1, \quad \gamma = 0.1.$$

The monitor function in (4.3) is chosen as

$$\omega = \sqrt{1 + \alpha |\text{grad} \phi|^2}, \quad (6.1)$$

where α is generally a problem-dependent positive parameter, but is taken as 5 in our computations. The initial velocity \mathbf{u} is assumed to be equal to zero in the physical domain $\Omega_p = [0, 2\pi] \times [0, 2\pi]$, i.e., the initial fluid is at rest. A 2D structured mesh of 64×64 cells is always used to adaptively capture the phase evolution with the time step size Δt of 0.002.

Example 6.1. The first example is to exhibit the surface tension effect. Initially, a square bubble is centered at (π, π) , i.e., the center of the domain Ω_p , with the side length of 2, and ϕ is taken as 1 inside the bubble and -1 otherwise, respectively. Due to the surface tension, the initial bubble will deform into a circular bubble quickly. Note that a

uniform mesh of 128×128 has been adopted to get a satisfactory resolution in [22].

Fig. 6.1 gives the contours of the phase field ϕ at $t = 0.1, 0.2, 0.3, 0.5$. The results show the detailed time evolution of the interface: due to the surface tension effect, the square bubble first deforms into a circular bubble, and then deforms into a diamond bubble, and finally deforms back into circular bubble. Those results may be comparable to those in [22]. It is worth noting that the above deformation will repeat alternatively until the bubble reaches the steady circular bubble. The graphs of the velocity field given in Fig. 6.1 show the motion of the interface instantaneously. Plots of the adaptive meshes show that more mesh points are adaptively clustered in the region of the interface of the two fluids.

If $\alpha \equiv 0$ in (6.1), then Algorithm 2 reduces to a uniform mesh scheme. One has to use at least 256×256 uniform mesh in order to get satisfactory resolution in this example, because the width of the interface is about 0.02. Fig. 6.2 gives a comparison of the phase ϕ along $x = \pi$ calculated by using the adaptive and uniform mesh methods, where the symbol “circle” and the solid line denote the adaptive solution with a resolution of 64×64 and the computed solution obtained on a 256×256 uniform mesh. Table 1 gives the recorded CPU times on the Founder PC (Pentium IV, 2.8 GHz) under the Linux environment. The results show that both methods produce almost same numerical results, but the adaptive method takes much less CPU time than the uniform mesh method. Thus, the present adaptive phase field approach displays great advantage in saving the

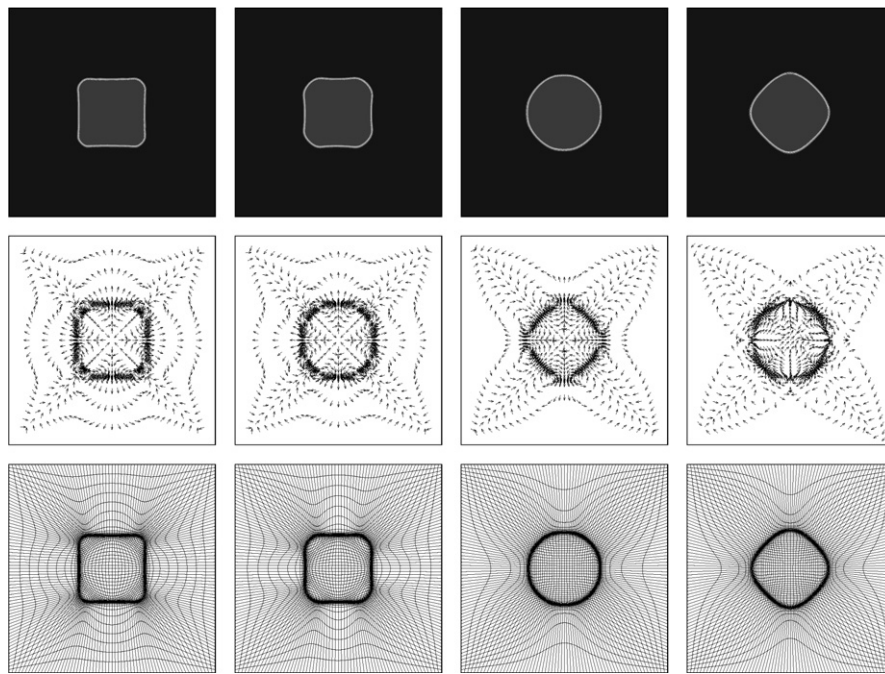


Fig. 6.1. Example 6.1: contours of the phase, the velocity fields, and mesh redistributions at $t = 0.1, 0.2, 0.3, 0.5$.

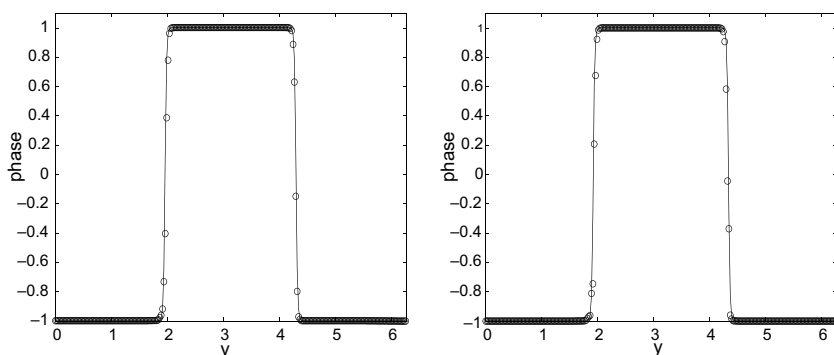


Fig. 6.2. Example 6.1: the phase ϕ along $x = \pi$. The symbol “circle” and solid line denote the adaptive solution with a resolution of 64×64 and the computed solution obtained on a 256×256 uniform mesh. Left: $t = 0.3$; right: $t = 0.5$.

Table 1

Example 6.1: a comparison of the CPU times (seconds) for the adaptive and fixed mesh methods

Algorithm (cell number)	$t = 0.3$	$t = 0.5$
Adaptive mesh method (64^2)	164.02	258.65
Fixed mesh method (256^2)	3716.29	5940.43

computer storage and the CPU time, comparing with corresponding uniform mesh method.

Example 6.2. The second example is to study the coalescence of two kissing bubbles. Initially, two unit circular bubbles are centered at $(\pi - 1, \pi)$ and $(\pi + 1, \pi)$, respectively, and thus are kissing at (π, π) .

Fig. 6.3 plots the phase field at several different times $t = 0, 0.1, 0.2, 0.3, 0.4,$ and $0.8,$ and show clearly the detailed phase evolution: as time evolves, two initially kissing bubbles first coalesce into a big and elliptic bubble, and then the elliptic bubble is transfigured and deforms into a circular bubble due to the surface tension effect. Our result is well comparable to that obtained by using the Fourier spectral method in [22] with 128×128 grid points. Fig. 6.4 plots the adaptive meshes at $t = 0.2, 0.5, 0.8.$ It is easy to see that most of the mesh points are clustered in the

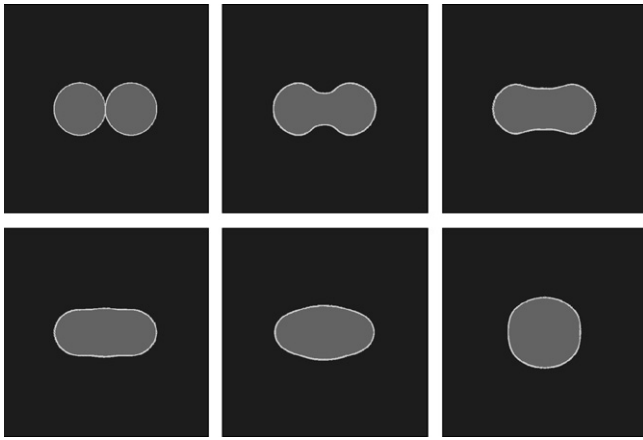


Fig. 6.3. Example 6.2: contours of the phase ϕ at $t = 0, 0.1, 0.2, 0.3, 0.4,$ and $0.8.$

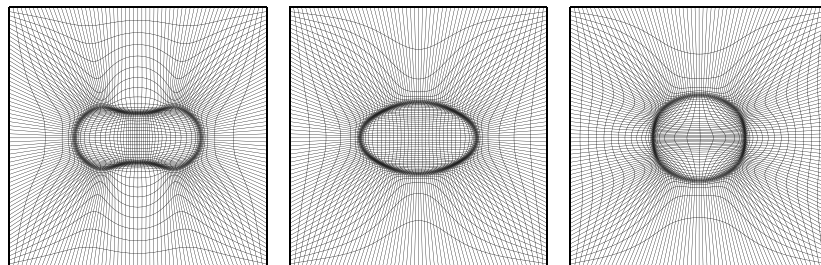


Fig. 6.4. Example 6.2: the adaptive mesh distributions at $t = 0.2, 0.5,$ and $0.8.$

region of the interface of the two fluids. It is worth noting that starting from $t = 0$ to $0.8,$ the thickness of the captured interface is always equal to $\hat{\eta}$ approximately.

Example 6.3. This example is to further exhibit coalescence of three equal bubbles which are initially kissing each other. Their radius are same and equal to $\pi/4,$ and ϕ is taken as 1 inside these bubbles and -1 otherwise.

Fig. 6.5 presents contours of the phase field ϕ and the adaptive mesh distributions at several different times. The interface dynamics are similar to those observed in the previous two examples: the three bubbles first coalesce into one big bubble due to the surface tension, and then that big bubble deforms into a inverted triangle bubble at $t = 0.2,$ and a big circle bubble at $t = 2.$ This deformation procedure continues alternately until the bubble reaches a steady circular bubble.

Fig. 6.6 gives the absolute value of the divergence of the velocity vector at $t = 0.2$ and $1.$ It shows that the divergence-free constraint (2.4) is well approximated with the numerical accuracy $10^{-8} - 10^{-7}$ around the interface of the two fluids.

Example 6.4. This example considers the gravity effect on the phase evolution besides the surface tension. In this case, the Navier–Stokes equations (2.3) has to be replaced by

$$\rho_0(\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u}) + \nabla p - \nu \Delta \mathbf{u} + \lambda \nabla \cdot (\nabla \phi \otimes \nabla \phi) = -\mathbf{g}\phi(\rho_1 - \rho_2),$$

where $\rho_1 - \rho_2$ and ρ_0 are taken as dimensionless constants -1 and $1,$ respectively. We start with two circular bubbles of different sizes at different heights, and assume that the density of the bubbles is smaller than that of the surrounding fluid. The gravitational constant is taken to be $\mathbf{g} = (0, 0.1)^T.$

Fig. 6.7 shows contours of the phase field ϕ and the adaptive mesh distributions at several different times. It is seen that the small bubble is absorbed by the big one before noticeable rising, and then the big bubble gradually rises to the top of the domain. The bubbles are also tinely deformed in y -direction due to the gravitational effect. Fig. 6.8 shows that the divergence of the velocity vector is equal to zero almost everywhere. In this computation, the volume

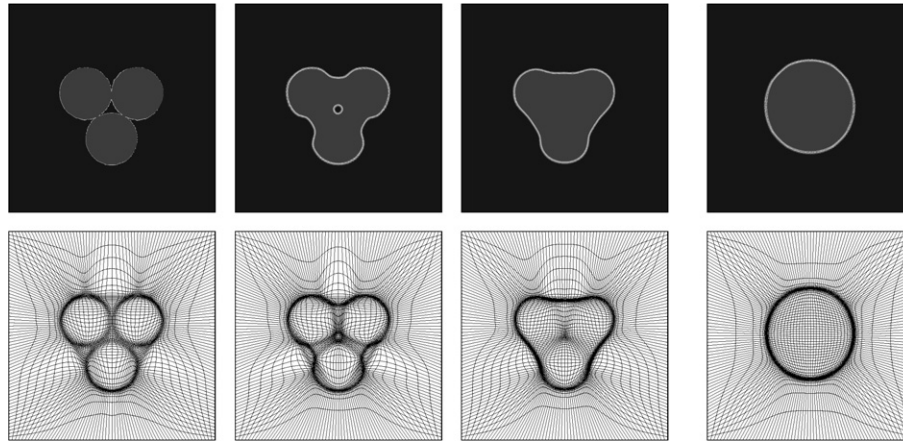


Fig. 6.5. Example 6.3: contours of the phase and the adaptive meshes at $t = 0, 0.1, 0.2, 2.0$.

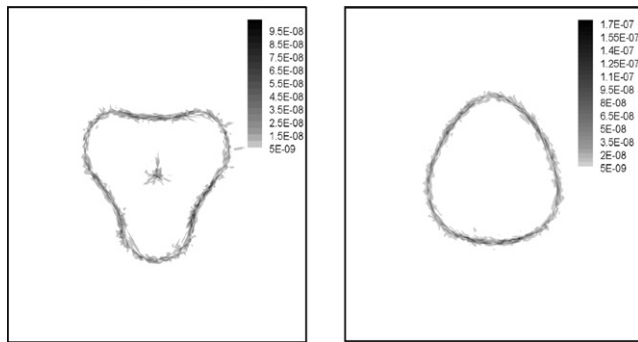


Fig. 6.6. Example 6.3: the absolute value of the divergence of velocity vector, $|\text{div}_v \mathbf{u}|$, at $t = 0.2$ (left) and 1 (right).

fraction of the two fluids is almost preserved all times because the phase is conservative approximately.

Example 6.5. The final example is to further consider the gravitational and surface tension effects. Initially, the two kissing bubbles are bigger and located higher than the third one. Take $\phi = 1$ inside three initial bubbles and $\phi = -1$ outside them. The gravitational constant and parameters ρ_i , $i = 0, 1, 2$, are taken as above.

Fig. 6.9 shows contours of the phase and the adaptive meshes at several different times. It is seen that two kissing, equal bubbles gradually coalesce into one big bubble and deform, while the small bubble is synchronously absorbed by the two upper coalescing bubbles until it vanishes. Before the three bubbles deform into one big bubble, the gravitation has no obvious effect on them. It is also found that the shape of the final bubble deforms alternatively before it reaches the steady state. This phenomenon is similar to that observed in the previous examples. More mesh points have been adaptively clustered in the region of the interface of the two fluids. Fig. 6.10 gives the discrete divergence of velocity field at $t = 0.5$ and 1.5 . It shows that around the interface region the numerical divergence of the velocity vector reaches the accuracy of 10^{-9} – 10^{-7} , while in the interior region of each fluid it is zero.

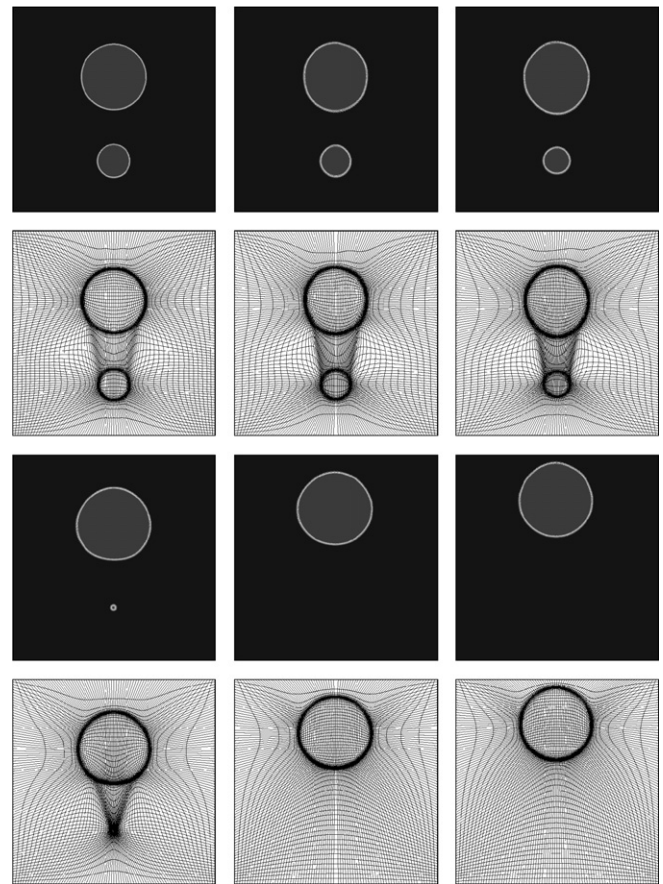


Fig. 6.7. Example 6.4: contours of the phase and the adaptive mesh distributions at $t = 0, 0.4, 0.8, 1.6, 4.5, 6.0$.

7. Concluding remarks

This paper has developed an efficient and fast adaptive moving mesh method to solve the Allen–Cahn type of phase field model for the mixture of two incompressible fluids. In the present algorithm the rotational incremental pressure-correction scheme was successfully implemented on a half-staggered, moving quadrilateral mesh to keep

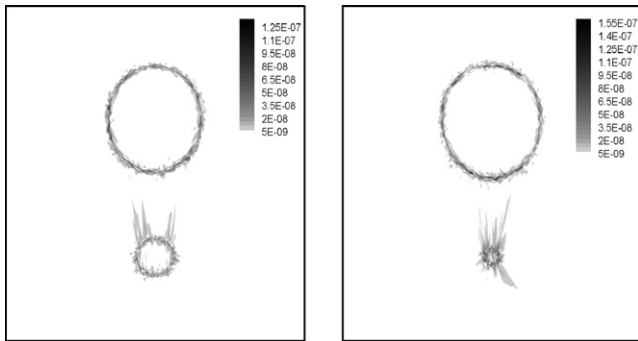


Fig. 6.8. Example 6.4: the absolute value of the velocity divergence, $|\text{div}_h \mathbf{u}|$, at $t = 0.2$ (left) and $t = 1.5$ (right).

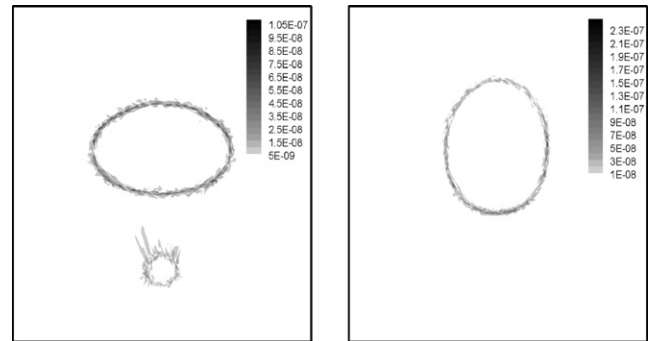


Fig. 6.10. Example 6.5: the absolute value of the velocity divergence $|\text{div}_h \mathbf{u}|$ at $t = 0.5$ (left) and $t = 1.5$ (right).

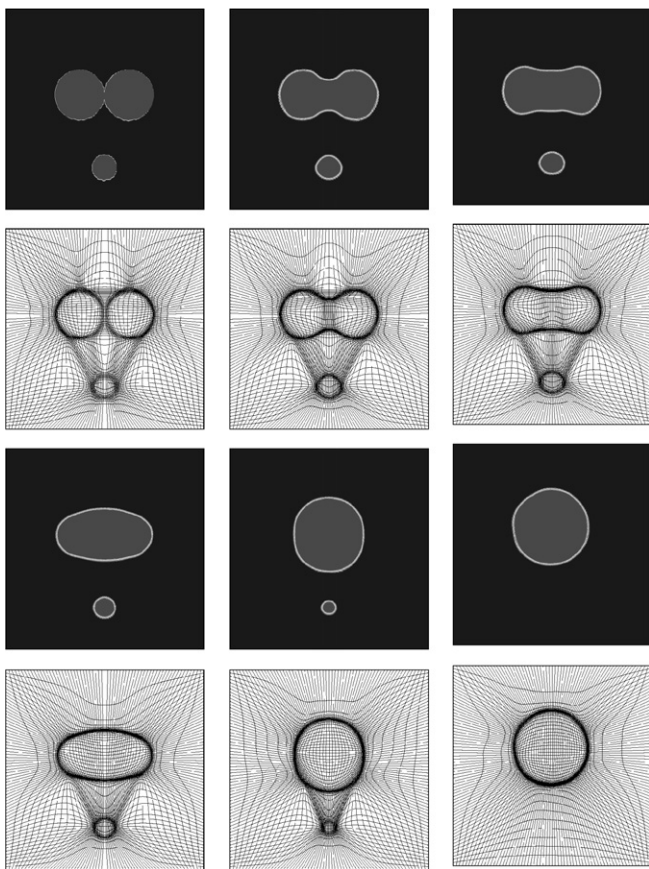


Fig. 6.9. Example 6.5: contours of the phase, and the mesh distributions at $t = 0, 0.1, 0.2, 0.4, 1.0, 2.0$.

the velocity field divergence-free, and the conjugate gradient or multigrid method was employed to fast solve the discrete Poisson equations.

The proposed algorithm consists of two independent parts: evolution of the governing equations and mesh-redistribution. In the first part, the rotational incremental pressure-correction scheme is used to solve the incompressible Navier–Stokes equations on a fixed half-staggered mesh, and the Allen–Cahn type of phase equation modified by the Lagrangian multiplier is approximated by a conservative, second-order accurate central difference scheme,

where the Lagrangian multiplier is used to preserve the overall mass of bubbles. The second part is an iteration procedure. The phase field is remapped onto the newly resulting meshes by the high-resolution conservative interpolation formula, while the non-conservative interpolation scheme is applied to the velocity field. The projection method is again used to obtain a divergence-free velocity at the end of the iterative mesh redistribution. Several numerical experiments have been conducted to demonstrate that the resultant numerical scheme is stable, mass conservative, highly efficient and fast, and capable of handling variable density and viscosity.

It is convenient to extend the present adaptive phase field method to three-dimensional phase field model for the mixture of two incompressible fluids.

Acknowledgements

The authors thank Professor Jie Shen and Professor Chun Liu for introducing this interesting research topic, as well as Professor Tao Tang for his helpful discussions during the preparation of this work. They would also like to thank the anonymous referees for the detailed reports that helped to improve the manuscript. Zhengru Zhang was partially supported by the National Natural Science Foundation of China (No. 10476032, 10601007). Huazhong Tang was partially supported by the National Basic Research Program under the Grant 2005CB321703, the National Natural Science Foundation of China (No. 10431050, 10576001), and Laboratory of Computational Physics.

References

- [1] Alikakos ND, Bates PW, Chen XF. Convergence of the Cahn–Hilliard equation to the Hele–Shaw model. *Arch Rational Mech Anal* 1994;128:165–265.
- [2] Beckett G, Mackenzie JA, Robertson ML. An r -adaptive finite element method for the solution of the two-dimensional phase-field equations. *Commun Comput Phys* 2006;1:805–26.
- [3] Brackbill JU. An adaptive grid with directional control. *J Comput Phys* 1993;108:38–50.
- [4] Cao WM, Huang WZ, Russell RD. An r -adaptive finite element method based upon moving mesh PDEs. *J Comput Phys* 1999;149:221–44.

- [5] Cenicerros HD, Hou TY. An efficient dynamically adaptive mesh for potentially singular solutions. *J Comput Phys* 2001;172:609–39.
- [6] Cliffe KA, Tavener SJ. Marangoni–Benard convection with a deformable free surface. *J Comput Phys* 1998;145:193–227.
- [7] Davis SF, Flaherty JE. An adaptive finite element method for initial-boundary value problems for partial differential equations. *SIAM J Sci Statist Comput* 1982;3:6–27.
- [8] Di YN, Li R, Tang T, Zhang PW. Moving mesh finite element methods for the incompressible Navier–Stokes equations. *SIAM J Sci Comput* 2005;26:1036–56.
- [9] Dvinsky AS. Adaptive grid generation from harmonic maps on Riemannian manifolds. *J Comput Phys* 1991;95:450–76.
- [10] E WN, Liu JG. Essentially compact schemes for unsteady viscous incompressible flows. *J Comput Phys* 1996;126:122–38.
- [11] E WN, Liu JG. Gauge method for viscous incompressible flows. *Commun Math Sci* 2003;1:317–32.
- [12] Geoge A, Huang LC, Tang WP, Wu YD. Numerical simulation of unsteady incompressible flow ($Re \leq 9500$) on the curvilinear half-staggered mesh. *SIAM J Sci Comput* 2000;21:2331–51.
- [13] Golub GH, Huang LC, Simon H, Tang WP. A fast Poisson solver for the finite difference solution of the incompressible Navier–Stokes equations. *SIAM J Sci Comput* 1998;19:1606–24.
- [14] Goodrich JW, Soh WY. Time-dependent viscous incompressible Navier–Stokes equations: The finite difference Galerkin formulation and streamfunction algorithms. *J Comput Phys* 1989;84:207–41.
- [15] Huang WZ. Metric tensors for anisotropic mesh generation. *J Comput Phys* 2005;204:633–65.
- [16] Huang WZ. Mathematical principles of anisotropic mesh adaptation. *Commun Comput Phys* 2006;1:276–310.
- [17] Karma A, Rappel WJ. Quantitative phase field modeling of dendritic growth in two and three dimensions. *Phys Rev E* 1998;57:4323–49.
- [18] Li R, Liu WB. The AFEPack Handbook. Available from: <http://circus.math.pku.edu.cn/AFEPack>.
- [19] Li R, Tang T, Zhang PW. Moving mesh methods in multiple dimensions based on harmonic maps. *J Comput Phys* 2001;170:562–88.
- [20] Li R, Tang T, Zhang PW. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *J Comput Phys* 2002;177:365–93.
- [21] Lin P, Liu C. Simulations of singularity dynamics in liquid crystal flows: A C^0 finite element approach. *J Comput Phys* 2006;215:348–62.
- [22] Liu C, Shen J. A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method. *Physica D* 2003;179:211–28.
- [23] Miller K, Miller RN. Moving finite element. I. *SIAM J Numer Anal* 1981;18:1019–32.
- [24] Nohetto RH, Pyo JH. The Gauge–Uzawa finite element method Part I: The Navier–Stokes equations. Preprint, University of Maryland, 2006. Available from: <http://www.math.umd.edu/~rhn>.
- [25] Ren WQ, Wang XP. An iterative grid redistribution method for singular problems in multiple dimensions. *J Comput Phys* 2000;159:246–73.
- [26] Rey AD. Viscoelastic theory for nematic interfaces. *Phys Rev E* 2000;61:1540–9.
- [27] Shashkov M, Steinberg S. Conservative finite-difference methods on general grids. CRC Press; 1996.
- [28] Tan ZJ, Tang T, Zhang ZR. A simple moving mesh method for one- and two-dimensional phase field equations. *J Comput Appl Math* 2006;190:252–69.
- [29] Tan ZJ, Zhang ZR, Tang T, Huang YQ. Moving mesh methods with locally varying time steps. *J Comput Phys* 2004;200:347–67.
- [30] Tang HZ, Tang T. Moving mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J Numer Anal* 2001;41:487–515.
- [31] Tang HZ, Tang T, Zhang PW. An adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimensions. *J Comput Phys* 2003;188:543–72.
- [32] Timmermans LJP, Mineev PD, Van De Vosse FN. An approximate projection scheme for incompressible flow using spectral elements. *Int J Numer Methods Fluids* 1996;22:673–88.
- [33] Trebotic DP, Colella P. A projection method for incompressible viscous flow on moving quadrilateral grids. *J Comput Phys* 2001;166:191–217.
- [34] Winslow A. Numerical solution of the quasi-linear Poisson equation. *J Comput Phys* 1967;1:149–72.
- [35] Xing Y, Shu CW. High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *J Comput Phys* 2006;214:567–98.
- [36] Xing Y, Shu CW. A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *Commun Comput Phys* 2006;1:100–34.
- [37] Yang XF, Feng JJ, Liu C, Shen J. Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method. *J Comput Phys* 2006;218:417–28.
- [38] De Zeeuw PM. Matrix-dependent prolongation and restrictions in a blackbox multigrid solver. *J Comput Phys* 1990;33:1–27.
- [39] Zhang ZR, Tang T. An adaptive mesh redistribution algorithm for convection-dominated problems. *Commun Pure Appl Anal* 2002;1:341–57.
- [40] Zhang ZR. A moving mesh method using a conservative interpolation based on L^2 projection. *Commun Comput Phys* 2006;1:930–44.