

# An Adaptive Phase Field Method for the Mixture of Two Incompressible Fluids\*

Huazhong Tang

School of Mathematical Sciences

Peking University, Beijing 100871, China

joint work with **Zhengru Zhang** (Beijing Normal University)

---

\*Workshop on “Rheology of Complex Fluids”, Jan 5-9, 2009, Ecole des Ponts, Marne-la-Vallée

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Governing Equations</b>	<b>8</b>
<b>3</b>	<b>Numerical Method</b>	<b>14</b>
3.1	INSE solver . . . . .	15
3.2	Adaptive mesh redistribution . . . . .	27
3.3	Solution Procedure . . . . .	39
<b>4</b>	<b>Numerical Results</b>	<b>43</b>
<b>5</b>	<b>Conclusion</b>	<b>53</b>

# 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.
- The **phase field approach**, a mathematical technique, based on thermodynamics, for describing the process of phase transition in a material (e.g. from a liquid to solid or another liquid), in-

roduces 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.

- This has been studied as early as 19th century by Rayleigh and van der Waals (see the wonderful survey paper by Anderson et al.[Ann. Rev. Fluid Mech. 1998] in this area).
- Such an approach coincides with the usual phase field models that were developed in the theory of phase transition, and attracted many interests in the mathematical community.
- These models allow topological changes of the interface and

have many advantages in numerical simulations of the interfacial motion.

- At the cost of solving an additional PDE, **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 and the Fourier-spectral method etc. **The adaptive mesh methods** have also been proved very effective for the phase field model. However, in 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 [JCP1967], Brackbill [JCP1993], Ren and Wang [JCP2000]; **moving finite element methods** of Millers [SINUM and Davis and Flaherty [SISC1982]; **moving mesh PDEs methods** of Russell et al. [JCP1999; CiCP2006], and Cenicerros and Hou [JCP2001]; and **moving mesh methods based on the harmonic mapping** of Dvinsky [JCP1991], and Li, Tang, and Zhang [JCP2001, JCP2002, SISC2005]. Computational costs of moving mesh methods can be efficiently saved with locally varying time steps [Tan et al. JCP2004].

Objective: **Develop an adaptive moving mesh method for the INSEs coupled the phase field model**

## 2. Governing Equations

- Let  $\Omega_p$  be a 2D (physical) domain with the Cartesian Coordinate system  $x = (x, y)$ , and assume that  $\Omega_p$  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. A phase function  $\phi(x, t)$  is introduced to represent the interface at the time  $t$  by the set  $\{x : \phi(x, t) = 0\}$ , and label the inside and the outside of the bubble as  $\{x : \phi(x, t) > 0\}$  and  $\{x : \phi(x, t) < 0\}$ , respectively.
- The dynamics of the phase function  $\phi(x, t)$  can be relaxed (approximated) according to either **Allen-Cahn** or **Cahn-Hilliard** types 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 4th-order differential operators, this work chooses to use the modified (fluid transported) Allen-Cahn phase equation

$$\begin{cases} \phi_t + \mathbf{u} \cdot \text{grad} \phi = \gamma(\Delta \phi - F'(\phi) + \zeta(t)), \\ \frac{d}{dt} \int_{\Omega_p} \phi \, dx = 0, \end{cases} \quad (2.1)$$

where  $\zeta(t)$  is the Lagrangian multiplier corresponding to the constant volume constraint in the last equation, and  $\gamma$  denotes the elastic relaxation time. In Eq.(2.1),  $F(\phi) = (\phi^2 - 1)^2 / 4\hat{\eta}^2$  is the usual double-well potential,  $\hat{\eta}$  represents the capillary width (width of the mixing layer).

- 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

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

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

- 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:

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

$$\text{div}\mathbf{u} = 0, \quad (2.4)$$

where  $\mathbf{g}$  is the external body force,  $p$  is the pressure,  $\nu$  is the kinematic viscosity,  $\lambda$  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 equation (2.3) can be simplified as follows:

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

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

$$u|_{t=0} = u_0(x), \quad \phi|_{t=0} = \phi_0(x), \quad x \in \Omega_p, \quad (2.6)$$

and boundary conditions

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

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

$$\zeta(t) = \frac{1}{|\Omega_p|} \int_{\Omega_p} F'(\phi(x, t)) \, dx,$$

where  $|\Omega_p|$  denotes the area of  $\Omega_p$ .

- **Remark:** As  $\hat{\eta} \rightarrow 0$ , the Cahn-Hilliard phase model and Allen-Cahn type of phase equation coupling with the INSEs (2.3) and (2.4) will tend to the same limit, see [C. Liu & J. Shen, *Physica D*, **179** (2003), pp. 211–228] as well as [N.D. Alikakos, P.W. Bates, and X.F. Chen, *Arch. Rational Mech. Anal.*, **128**(1994), pp. 165–265]. ]

### 3. Numerical Method

- Our adaptive phase field method consists of two independent parts: **the evolution of the governing equation** and **the mesh iterative redistribution**.
- In the following, the INSEs (2.3)-(2.4) will be solved by using the **rotational incremental pressure–correction scheme** of Timmermans, Mineev, and Van De Vosse [Int. J. Numer. Meth. Fluids, 22 (1996), pp. 673–688] on a fixed half-staggered mesh, and a conservative scheme is given for the phase field equation (2.1). **Its main advantage** is that it can overcome the difficulty caused by the artificial pressure Neumann BC.

### 3.1. INSE solver

Give a partition of the physical domain  $\Omega_p$ ,  $\{A_{j+\frac{1}{2},k+\frac{1}{2}} | j, k \in \mathbb{Z}\}$ , a uniform partition of the computational or logical domain  $\Omega_c$  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  $x_{j,k}$ ,  $x_{j+1,k}$ ,  $x_{j+1,k+1}$ , and  $x_{j,k+1}$ . For convenience, define  $I_o := \{x_{j+\frac{1}{2},k+\frac{1}{2}} \in \Omega_p, j, k \in \mathbb{Z}\}$  and  $I_n := \{x_{j,k} \in \Omega_p, j, k \in \mathbb{Z}\}$ , where  $x_{j+\frac{1}{2},k+\frac{1}{2}}$  denotes the centroid coordinates of the quadrangle  $A_{j+\frac{1}{2},k+\frac{1}{2}}$ . Let the scalar variables, e.g.  $p$  and  $\phi$ , be approximated at cell centers by their cell averages

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

and the vectors such as the velocity  $u$  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  $\text{grad}_h$ ,  $\text{div}_h$ , and  $\Delta_h$  are used to denote the discretizations of the gradient, divergence, and Laplacian operators, respectively, but will be given later, where  $\Delta_h = \text{div}_h \text{grad}_h$ .

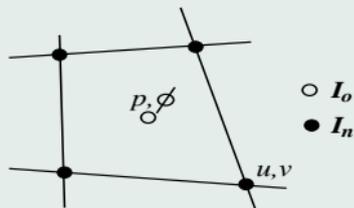


Figure 3.1: Half-staggered mesh, see [A. Geoge, L.C. Huang, W.P. Tang, and Y.D. Wu, *SIAM J. Sci. Comput.*, **21** (2000), pp. 2331–2351; G.H. Golub, L.C. Huang, H. Simon, and W.P. Tang, *SIAM J. Sci. Comput.*, **19** (1998), pp. 1606–1624].

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

### Algorithm 1

**Step 0** Give initial data  $\phi^n, u^n, p^n$ .

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

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

with homogeneous Dirichlet boundary conditions

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

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

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

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

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

The projection step is equivalent to solving a PPE together with the homogeneous Neumann BC

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

**Step 3 (Phase evolution step)** Compute the phase field  $\phi^{n+1}$  by

$$\frac{\phi^{n+1} - \phi^n}{\Delta t_n} - \gamma \Delta_h \phi^{n+1} = -\operatorname{div}_h(u^{n+1} \phi^n) - \gamma f(\phi^n) + \gamma \zeta(t^n)$$

on  $I_o$ . (3.6)

The detailed definitions of the discrete operators  $\operatorname{grad}_h$ ,  $\operatorname{div}_h$ , and  $\Delta_h$  used in the above projection method are presented as follows. The gradient, divergence, and Laplacian operators are first transferred into the logical domain  $\Omega_c$  by the coordinate transformation  $x = x(\xi)$  as

follows

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

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

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

where  $\mathbf{J} = x_\xi\mathbf{y}_\eta - x_\eta\mathbf{y}_\xi$  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

$$\text{grad}_h \psi = \frac{1}{\widehat{J}} \left( \widehat{(\mathbf{y}_\eta)} \widehat{(\psi_\xi)} - \widehat{(\mathbf{y}_\xi)} \widehat{(\psi_\eta)}, -\widehat{(\mathbf{x}_\eta)} \widehat{(\psi_\xi)} + \widehat{(\mathbf{x}_\xi)} \widehat{(\psi_\eta)} \right)^T, \quad \text{on } I_n, \quad (3.10)$$

where

$$\widehat{(\psi_\xi)}_{j,k} := (\psi_{j+\frac{1}{2},k+\frac{1}{2}} - \psi_{j-\frac{1}{2},k+\frac{1}{2}} + \psi_{j+\frac{1}{2},k-\frac{1}{2}} - \psi_{j-\frac{1}{2},k-\frac{1}{2}}) / 2,$$

$$\widehat{(\psi_\eta)}_{j,k} := (\psi_{j+\frac{1}{2},k+\frac{1}{2}} - \psi_{j+\frac{1}{2},k-\frac{1}{2}} + \psi_{j-\frac{1}{2},k+\frac{1}{2}} - \psi_{j-\frac{1}{2},k-\frac{1}{2}}) / 2,$$

$$\widehat{(\mathbf{Z}_\xi)}_{j,k} := (\mathbf{Z}_{j+1,k} - \mathbf{Z}_{j-1,k}) / 2, \quad \widehat{(\mathbf{Z}_\eta)}_{j,k} := (\mathbf{Z}_{j,k+1} - \mathbf{Z}_{j,k-1}) / 2,$$

$$\widehat{J}_{j,k} = (\widehat{\mathbf{x}_\xi} \widehat{\mathbf{y}_\eta} - \widehat{\mathbf{x}_\eta} \widehat{\mathbf{y}_\xi})_{j,k}, \quad \mathbf{Z} = \mathbf{x} \text{ or } \mathbf{y}.$$

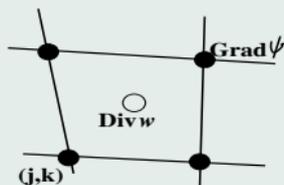


Figure 3.2: Half-staggered mesh for  $\text{div}_h$  and  $\text{grad}_h$ .

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

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

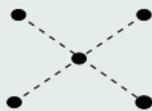
where

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

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 [M. Shashkov and S. Steinberg, *Conservative Finite-Difference Methods on General Grids*, CRC Press, 1996] 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 multi-level dissection method, see [A. Geoge, L.C. Huang, W.P. Tang, and Y.D. Wu, *SIAM J. Sci. Comput.*, **21** (2000), pp. 2331–2351; G.H. Golub, L.C. Huang, H. Simon, and W.P. Tang, *SIAM J. Sci. Comput.*, **19** (1998), pp. 1606–1624]. In this study, the AMG package [R. Li & W.B. Liu, *The AFEPack Handbook*, <http://circus.math.pku.edu.cn/A> 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 5 point scheme of the Poisson equation, see Fig. 3.3(a). The solution  $\psi = (\dots, \psi_{j+\frac{1}{2}, k+\frac{1}{2}}, \dots)^T$  has two degrees of freedom, that say,  $\psi + \psi_1 + c\psi_2$  is also its solution, where  $c$  is an arbitrary constant,  $\psi_1$  is a constant vector, see Fig. 3.3(b), and  $\psi_2$  is a so-called “checkerboard mode”, see Fig. 3.3(c). However, neither  $\psi_1$  nor  $\psi_2$  will affect  $\text{grad}_h \psi$ , see (3.10).



(a) Skewed scheme;

$c$	$c$	$c$	$c$	
$c$	$c$	$c$	$c$	
$c$	$c$	$c$	$c$	

(b)  $\psi_1$ ;

	$-c$	$c$	$-c$	$c$	$-c$
$c$	$-c$	$c$	$-c$	$c$	$-c$
	$-c$	$c$	$-c$	$c$	$-c$
$c$	$-c$	$c$	$-c$	$c$	$-c$

(c)  $\psi_2$

Figure 3.3: Stencil of the skewed scheme and profile of the vector  $\psi_i$ ,  $i = 1, 2$ .

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

**Remark 3.1** *grad $\psi$  and 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 approach will also be used to project the velocity vector onto a divergence-free space at the end of the iterative mesh redistribution.*

**Remark 3.2** *The gradient operator in the momentum equation (3.2) may be either non-conservative or conservative. Moreover, the Laplacian operators in (3.2) and (3.6) may also be a 2nd-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 [R. Li and W.B. Liu, The AFEPack Handbook, <http://circus.math.pku.edu.cn/AFEPack>*

*J. The tolerance for the  $l^2$  norm of the residual is set to be  $10^{-8}$  and the average iterative number  $3 \sim 5$ .*

**Remark 3.3** *To preserve the mass-conservation of the phase field  $\phi$ , 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  $|\Omega_p^n| = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^n|$ .*

## 3.2. Adaptive mesh redistribution

This section introduces our adaptive mesh redistribution briefly. The readers are referred to [H.Z. Tang & T. Tang, SIAM J. Numer. Anal. 2003] for detailed descriptions.

### Iterative mesh redistribution

Let  $x = (x, y)$  and  $\xi = (\xi, \eta)$  denote the physical and logical or computational coordinates, respectively. A one-to-one coordinate transformation from the logical or computational domain  $\Omega_c$  to the physical domain  $\Omega_p$  is denoted by

$$x = x(\xi), \quad \xi \in \Omega_c. \quad (3.12)$$

Our attention is limited to the case of that the physical domain  $\Omega_p$  is convex and the map (3.12) is to find the minimizer of the following functional

$$\tilde{E}(x) = \frac{1}{2} \sum_{i=1}^2 \int_{\Omega_c} (\tilde{\nabla} x_i)^T G_i \tilde{\nabla} x_i d\xi, \quad (3.13)$$

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  $G_i = \omega I$ ,  $i = 1, 2$ , where  $I$  denotes the identity matrix and  $\omega$  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.

Using Winslow's choice, one deduces the Euler-Lagrange equations

of the functional (3.13) to

$$\tilde{\nabla} \cdot (\omega \tilde{\nabla} x) = 0. \quad (3.14)$$

In this study,  $\omega = \omega(\text{grad}\phi)$ . Eq. (3.14) is first discretized by the central difference scheme and then solved by the Gauss-Seidel (GS) iteration, that is to say,

$$\begin{aligned} &\omega_{j+\frac{1}{2},k}^{[m]} \left( x_{j+1,k}^{[m]} - x_{j,k}^{[m+1]} \right) - \omega_{j-\frac{1}{2},k}^{[m]} \left( x_{j,k}^{[m+1]} - x_{j-1,k}^{[m+1]} \right) \\ &+ \omega_{j,k+\frac{1}{2}}^{[m]} \left( x_{j,k+1}^{[m]} - x_{j,k}^{[m+1]} \right) - \omega_{j,k-\frac{1}{2}}^{[m]} \left( x_{j,k}^{[m+1]} - x_{j,k-1}^{[m+1]} \right) = 0, \end{aligned} \quad (3.15)$$

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  $\|x^{[m]} - x^{[m+1]}\| < \epsilon$  or  $m < \mu$ , where  $\mu$  is a given small integer. In practice, a few iterations (say  $\mu = 3$  to 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

$$\omega_{j,k} \leftarrow \frac{4}{16}\omega_{j,k} + \frac{2}{16}(\omega_{j+1,k} + \omega_{j-1,k} + \omega_{j,k+1} + \omega_{j,k-1}) \\ + \frac{1}{16}(\omega_{j-1,k-1} + \omega_{j-1,k+1} + \omega_{j+1,k-1} + \omega_{j+1,k+1}).$$

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

## Interpolation of the cell-averaged variables

After each GS iterative step, the approximate solutions need to be remapped onto the newly resulted mesh  $\{x_{j,k}^{[m+1]}\}$  from the old mesh  $\{x_{j,k}^{[m]}\}$ . The remapping procedure of the cell-averaged variables such as the phase field  $\phi$  and  $p$  can be conducted by using the conservative interpolation technique proposed by Tang & Tang [SINUM2003], which is

$$\begin{aligned} |A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]}| U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]} &= |A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}| U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]} - \left( (c_n^2 U^{[m]})_{j+1,k+\frac{1}{2}} \right. \\ &\quad \left. + (c_n^4 U^{[m]})_{j,k+\frac{1}{2}} \right) - \left( (c_n^3 U^{[m]})_{j+\frac{1}{2},k+1} + (c_n^1 U^{[m]})_{j+\frac{1}{2},k} \right), \end{aligned} \quad (3.16)$$

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  $n^l =$

$(n_x^l, n_y^l)$ , defined as follows:

$$c_1^n = \frac{1}{2}(c_{j,k}^x + c_{j+1,k}^x)(y_{j+1,k} - y_{j,k}) - \frac{1}{2}(c_{j,k}^y + c_{j+1,k}^y)(x_{j+1,k} - x_{j,k}),$$

$$c_2^n = \frac{1}{2}(c_{j+1,k}^x + c_{j+1,k+1}^x)(y_{j+1,k+1} - y_{j+1,k}) - \frac{1}{2}(c_{j+1,k}^y + c_{j+1,k+1}^y)(x_{j+1,k+1} - x_{j+1,k}),$$

$$c_3^n = \frac{1}{2}(c_{j+1,k+1}^x + c_{j,k+1}^x)(y_{j,k+1} - y_{j+1,k+1}) - \frac{1}{2}(c_{j+1,k+1}^y + c_{j,k+1}^y)(x_{j,k+1} - x_{j+1,k+1}),$$

$$c_4^n = \frac{1}{2}(c_{j,k+1}^x + c_{j,k}^x)(y_{j,k} - y_{j,k+1}) - \frac{1}{2}(c_{j,k+1}^y + c_{j,k}^y)(x_{j,k} - x_{j,k+1}),$$

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

$$(c_n^2 U)_{j+1, k+\frac{1}{2}} = \frac{c_n^2}{2}(U_{j+\frac{3}{2}, k+\frac{1}{2}} + U_{j+\frac{1}{2}, k+\frac{1}{2}}) - \frac{|c_n^2|}{2}(U_{j+\frac{3}{2}, k+\frac{1}{2}} - U_{j+\frac{1}{2}, k+\frac{1}{2}}). \quad (3.17)$$

The above approximation is only first order accurate in space. In order to avoid large numerical dissipation, the reconstruction technique

will be used, and (3.17) is replaced by

$$(c_n^2 U)_{j+1, k+\frac{1}{2}} = \frac{c_n^2}{2} (U_{j+1+0, k+\frac{1}{2}} + U_{j+1-0, k+\frac{1}{2}}) \quad (3.18)$$

$$- \frac{|c_n^2|}{2} (U_{j+1+0, k+\frac{1}{2}} - U_{j+1-0, k+\frac{1}{2}}), \quad (3.19)$$

with

$$U_{j+1+0, k+\frac{1}{2}} = U_{j+\frac{3}{2}, k+\frac{1}{2}} - \frac{1}{2} S_{j+\frac{3}{2}, k+\frac{1}{2}}^\xi,$$

$$U_{j+1-0, k+\frac{1}{2}} = U_{j+\frac{1}{2}, k+\frac{1}{2}} + \frac{1}{2} S_{j+\frac{1}{2}, k+\frac{1}{2}}^\xi.$$

Here  $S^\xi$  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  $vLL(a, b)$  denotes van Leer's limiter defined by

$$vLL(a, b) = (\text{sign}(a) + \text{sign}(b)) \frac{|ab|}{|a| + |b| + \varepsilon}, \quad (3.20)$$

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

The formula (3.16) is obtained by using the classical perturbation theory, and satisfies the following mass-conservation property:

$$\sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]}| |U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m+1]}| = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}| |U_{j+\frac{1}{2},k+\frac{1}{2}}^{[m]}|, \quad U = \phi \text{ or } p. \quad (3.21)$$

Some further theoretical properties of this conservative interpolation can be found in [H.Z. Tang & T. Tang, SINUM2003].

## Interpolation of the velocity variables

This subsection begins to remap the velocity variables onto the newly resulted mesh  $\{x_{j,k}^{[m+1]}\}$  from the old mesh  $\{x_{j,k}^{[m]}\}$ . For the velocity vector  $u = (u, v)$ , the high-resolution, nonconservative interpolation of Tang, Tang, and Zhang [JCP2003] is employed, which is obtained by using Taylor's expansion, i.e.

$$u(x_{j,k}^{[m+1]}) \approx u(x_{j,k}^{[m]}) - (x_{j,k}^{[m]} - x_{j,k}^{[m+1]}) \cdot \text{grad}u(x_{j,k}^{[m]}). \quad (3.22)$$

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

$$\begin{aligned} u_{j,k}^{[m+1]} = & u_{j,k}^{[m]} - \frac{1}{2} \left( c_{j,k}^{\xi} (v_{j+0,k}^{[m]} + v_{j-0,k}^{[m]}) - |c_{j,k}^{\xi}| (v_{j+0,k}^{[m]} - v_{j-0,k}^{[m]}) \right) \\ & - \frac{1}{2} \left( c_{j,k}^{\eta} (w_{j,k+0}^{[m]} + w_{j,k-0}^{[m]}) - |c_{j,k}^{\eta}| (w_{j,k+0}^{[m]} - w_{j,k-0}^{[m]}) \right) \end{aligned} \quad (3.23)$$

where

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

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

and

$$v_{j+0,k} = \Delta_\xi u_{j,k} - \frac{1}{2} \text{vLL}(\Delta_\xi u_{j+1,k} - \Delta_\xi u_{j,k}, \Delta_\xi u_{j,k} - \Delta_\xi u_{j-1,k}),$$

$$v_{j-0,k} = \Delta_\xi u_{j-1,k} + \frac{1}{2} \text{vLL}(\Delta_\xi u_{j,k} - \Delta_\xi u_{j-1,k}, \Delta_\xi u_{j-1,k} - \Delta_\xi u_{j-2,k}),$$

$$w_{j,k+0} = \Delta_\eta u_{j,k} - \frac{1}{2} \text{vLL}(\Delta_\eta u_{j,k+1} - \Delta_\eta u_{j,k}, \Delta_\eta u_{j,k} - \Delta_\eta u_{j,k-1}),$$

$$w_{j,k-0} = \Delta_\eta u_{j,k-1} + \frac{1}{2} \text{vLL}(\Delta_\eta u_{j,k} - \Delta_\eta u_{j,k-1}, \Delta_\eta u_{j,k-1} - \Delta_\eta u_{j,k-2}),$$

here  $\Delta_\xi u_{j,k} = u_{j+1,k} - u_{j,k}$ ,  $\Delta_\eta u_{j,k} = u_{j,k+1} - u_{j,k}$ . The function  $\text{vLL}(a, b)$  denotes van Leer's limiter, see (3.20).

It is worth noting that the velocity  $u_{j,k}^{[m+1]}$  updated by (3.23) is not divergence-free generally. So  $u_{j,k}^{[\mu]}$  cannot be considered as the “initial” data of the INSEs, that is to say, one has to project  $u_{j,k}^{[\mu]}$  onto the divergence-free space before evolving the INSEs. Following the projection method for the INSEs, the standard Helmholtz decomposition is used:

$$\begin{cases} u^{n+1} = u^{[\mu]} - \text{grad}_h \psi, & \text{div}_h u^{n+1} = 0, \\ u^{n+1} \cdot n = 0, & \text{on } \partial\Omega_p. \end{cases} \quad (3.24)$$

to get the divergence-free velocity vector  $u^{n+1}$  on the resultant adaptive mesh  $\{x_{j,k}^{[\mu]}\}$ . It is equivalent to solving the Neumann BVP of a Poisson equation:

$$\begin{cases} \Delta_h \psi = \text{div}_h u^{[\mu]}, \\ \text{grad}_h \psi \cdot n = 0, & \text{on } \partial\Omega_p. \end{cases} \quad (3.25)$$

Definitions of the above discrete operators are similar to those given before.

### 3.3. 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. In each iteration of the second part, mesh points are first redistributed by the GS method (3.15), and then  $\phi$  and  $p$  are updated on the newly generated meshes by the conservative-interpolation formula (3.16), while the velocity vector  $u$  is remapped by the non-conservative approach (3.23) and corrected finally by (3.24). The solution procedure can be illustrated by the following flowchart:

## Algorithm 2

**Step 0** Give an initial adaptive mesh  $x_{j,k}^n$  based on the initial function,  $n \geq 0$ .

**Step 1** Advance the solution one time step  $\Delta t_n$  by **Algorithm 1** to get  $u^{n+1}$ ,  $p^{n+1}$  and  $\phi^{n+1}$ .

**Step 2** Set  $x_{j,k}^{[0]} := x_{j,k}^n$ ,  $u_{j,k}^{[0]} := u_{j,k}^{n+1}$ , 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:

- Solve the mesh redistributing equation (a generalized Laplacian equation) by one Gauss-Seidel iteration to get  $x^{[m+1]}$ ;
- Remap the approximate solutions on the new grid  $x^{[m+1]}$  to get  $u^{[m+1]}$ ,  $p^{[m+1]}$  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  $u_{j,k}^{n+1}$ , and reset  $x_{j,k}^{n+1} := x_{j,k}^{[\mu]}$  and  $U_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} := U_{j+\frac{1}{2},k+\frac{1}{2}}^{[\mu]}$ ,  $U = \phi$  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 3.4** The conservative interpolation (3.16) and the non-conservative interpolation (3.23) are carried out after each GS iteration step, while the divergence free correction in (3.24) is only performed once after all  $\mu$  GS iterations are finished. In our computations, the parameter  $\mu$  is taken as  $3 \sim 5$  to get a satisfactory mesh redistribution.

**Remark 3.5** Using **Algorithm 2** can get a new set of mesh points and cor-

responding solutions, e.g.  $\{x^{n+1}, \phi^{n+1}, p^{n+1}, u^{n+1}\}$ , which satisfy

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

$$(\operatorname{div}_h u)_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = 0,$$

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

## 4. Numerical Results

**Example 1** We start with a square bubble centered at  $(\pi, \pi)$ . The length of the square is 2.

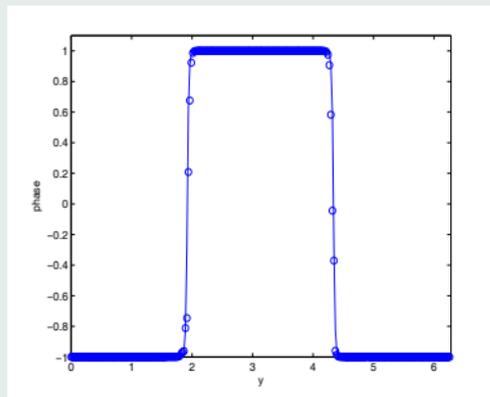
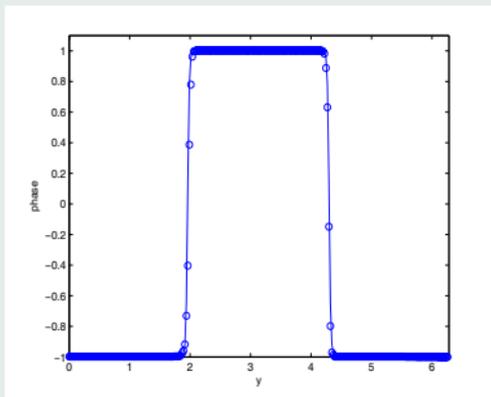


Figure 4.1: Example 1: the phase  $\phi$  along  $x = \pi$ . The symbol “circle” and solid line denote the adaptive solution with a resolution of  $64 \times 64$  and the computed solution obtained on a  $256 \times 256$  uniform mesh. Left:  $t = 0.3$ ; right:  $t = 0.5$ .

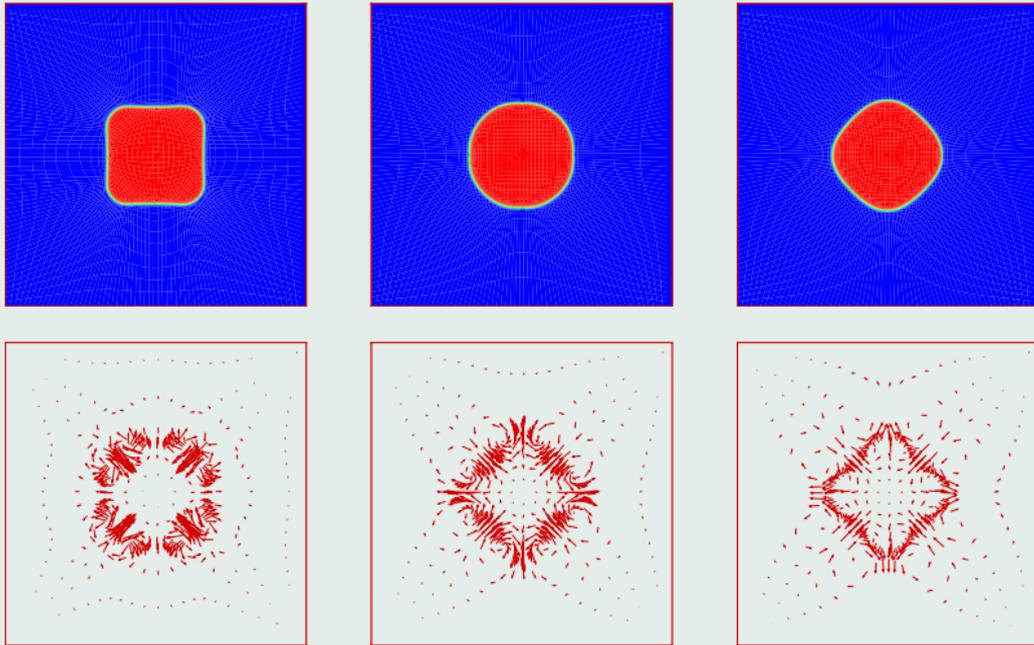


Figure 4.2: Example 1: the phase evolution, velocity field and mesh redistribution at  $t = 0.1, 0.3, 0.5$ .

**Example 2** We start with two initially kissing unit cycle bubbles. The radius of each bubble is unit.

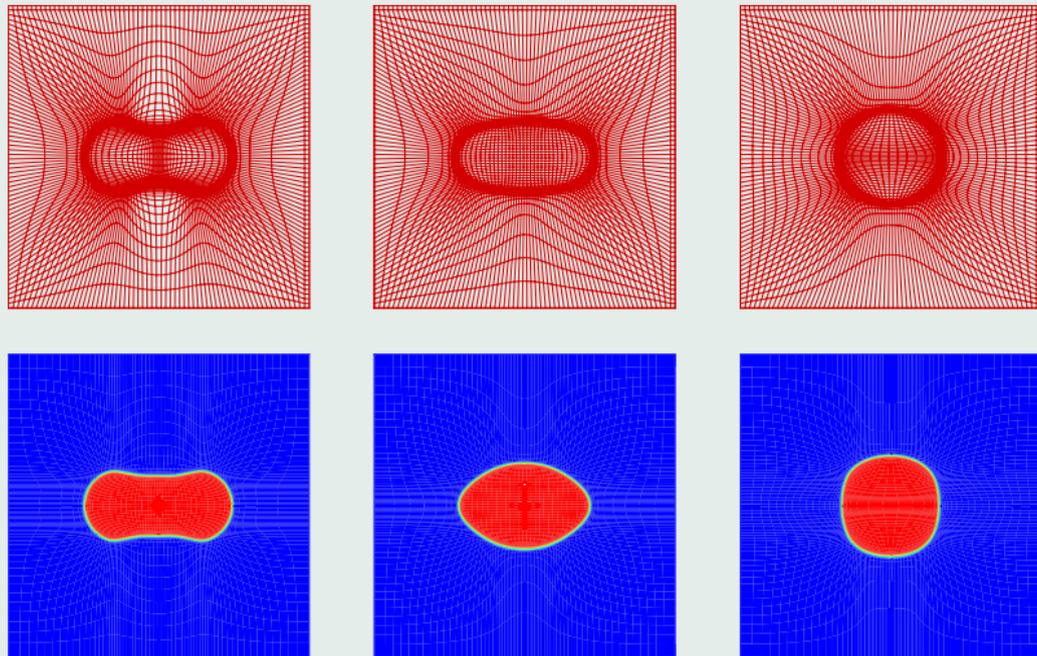


Figure 4.3: Example 2: Phase evolution at  $t = 0.2, 0.5, 0.8$ .

**Example 3** Consider three kissing bubbles. Inside the bubbles  $\phi = 1$ ; outside the bubbles  $\phi = -1$ . The radius of each bubble is  $\pi/4$ .

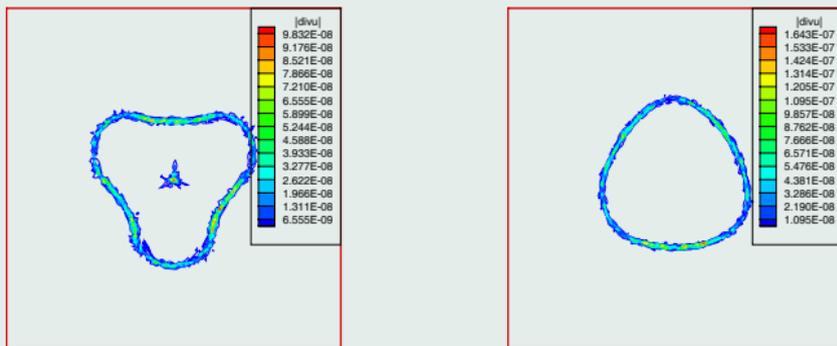


Figure 4.4: Example 3: the divergence of velocity  $|\text{div}_h u|$  at  $t = 0.2$ (left),  $1.0$ (right).

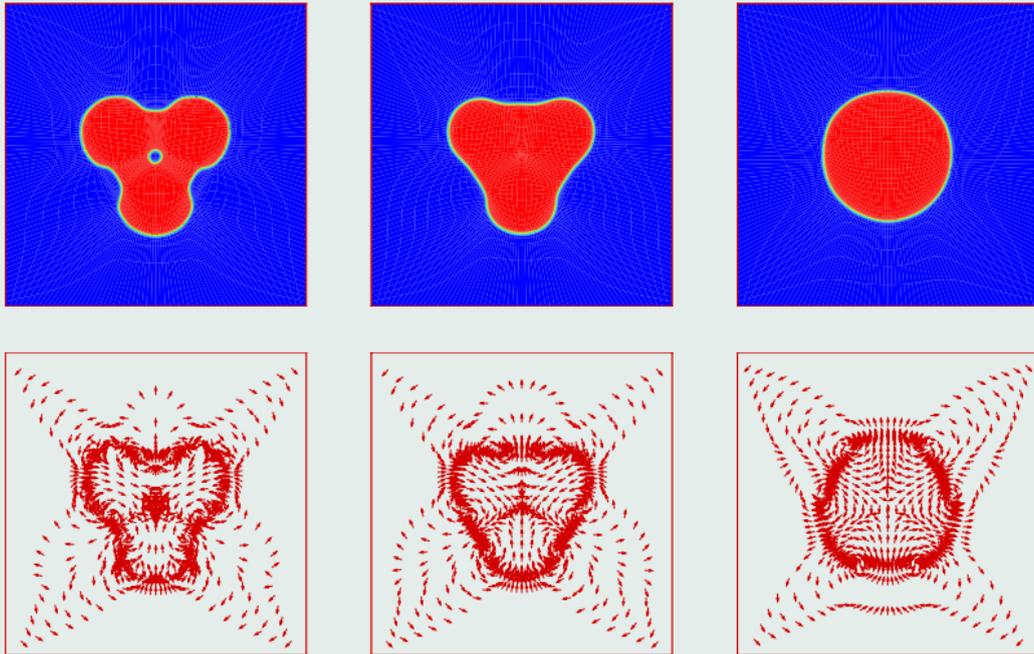


Figure 4.5: Example 3: the development of phase, velocity at  $t = 0.1, 0.2, 0.5$ .

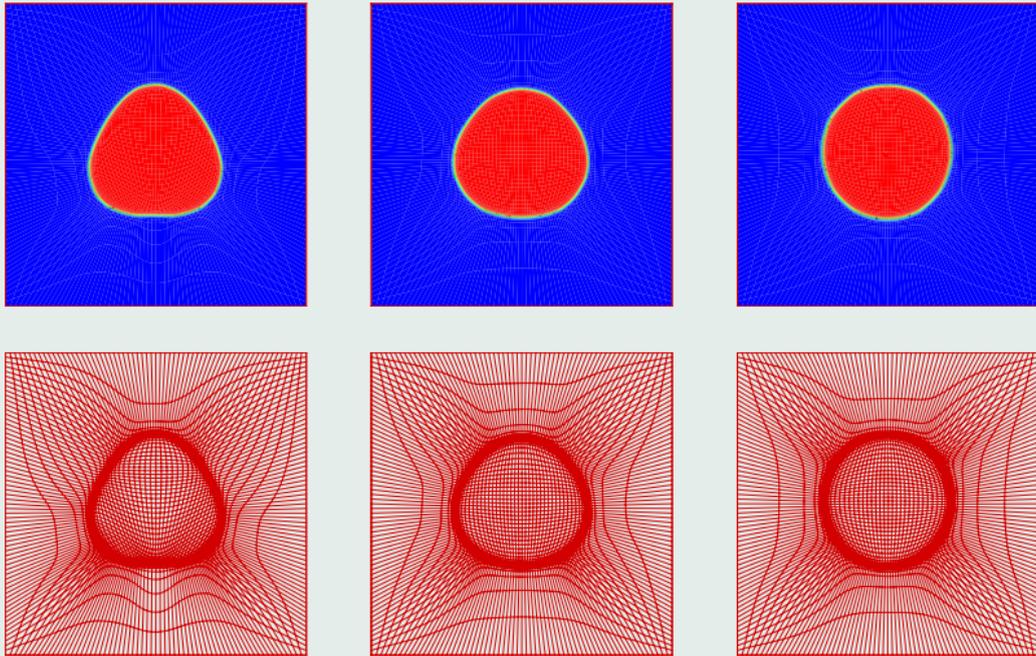


Figure 4.6: Example 3: the development of phase and mesh redistribution at  $t = 1.0, 1.2, 2.0$ .

**Example 4** Consider the gravitational effect. We start with a bigger circular bubble higher than the smaller bubble. The density of the bubbles is lighter than the density of the surrounding fluid.  $\mathbf{g} = (0, 1)^T$ . The NS equations become

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

where  $\rho_1 - \rho_2 = -1$ .



Figure 4.7: Example 4:  $|\text{div}_h \mathbf{u}|$  at  $t = 0.2$ (left) and  $t = 1.5$ (right).

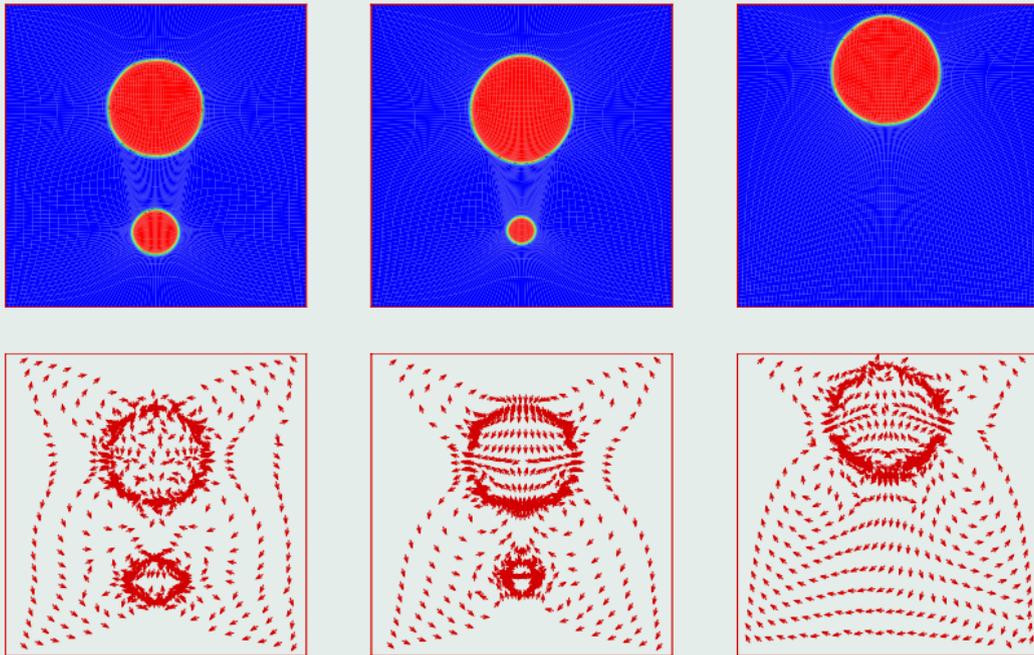


Figure 4.8: Example 4: the phase and velocity at  $t = 0.2, 1.2, 6.0$ .

**Example 5** Consider the gravitational effect on three bubbles. Initially, the two kissing bubbles are located higher than the third one which is smaller.  $g = (0, 0.1)^T$ .

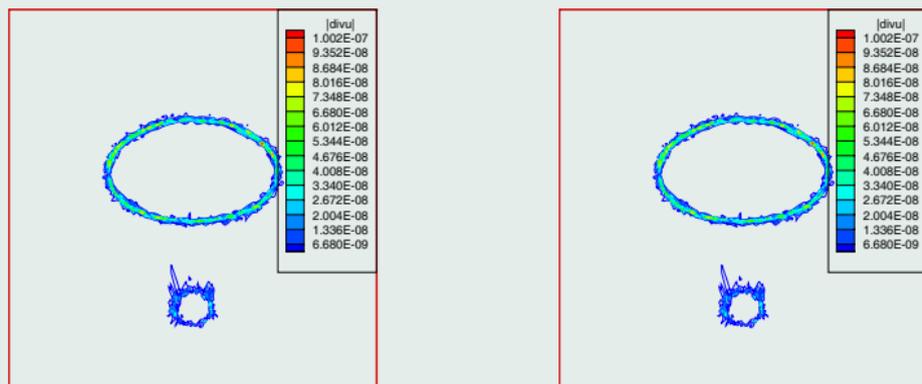


Figure 4.9: Example 5:  $|\text{div}_h u|$  at  $t = 0.5$ (left) and  $1.5$ (right).

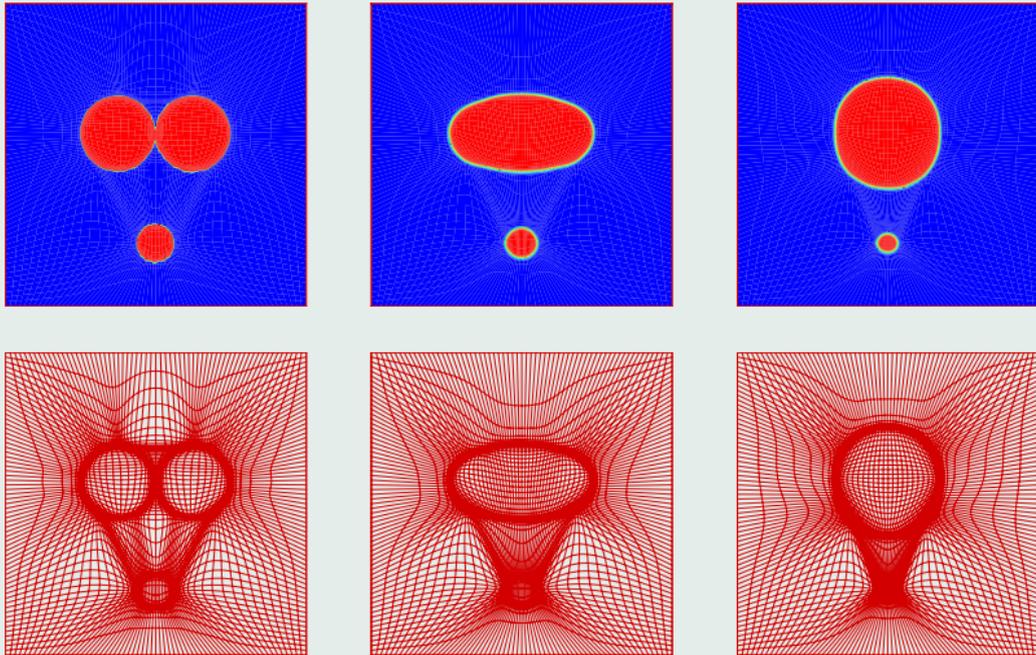


Figure 4.10: Example 5: the phase field and mesh distribution at  $t = 0, 0.4, 1.0$ .

## 5. Conclusion

- We have 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 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 1st part, the rotational incremental pressure-correction scheme is used to solve the INSEs on a fixed half-staggered

mesh, and the Allen-Cahn type of phase equation modified by the Lagrangian multiplier is approximated by a conservative, 2nd-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 resulted 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.

## Some references

1. A. Geoge, L.C. Huang W.P. Tang and Y.D. Wu, Numerical simulation of unsteady incompressible flow on the curvilinear half-staggered mesh. *SIAM J. Sci. Comput.*, **21** (2000), pp. 2331-2351.
2. C. Liu and J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method. *Physica D*, **179** (2003), pp. 211-228.
3. H.Z. Tang and T. Tang, Moving mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J. Numer. Anal.*, **41** (2001), pp. 487-515.
4. 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** (2003), pp. 543-572.

---

# Thanks!

---