

Simulating Two-Phase Viscoelastic Flows Using Moving Finite Element Methods

Yubo Zhang¹, Heyu Wang^{2*} and Tao Tang¹

¹ *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.*

² *Department of Mathematics, Zhejiang University, Hangzhou, Zhejiang 310027, China.*

Received 4 February 2009; Accepted (in revised version) 8 June 2009

Available online 24 August 2009

Abstract. Phase-field models provide a way to model fluid interfaces as having finite thickness; the interface between two immiscible fluids is treated as a thin mixing layer across which physical properties vary steeply but continuously. One of the main challenges of this approach is in resolving the sharp gradients at the interface. In this paper, moving finite-element methods are used to simulate interfacial dynamics of two-phase viscoelastic flows. The finite-element scheme can easily accommodate complex flow geometry and the moving mesh strategy can cluster more grid points near the thin interfacial areas where the solutions have large gradients. A diffused monitor function is used to ensure high quality meshes near the interface. Several numerical experiments are carried out to demonstrate the effectiveness of the moving mesh strategy.

AMS subject classifications: 65M05, 76M10, 76T10

Key words: Moving finite-element method, two-phase flow, viscoelastic flow, non-Newtonian flow.

1 Introduction

Modeling and simulating two-phase viscoelastic flows have been challenging both mathematically and technically. There have been many computational techniques developed to tackle the problem, including diffuse-interface methods [1], interface tracking methods [17], level-set methods [16, 20], finite-element methods with adaptive mesh refinements [22] and spectral methods with adaptive mesh redistribution [7]. The governing

*Corresponding author. *Email addresses:* zybx@msn.com (Y. Zhang), wangheyu@zju.edu.cn (H. Wang), ttang@math.hkbu.edu.hk (T. Tang)

equations require additional constitutive equations for stress tensor and numerical computations require higher mesh resolution around fluid interfaces. Unlike the Newtonian flow whose local stress is proportional to the local strain rate, for polymer fluids the local stress usually depends on its deformation history due to the long chain molecular structure. Therefore an additional constitutive relation between the stress and the strain of flow such as Upper Convected Maxwell (UCM) model by Oldroyd [13] should be coupled into the flow system.

When comes to multi-phase flows, problems arise due to the sharp interface between fluids. The classical jump conditions and surface tension are introduced into the Navier-Stokes equations in which the interface has zero width. Different approaches such as volume-of-fluid (VOF) and level-set method have been developed to handle this problem, see, e.g., [16, 20]. Compared to sharp interface methods, diffuse interface methods are based on a different theoretical model where the interface is treated as a smooth transition region from one phase to another. The original idea can be found in [2, 3], and [1] is a useful review of the diffuse interface model. In [12, 22], diffuse interface model is applied for Newtonian and non-Newtonian flows.

The main challenge for simulating the phase-field evolution is that very fine meshes are needed for resolving thin interfaces. In order to produce physically correct results one needs a very thin interface and it is almost impossible to solve the problem practically when using uniform meshes. In past years, many adaptive mesh techniques have been proposed which can be classified as adaptive mesh refinement methods and adaptive mesh redistribution methods, see [14]. In this work, we will simulate multi-phase flows using a moving mesh method (i.e., adaptive mesh redistribution method in the sense of [14]). In particular, we will use the moving mesh algorithms developed in Li et al. [10, 11] which redistribute mesh nodes based on harmonic mapping. The moving mesh method based on harmonic mapping has been applied successfully to several complex problems including incompressible flow [5, 6], reaction-diffusion systems [15], and dendritic growth [18, 19]. The goal of the moving mesh method is to reduce the computational cost and to enhance the accuracy in resolving the diffuse interfaces.

In this work, the Oldroyd-B model for constitutive relation of viscoelastic fluids is used. The following section will briefly review the Oldroyd-B model for viscoelastic flow and the phase-field model for two-phase flows. Section 3 will describe the coupled equations and the finite-element formulation used in our computations. The moving mesh methods and the corresponding monitor function will be discussed in Section 4 and several numerical tests will be given in Section 5. The last section draws the conclusion with some discussions on the future works.

2 The Oldroyd-B model and the phase-field model

In this section, we will briefly review the governing equations for viscoelastic flows and the phase-field model for multi-phase flows.

2.1 The Oldroyd-B model for polymer fluids

The incompressible Navier-Stokes equations read

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1a)$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \nabla \cdot (-p\mathbf{I} + \mathbf{T}) + \rho \mathbf{g}, \quad (2.1b)$$

where \mathbf{u} is velocity, ρ is density, p is pressure, \mathbf{g} is gravity and \mathbf{T} is extra-stress tensor. In Newtonian fluids the extra-stress tensor is $\mathbf{T} = 2\mu\mathbf{D}$, and $\mathbf{D} = \frac{1}{2} [(\nabla \mathbf{u})^T + \nabla \mathbf{u}]$ is the strain rate.

To study viscoelastic fluids, the Upper Convected Maxwell (UCM) model is a simple one to begin with. In this model the relation between extra-stress tensor and strain rate is given by

$$\mathbf{T} + \lambda \mathbf{T}^\nabla = 2\mu\mathbf{D},$$

where λ is the relaxation time of stress, μ is the total viscosity and

$$\mathbf{T}^\nabla = \frac{\partial \mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} - (\nabla \mathbf{u})^T \cdot \mathbf{T} - \mathbf{T} \cdot (\nabla \mathbf{u})$$

is the upper-convected derivative of stress tensor.

In Oldroyd-B model the extra-stress tensor is divided into polymeric and Newtonian parts,

$$\mathbf{T} = \tau + 2\mu_s \mathbf{D},$$

where μ_s is the Newtonian viscosity. The constitutive relation between polymeric stress and strain rate is given by

$$\tau + \lambda \tau^\nabla = 2\mu_p \mathbf{D},$$

where $\mu_p = \mu - \mu_s$ is the polymeric viscosity. Now the conservation of mass, momentum with constitutive equations can be written as

$$\nabla \cdot \mathbf{u} = 0, \quad (2.2a)$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \nabla \cdot (-p\mathbf{I} + \tau + 2\mu_s \mathbf{D}) + \rho \mathbf{g}, \quad (2.2b)$$

$$\tau + \lambda \tau^\nabla = 2\mu_p \mathbf{D}. \quad (2.2c)$$

2.2 Phase-field model for interfacial dynamics

In phase-field model, the fluid phase is represented by a phase function ϕ . Let $\phi = \pm 1$ denote two stable phases and the interface is a smooth transition region where $-1 < \phi < 1$. In order to model the interfacial dynamics, we need to define a free energy of the interface.

The Cahn-Hilliard type free energy of a phase-field is a function of the phase parameter ϕ defined as

$$f(\phi) = \frac{\alpha}{2} |\nabla\phi|^2 + \beta g(\phi), \quad (2.3)$$

where $\sqrt{\alpha\beta}$ produces the surface tension and $\sqrt{\alpha/\beta}$ controls the interfacial width. The first term on the right hand side of (2.3) is the gradient energy and the second one is the bulk energy which has two minima corresponding to two stable phases. The chemical potential is the rate of change of the free energy with respect to ϕ

$$c = \frac{\delta \int_{\Omega} f(\phi) d\Omega}{\delta\phi} = -\alpha \nabla^2 \phi + \beta g'(\phi). \quad (2.4)$$

Finally the phase parameter ϕ is evolved by the Cahn-Hilliard equation:

$$\frac{\partial\phi}{\partial t} = \gamma \nabla^2 c = \gamma \nabla^2 [-\alpha \nabla^2 \phi + \beta g'(\phi)], \quad (2.5)$$

where γ is the mobility. The interfacial width is of order

$$\epsilon \sim \sqrt{\alpha/\beta}. \quad (2.6)$$

If the phase is convected by fluid velocity and if we let

$$\alpha = k\epsilon^2, \quad \beta = k, \quad g(\phi) = \frac{1}{4}(\phi^2 - 1)^2,$$

then

$$\frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = \gamma \nabla^2 [k(\phi^3 - \phi - \epsilon^2 \nabla^2 \phi)], \quad (2.7)$$

which has an equilibrium profile

$$\phi(x) = \tanh\left(\frac{x}{\sqrt{2}\epsilon}\right).$$

As $\epsilon \rightarrow 0$, the ratio α/ϵ produces the interfacial tension σ in the classical sense [9]:

$$\sigma = \frac{2\sqrt{2}}{3} k\epsilon. \quad (2.8)$$

The original fluid momentum equation (2.1b) coupled with surface tension becomes

$$\rho(\phi) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \nabla \cdot (-p\mathbf{I} + \mathbf{T}(\phi)) + c \nabla \phi + \rho(\phi) \mathbf{g}, \quad (2.9)$$

where the density ρ and the stress tensor \mathbf{T} depend on the phase parameter ϕ .

3 The coupled system

For two-phase non-Newtonian and Newtonian mixture, the incompressible Navier-Stokes equations with the Oldroyd-B model are

$$\nabla \cdot \mathbf{u} = 0, \quad (3.1a)$$

$$\rho(\phi) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \nabla \cdot (-p \mathbf{I} + \mathbf{T}(\phi)) + c \nabla \phi + \rho(\phi) \mathbf{g}, \quad (3.1b)$$

$$\tau + \lambda \left[\frac{\partial \tau}{\partial t} + \mathbf{u} \cdot \nabla \tau - (\nabla \mathbf{u})^T \cdot \tau - \tau \cdot (\nabla \mathbf{u}) \right] = 2\mu_p \mathbf{D}, \quad (3.1c)$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \gamma \nabla^2 [k(\phi^3 - \phi - \epsilon^2 \nabla^2 \phi)], \quad (3.1d)$$

where $c = k(\phi^3 - \phi - \epsilon^2 \nabla^2 \phi)$ is the chemical potential, k is a constant determined by (2.8) and γ is the mobility of c . In (3.1a), the density and stress tensors, which depend on the phase function ϕ are given by

$$\rho(\phi) = \frac{1+\phi}{2} \rho_1 + \frac{1-\phi}{2} \rho_2, \quad (3.2a)$$

$$\mathbf{T}(\phi) = \frac{1+\phi}{2} (\tau + 2\mu_{s1} \mathbf{D}) + \frac{1-\phi}{2} (2\mu_{s2} \mathbf{D}). \quad (3.2b)$$

If both components are non-Newtonian, the formulations are similar. Moreover, appropriate boundary conditions should be chosen for specifically given problems.

Although there are many ways to discretize the governing equations above, we seek for ones that are easy to implement. One thing in mind is to make the resulting matrices symmetric positive definite so that we can make use of the Gauss-Seidel-iteration based algebraic multi-grid solver whose average time complexity is $\mathcal{O}(N)$. Let $\hat{\mathbf{u}}$, \hat{p} , $\hat{\tau}$ and $\hat{\phi}$ be the test functions corresponding to \mathbf{u} , p , τ and ϕ , respectively. For the momentum equations we adopt a simple pressure correction projection scheme to satisfy the incompressibility condition (3.1a). The time discretization form reads

$$\int_{\Omega} \left\{ \left[\rho(\phi_{n+1}) \left(\frac{\tilde{\mathbf{u}} - \mathbf{u}_n}{\Delta t} + \mathbf{u}_{n+1} \cdot \nabla \mathbf{u}_{n+1} - \mathbf{g} \right) - c_{n+1} \nabla \phi_{n+1} \right] \cdot \hat{\mathbf{u}} + \mathbf{T}_{n+1}(\phi_{n+1}) : \nabla \hat{\mathbf{u}} \right\} d\Omega = 0, \quad (3.3)$$

$$\int_{\Omega} (\nabla^2 \tilde{p}) \hat{p} d\Omega = \int_{\Omega} (\nabla \cdot \tilde{\mathbf{u}}) \hat{p} d\Omega, \quad (3.4)$$

$$\int_{\Omega} \mathbf{u}_{n+1} \cdot \hat{\mathbf{u}} d\Omega = \int_{\Omega} (\tilde{\mathbf{u}} - \nabla \tilde{p}) \cdot \hat{\mathbf{u}} d\Omega. \quad (3.5)$$

In (3.3), the inner product between two vectors \mathbf{A} and \mathbf{B} is denoted by $\mathbf{A} : \mathbf{B}$, and is given by

$$\mathbf{A} : \mathbf{B} = \text{trace}(\mathbf{B}^T \cdot \mathbf{A}).$$

In (3.5), $\tilde{\mathbf{u}}$ is the intermediate velocity which is not divergence free. We compute

$$\tilde{p} = \frac{\Delta t}{\rho(\phi_{n+1})} p$$

instead of p to avoid extra errors due to the varying time step Δt and different density values. A detailed overview of projection schemes and their error estimates can be found in [8]. For constitutive equations we use the following weak formulation

$$\int_{\Omega} \left\{ \tau + \lambda \tau^{\nabla} - 2\mu_p \mathbf{D} \right\} : (\hat{\tau} + \alpha \mathbf{u} \cdot \nabla \hat{\tau}) d\Omega = 0. \quad (3.6)$$

Note that we choose a different test space to improve the computation; which is as same as [22]. The Cahn-Hilliard equation is solved in two steps because C^0 elements are used in our computation. The time discretization reads

$$\int_{\Omega} \left\{ \frac{\phi_{n+1} - \phi_n}{\Delta t} + \mathbf{u}_{n+1} \cdot \nabla \phi_{n+1} - \gamma \nabla^2 c_{n+1} \right\} \hat{\phi} d\Omega = 0, \quad (3.7)$$

$$\int_{\Omega} c_{n+1} \hat{\phi} d\Omega = \int_{\Omega} k(\phi_{n+1}^3 - \phi_{n+1} - \epsilon^2 \nabla^2 \phi_{n+1}) \hat{\phi} d\Omega. \quad (3.8)$$

Finally, the whole system can be written in the following general form

$$\frac{\mathbf{U}_{n+1} - \mathbf{U}_n}{\Delta t} + \mathbf{F}(\mathbf{U}_{n+1}) = 0, \quad (3.9)$$

where \mathbf{U}_n and \mathbf{U}_{n+1} are the solution vector at step n and step $n+1$ levels respectively. Using standard iteration procedure, we let $\mathbf{U}_{n+1}^{(0)} = \mathbf{U}_n$ and solve

$$\frac{\mathbf{U}_{n+1}^{(k+1)} - \mathbf{U}_n}{\Delta t} + \mathbf{F}(\mathbf{U}_{n+1}^{(k)}) = 0, \quad (3.10)$$

repeatedly. In each iteration step, we pass the temporary solution $\mathbf{U}_{n+1}^{(k)}$ to \mathbf{F} in Eq. (3.10) and obtain the updated solution $\mathbf{U}_{n+1}^{(k+1)}$. These sub-steps are explicit except that all the terms involving the Laplacian operators are treated implicitly. Note that the whole system is implicit; so the block Gauss-Seidel iterations are used instead of solving a large linear system directly. The system is divided into six sub-systems and the algebraic multi-grid solver is applied to each of the sub-matrix blocks. In our computations, five to eight block-iterations are needed for each time step, which makes the iteration errors drop to below 10^{-10} .

4 Space and time adaptation

4.1 Moving mesh method for capturing interface

Solving the Cahn-Hilliard equation requires high mesh resolution around sharp interface, usually we need two or more elements within the interfacial region. But using a uniform

fine mesh will slow down the computation significantly. In addition, if we increase the mesh resolution globally without using local refinements, it is likely to cause instability. One way to handle this is to use coarse meshes with local refinement which can reduce computational costs [22]. Because of mesh structure modification, this method needs storage reallocation which is a fairly complex process. An alternative simpler approach is to use moving mesh methods which can concentrate the mesh nodes on sharp interface without changing mesh structure and the mesh size changes smoothly near the interfacial regions. In our computations, a moving mesh strategy based on the framework of Li et al. [11] is adopted. In [11], a mesh generation technique using harmonic mappings was developed. Apart from clustering more grid points near the interfaces, another benefit we wish to gain from using the moving mesh technique is the anisotropic mesh adaptation. It is hoped that more mesh nodes should be placed along the interface normal than that on the interface tangent.

The basic idea is to redistribute mesh nodes according to the regularity of solutions, which can be achieved by using harmonic mapping. Let \mathbf{x} and ξ be physical and logical mesh coordinates, respectively. The following Euler-Lagrange equation

$$\frac{\partial}{\partial x^i} \left(G^{ij} \frac{\partial \xi^k}{\partial x^j} \right) = 0 \quad (4.1)$$

gives a continuous and one-to-one mapping which is differentiable and has a non-zero Jacobian, where the inverse of $G = (G^{ij})$ is called monitor function which is in general dependent on the underlying solutions. The corresponding mesh energy is

$$E(\xi) = \sum_k \int_{\Omega} G^{ij} \frac{\partial \xi^k}{\partial x^i} \frac{\partial \xi^k}{\partial x^j}. \quad (4.2)$$

To make the moving mesh simulations effective, the key is to develop an efficient mesh redistribution procedure so that this part will cost as little as possible comparing with the solution evolution part. This issue was discussed in a recent paper of Di et al. [4].

In order to increase mesh resolution at interface we use the following function to realize the transition zone of the phase function:

$$g(\mathbf{x}) = |\nabla \phi(\mathbf{x})|^2. \quad (4.3)$$

Consequently, the corresponding monitor function is

$$m(\mathbf{x}) = \sqrt{1 + \alpha_1 g_s(\mathbf{x})}, \quad (4.4)$$

where g_s is a smoothed version of g and α_1 is a scaling parameter. It is pointed out that the smoothed function g_s is still of compact support, which is non-zero only within or near the interface because the phase function is constant away from the interface. This will make the mesh near the interface badly stretched. An example is shown in Fig. 1 which

comes from our first numerical example in Section 5. However, we still need regular elements with enough resolution for correctly computing the velocity and viscoelastic stress field away from the interface. One remedy is to smooth the function several times but this approach may be too costly. Instead, we solve a heat equation implicitly to diffuse the monitor function which is a much faster process due to the use of the algebraic multi-grid solvers. Let $g_d(\mathbf{x}) = g(\mathbf{x}, 1)$, where $g(\mathbf{x}, t)$ satisfies

$$\frac{\partial g}{\partial t} = \nu \nabla^2 g. \quad (4.5)$$

We define the new monitor function as

$$m(\mathbf{x}) = \sqrt{1 + \alpha_1 g_s(\mathbf{x}) + \alpha_2 g_d(\mathbf{x})}, \quad (4.6)$$

where $\alpha_2 > 0$ is a blend factor. This monitor function contains both g_s and g_d . On the one hand, we need the original high frequency monitor to control the local element size at the interface. On the other hand, using only g_d we may not have sufficiently small elements at the interface since g_d is very smooth. A similar idea was also used in [18]. The diffused monitor function with a few smooth steps produces more reasonable mesh distribution as demonstrated in Fig. 1. It is seen from Fig. 1 that the monitor (4.6) yields smaller ratio between the large and small element sizes which is found important for solving the multi-phase flow problems.

4.2 Adaptive time step

In moving mesh implementation, another important consideration is about adaptive time step due to the varying element size and velocity field. Time step must be determined based on physical and artificial parameters to avoid inaccuracy or instability so that we do not need to set time step manually. From the coupled equations described in Section 3, we consider the terms involving spatial derivatives. Roughly speaking, the following condition should be considered

$$\Delta t < \min \left\{ \Delta t_{\max}, C_1 \frac{h}{|u|}, C_2 \frac{h^4}{\gamma \sigma \epsilon} \right\}, \quad (4.7)$$

where C_1 and C_2 are constants, h is the local element size and u is the local velocity norm. It is obvious that the time step is constrained by the CFL condition and the Cahn-Hilliard equation. For correct phase-field computation, $h < \epsilon$ should be satisfied around interfacial regions and increasing ϵ/h_{\min} may produce more accurate results while slow down the overall computation. So we keep ϵ/h_{\min} being some constant to balance the accuracy and efficiency. A more detailed analysis of ϵ and h can be found in [9].

In our numerical computations, we choose $\Delta t_{\max} = 0.01$ in (4.7) in all of our numerical experiments.

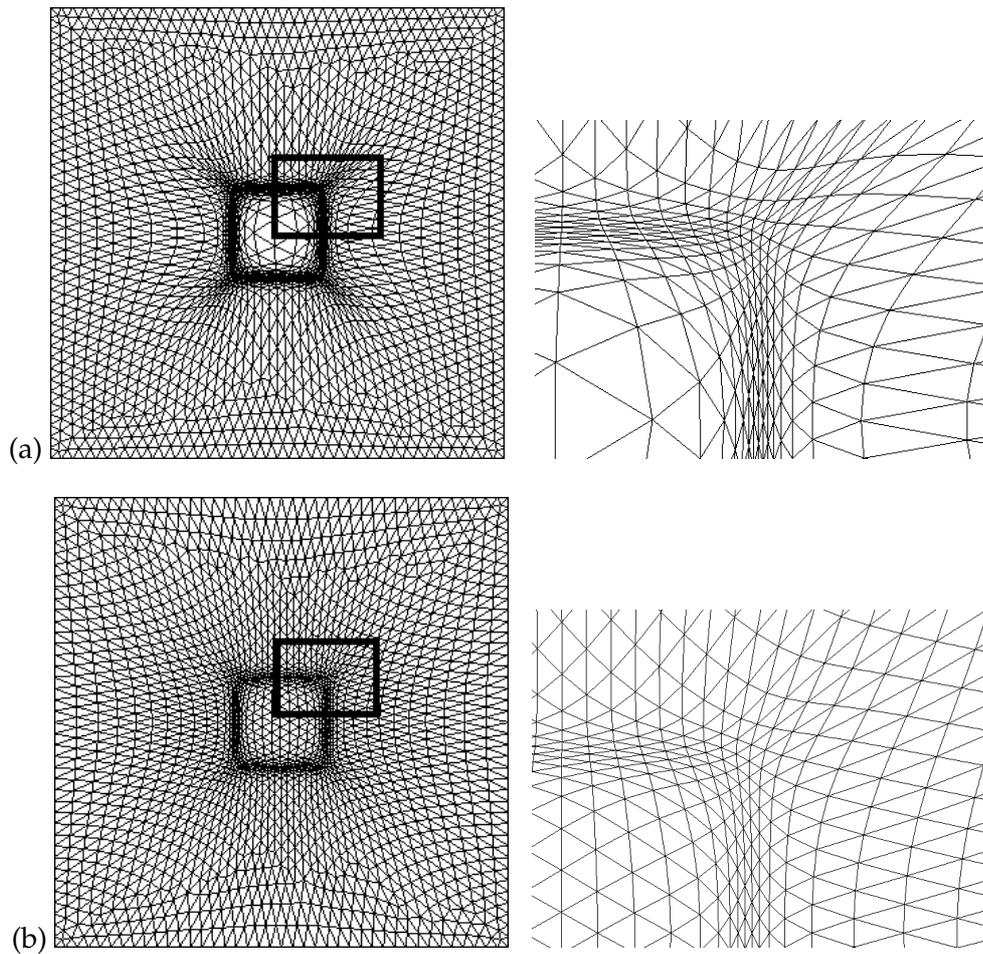


Figure 1: Sample moving meshes for the example in Section 5.1 obtained by using (a): the monitor function (4.4) and (b): the monitor function (4.6). The right gives the zoomed structure that is boxed on the left figure.

5 Numerical results

5.1 Vibrating of square bubbles

In our first experiment we consider the surface tension effects of a bubble by measuring its shape deformation history. A similar test of Newtonian flow is investigated in [12] without quantitative measurement of the bubble shape. At first a square bubble is placed in the center of domain shown as in Fig. 1 and the bubble begins to vibrate and its shape finally becomes round. The domain size is 1×1 and the bubble size is 0.2×0.2 . Both fluid component are non-Newtonian and the physical parameters are $\epsilon = 0.01, 0.005, 0.0025$, $\mu_{s1} = \mu_{p1} = \mu_{s2} = \mu_{p2} = 0.1$, $\lambda = 0.1$, $\gamma = 10^{-4}$ and $k = 0.1$.

We use both moving mesh and uniform mesh to validate the convergence. The shape

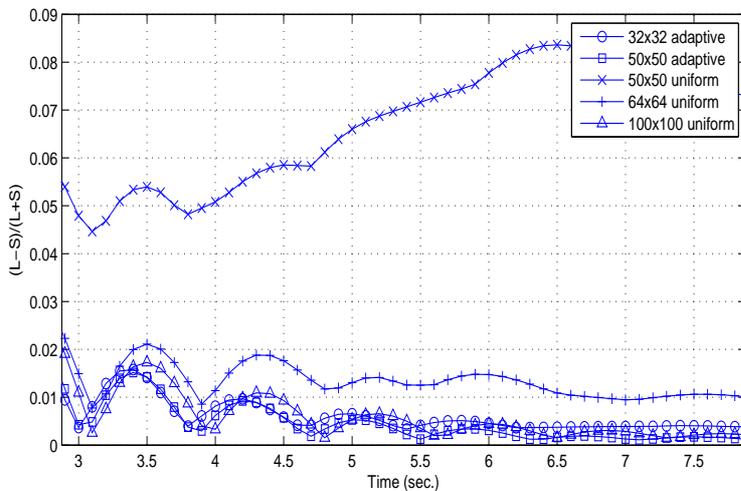


Figure 2: Example 5.1: transient deformation of bubble evolutions due to the surface tension effects. Both uniform and moving grids are used.

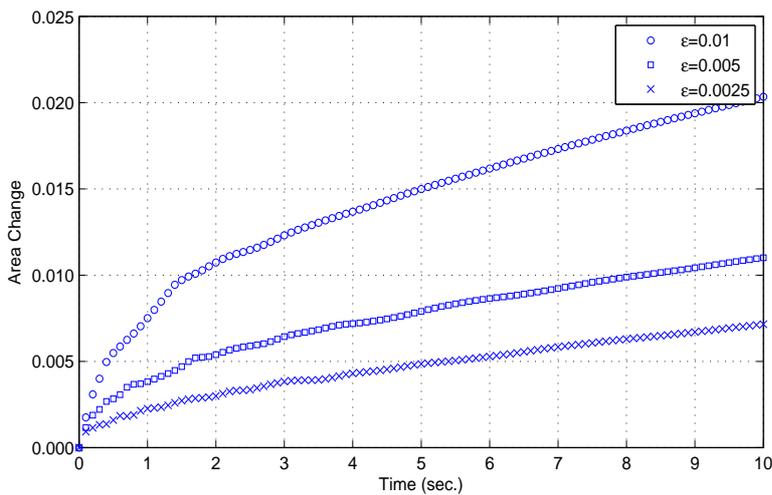


Figure 3: Example 5.1: bubble area change using different interfacial width.

deformation history

$$D = \frac{L - S}{L + S} \tag{5.1}$$

is plotted in Fig. 2, where L and S are the largest and smallest distance from bubble center to the interface. It is observed from Fig. 2 that the 32×32 and 50×50 moving mesh results and the 100×100 uniform mesh results are in good agreement, while 50×50 and 64×64 uniform meshes produce poor results. A performance statistics for this example is given

Table 1: Example 5.1: performance statistics.

Grid Size	Mesh nodes	Adaptive	CPU (sec.)
32×32	1259	Yes	7614
40×40	1932	Yes	11765
50×50	3009	Yes	18394
50×50	3009	No	12073
64×64	4880	No	20132
80×80	7579	No	32840
100×100	11833	No	51552

Table 2: Example 5.3: test results of rising bubbles.

Case	σ	μ_1	Newtonian matrix	U
A	0.1	0.562	Yes	0.090
B	0.1	0.562	No	0.086
C	0.1	0.0562	Yes	0.554
D	0.01	0.187	Yes	0.252
E	0.01	0.562	No	0.086

in Table 1, from which it is observed that a considerable CPU time is saved.

To reduce the area change ratio, defined by $(Area(t) - Area(0)) / Area(0)$, smaller ϵ may be chosen. Fig. 3 compares the area change ratio with different interfacial widths. It is observed from the figure that smaller interfacial widths preserves the area better.

5.2 Deformation of bubbles in shear flow

5.2.1 Deformation of Newtonian bubble

The deformation of a bubble in shear flow is also studied to validate our algorithm. The first test is Newtonian bubble in Newtonian matrix which is the same as that in [22] except the Reynolds number $Re = 1$ is used in our computation because the inertia in our computation is not ignored. The capillary number $Ca = \mu a \kappa / \sigma = 0.1$ is used, where a is drop radius and κ is shear rate. We set $\mu = \mu_{s1} = \mu_{s2}$, $\gamma = 10^{-4}$ and $\epsilon = 0.01$. The domain size is $16a \times 8a$ where the bubble is placed in the center of domain.

Again the shape deformation $D = (L - S) / (L + S)$ is considered, which is plotted in Fig. 4 where L and S are the longest and shortest distance from bubble center to interface, respectively. The deformation is also reported in [22] at $t = 4$ with several small values of ϵ ($\epsilon = 0.01, 0.005$ and 0.0025 vs. our value $\epsilon = 0.01$). It is shown that our results using moving mesh methods is in good agreement with those given in [22]. At $t = 4$ we obtained $D = 0.1081$ which is close to the values of $D = 0.1067, 0.1085, 0.1087$ (corresponds to $\epsilon = 0.01, 0.005$ and 0.0025 respectively) given in [22].

Fig. 5 plots the time-step history. When velocity is small, the time-step is almost constant since solving the Cahn-Hilliard equation requires small time-steps.

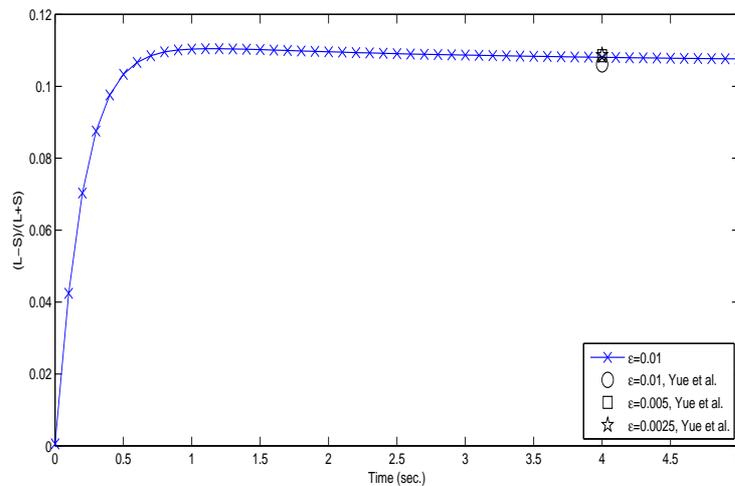


Figure 4: Example 5.2.1: Newtonian bubble deformation in Newtonian matrix (5960 mesh nodes are used with $\epsilon = 0.01$).

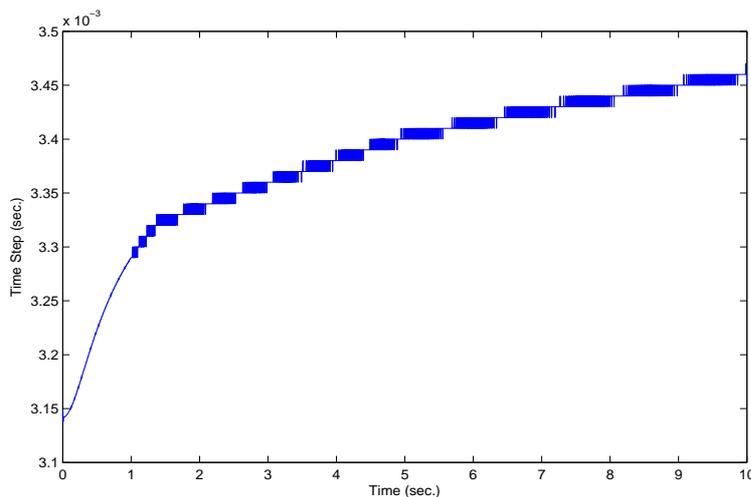


Figure 5: Example 5.2.1: adaptive time step history.

5.2.2 Deformation of viscoelastic bubble

The next test is a non-Newtonian bubble in a Newtonian matrix. We set $\gamma = 0.0016$ and $\epsilon = 0.04$ as in [21]. In this test we compared the shape deformation using different Deborah numbers $De = \lambda\kappa$. Fig. 6 plots the deformation $D = (L - S) / (L + S)$ for $De = 0.5, 1, 2$. It is observed that the non-Newtonian bubbles overshoot in D . The overshooting phenomenon agrees with the observations in [21] except that the curves of non-Newtonian bubbles are all above the Newtonian curve. This may be caused by the inertia of the bub-

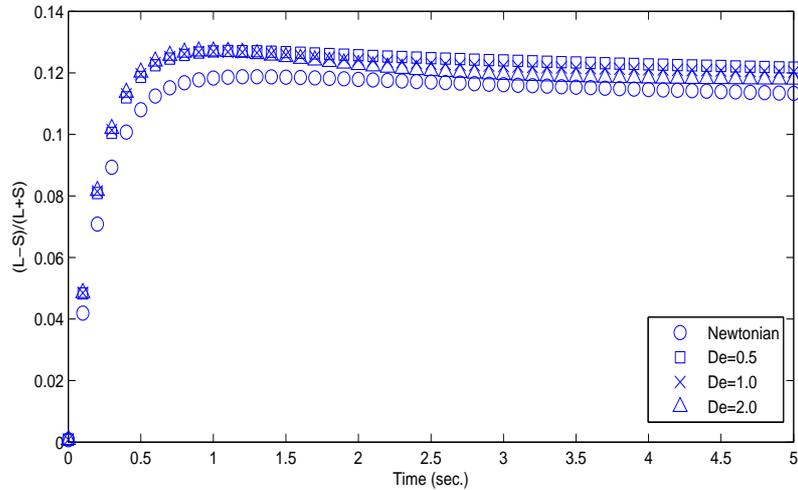


Figure 6: Example 5.2.2: deformation of viscoelastic bubbles in Newtonian matrix with different Deborah numbers (3841 mesh nodes are used with $\epsilon=0.04$)

ble which is not ignored in our computations. The non-Newtonian bubbles have smaller viscosity and move faster than the Newtonian bubble at the beginning; and the polymeric stress is damped due to the use of the coarse meshes.

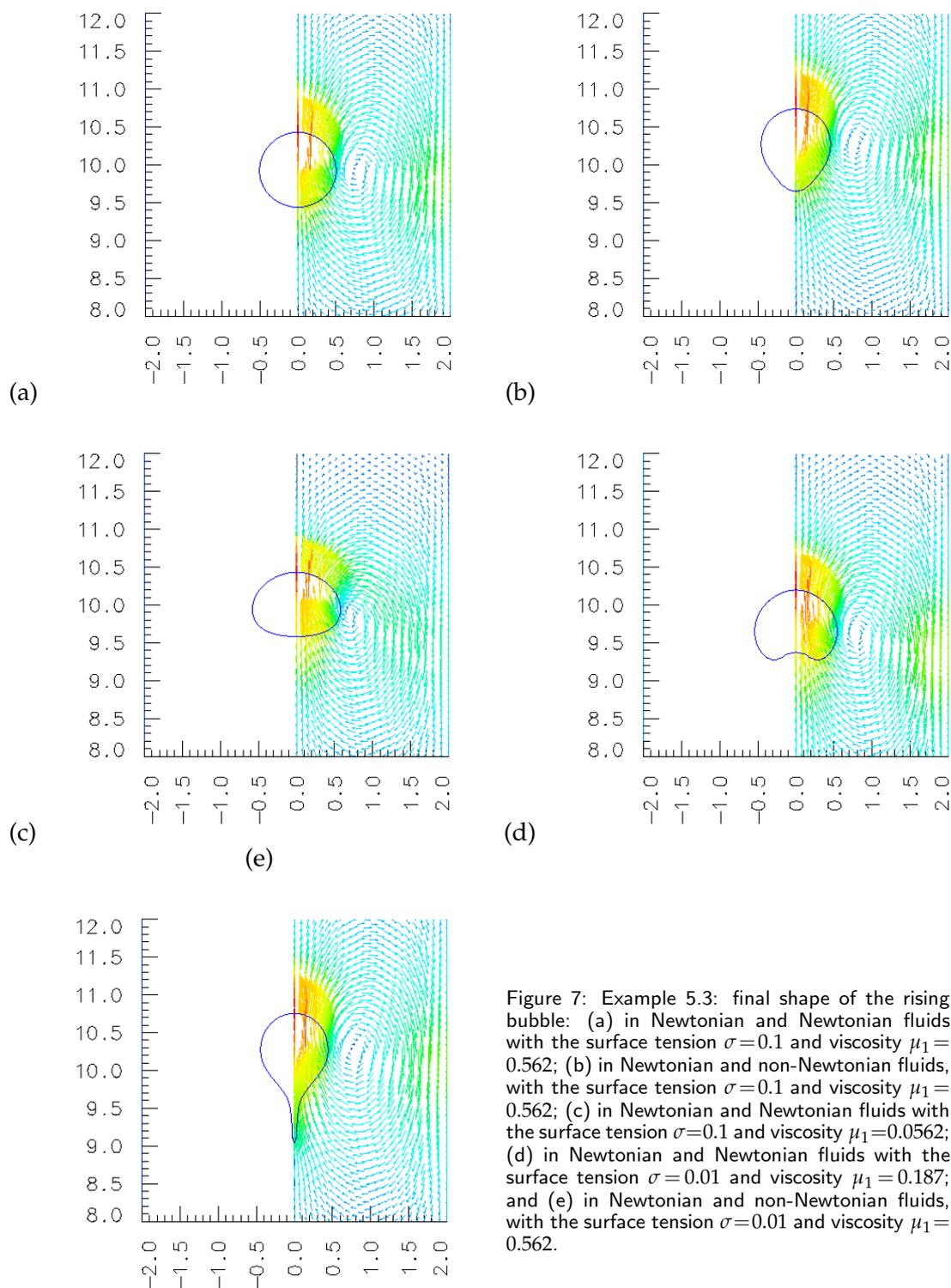
5.3 Rising of bubbles

We also compute the rising of Newtonian bubbles in Newtonian and non-Newtonian fluids. The configuration is same as that in [22]: a bubble with diameter $d=1$ is placed at $(0,2)$ in a 4×15 domain and we use a mesh with 6134 nodes for half of the domain. The fixed parameters are $\rho_1=0.5$, $\rho_2=1$, $g=2$ and the viscosity ratio $\mu_1/\mu_2=1$. If the matrix is non-Newtonian, we also set $\mu_p = \mu_s = 0.5\mu_2$. We measured the steady state velocity U of the rising bubbles and the results are listed in Table 2 where the parameters for 5 different cases are also listed.

Fig. 7 presents the corresponding bubble shapes for the tests (a)-(e) listed in Table 2, which correspond to $t=89.7s$, $98.2s$, $16.6s$, $34.7s$ and $101.3s$, respectively. It is observed that our moving mesh results are in good agreement with the cases a/b/d and the non-Newtonian case in [22]. The slight difference may be caused by the element size since the moving mesh method cannot control the smallest element size precisely.

It is noticed that both Cases (b) and (d) are non-Newtonian with the same μ_1 , but the surface tension is different: the surface in Case (e) is 10 times smaller than in Case (b). Consequently, it is observed that the trailing edge dragged by the viscoelastic stress at the bottom of bubble is much longer in Case (e).

Finally, we show the mesh distribution around the interface in Fig. 8. It is noticed that the element sizes away from the interface are not too small. This is in fact quite impor-



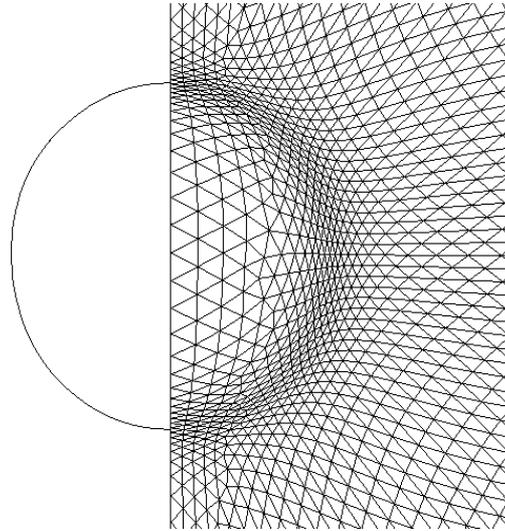


Figure 8: Example 5.3: mesh distribution around the interface of bubble.

tant in obtaining accurate numerical approximations. In fact, we are not only interested in resolving the phase function and its interface, but also in obtaining accurate flow fields which are governed by the nonlinear system (3.1a). As the moving mesh methods correspond to a continuous process, a reasonable resolution for the flow parameters in the whole solution domain is required. In our numerical computations, it is found that if the element sizes are too big even quite far away from the interface then the rising velocity and viscoelastic stress will be under estimated. In other words, if the flow fields are not well resolved then some physical details may be missing, which may have direct effect on the fluid velocity and the bubble shape.

6 Conclusion

In this work, we successfully applied the moving finite-element method for two-phase viscoelastic flow simulations. Since the mesh redistribution procedure normally requires to solve large size matrix equations, we decoupled the matrix equations to a much simpler block-tridiagonal type which can be efficiently solved by a particularly designed multi-grid method which is described in [10]. It is found that the moving mesh approach can lead to considerable savings in grid numbers and also the CPU time.

In our computations, it is observed that the balance between the largest and smallest element sizes should be given. In other words, we need to diffuse the monitor functions so that a certain portion of mesh nodes are distributed away from the interfaces. This will produce reasonably accurate velocity fields that in turn gives more accurate approximation to the phase functions.

In future works, we will study how the moving mesh interpolation affects the solution

of velocity field for unsteady problems. We will also extend the present 2D algorithms to 3D simulations.

Acknowledgments

The first and third authors would like to thank FRG grant of Hong Kong Baptist University and GIF grants of Hong Kong Research Grant Council. The research of the second author was supported by National Natural Science Foundation of China (No. 10801120). The research of the third author was supported by the Joint Research Fund for Hong Kong and Macau Young Scholars under the National Science Fund for Distinguished Young Scholars Scheme, National Natural Science Foundation of China.

References

- [1] D.M. Anderson, G.B. McFadden, and A.A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Reviews in Fluid Mechanics*, 30(1):139–165, 1998.
- [2] J.W. Cahn and J.E. Hilliard. Free energy of a nonuniform system. I. Interfacial free energy. *Journal of Chemical Physics*, 28:258–267, 1958.
- [3] J.W. Cahn and J.E. Hilliard. Free energy of a nonuniform system. III. Nucleation in a two-component Incompressible Fluid. *Journal of Chemical Physics*, 31:688–699, 1959.
- [4] Y. Di, R. Li, and T. Tang. A general moving mesh framework in 3d and its application for simulating the mixture of multi-phase flows. *Commun. Comput. Phys.*, 3:582–602, 2008.
- [5] Y. Di, R. Li, T. Tang, and P. Zhang. Moving mesh finite element methods for the incompressible navier-stokes equations. *SIAM J. Sci. Comput.*, 26(3):1036–1056, 2005.
- [6] Y. Di, R. Li, T. Tang, and P. Zhang. Level set calculations for incompressible two-phase flows on a dynamically adaptive grid. *J. Sci. Comput.*, 31(1-2):75–98, 2007.
- [7] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, and L.Q. Chen. Spectral implementation of an adaptive moving mesh method for phase-field equations. *Journal of Computational Physics*, 220(1):498–510, 2006.
- [8] JL Guermond, P. Mineev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6011–6045, 2006.
- [9] D. Jacqmin. Calculation of two-phase Navier-Stokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96–127, 1999.
- [10] R. Li, T. Tang, and P. Zhang. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *Journal of Computational Physics*, 177(2):365–393, 2002.
- [11] R. Li, P. Zhang, and T. Tang. Moving mesh methods in multiple dimensions based on harmonic maps. *Journal of Computational Physics*, 170(2):562–588, 2001.
- [12] 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: Nonlinear Phenomena*, 179(3-4):211–228, 2003.
- [13] J.G. Oldroyd. Non-Newtonian flow of liquids and solids. *Rheology: Theory and Applications*, 1:653–682, 1956.
- [14] E.S. Oran and J.P. Boris. *Numerical Simulation of Reactive Flow*. Cambridge University Press, 2001.

- [15] Z.-H. Qiao. Numerical investigations of the dynamical behaviors and instabilities for the gierer-meinhardt system. *Commun. Comput. Phys.*, 3:406–426, 2008.
- [16] J.A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35(1):341–372, 2003.
- [17] Y. Sun and C. Beckermann. Sharp interface tracking using the phase-field equation. *Journal of Computational Physics*, 220(1):626–653, 2007.
- [18] H. Wang, R. Li, and T. Tang. Efficient computation of dendritic growth with r -adaptive finite element methods. *Journal of Computational Physics*, 2008.
- [19] H.-Y. Wang and R. Li. Mesh sensitivity for numerical solutions of phase-field equations using r -adaptive finite element methods. *Commun. Comput. Phys.*, 3:357–375, 2008.
- [20] J.D. Yu, S. Sakai, and J.A. Sethian. Two-phase viscoelastic jetting. *Journal of Computational Physics*, 220(2):568–585, 2007.
- [21] P. Yue, J.J. Feng, C. Liu, and J. Shen. Transient drop deformation upon startup of shear in viscoelastic fluids. *Physics of Fluids*, 17:123101, 2005.
- [22] P. Yue, C. Zhou, J.J. Feng, C.F. Ollivier-Gooch, and H.H. Hu. Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing. *Journal of Computational Physics*, 219(1):47–67, 2006.