



An efficient moving mesh spectral method for the phase-field model of two-phase flows [☆]

Jie Shen ^{a,*}, Xiaofeng Yang ^b

^a Department of Mathematics, Purdue University, West Lafayette, IN 47907-1957, United States

^b Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3250, United States

ARTICLE INFO

Article history:

Received 30 July 2008

Received in revised form 31 December 2008

Accepted 6 January 2009

Available online 22 January 2009

Keywords:

Moving mesh

Two-phase flow

Spectral method

Phase-field

ABSTRACT

We develop in this paper a moving mesh spectral method for the phase-field model of two-phase flows with non-periodic boundary conditions. The method is based on a variational moving mesh PDE for the phase function, coupled with efficient semi-implicit treatments for advancing the mesh function, the phase function and the velocity and pressure in a decoupled manner. Ample numerical results are presented to demonstrate the accuracy and effectiveness of the moving mesh spectral method.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

The (diffusive) phase-field model, as an alternative to the sharp-interface model for free surface problems, has been successfully applied to describe meso-scale morphological pattern formations and interfacial motions in many material processes and two-phase flows (see, for instance, [1,2,19,7,18,26,25] and the references therein).

While the phase-field method has proven to be robust and versatile, it usually involves a high computational cost when a fixed grid method is used, for one needs to resolve the thin diffusive interfaces. In most cases, the interfacial regions occupy only a very small fraction of the total volume so most of the grid points in a fixed grid are not well utilized. This situation naturally calls for an adaptive procedure. Since the phase-field model consists of dynamic equations, it is more effective to use a moving mesh strategy than a local refinement one, we refer to [9] for a recent effort in this direction using a finite element approach in the physical domain. On the other hand, from the spectral method point of view, it appears that the only way to maintain spectral accuracy in an adaptive procedure seems to be finding a suitable *smooth* mapping which maps a function with large gradients to a smooth function. The moving mesh strategy provides a mean to determine such a mapping dynamically. Therefore, it is natural to develop a moving mesh spectral method for the phase-field model of two-phase flows in the computational domain (which is the mapped physical domain).

Although the moving mesh method has been well developed for finite difference and finite element methods (cf. [24,4,5,16,15,17] and the references therein), its applications to spectral method have been rather scarce, mainly because the transformed equation becomes highly complex and difficult to solve efficiently with a spectral method; and to a less extent, the numerical solution of the moving mesh partial differential equation (MMPDE) is not necessarily smooth so it is dif-

[☆] This work is partially supported by NSF DMS-0509665 and DMS-0610646.

* Corresponding author.

E-mail addresses: shen@math.purdue.edu (J. Shen), xfyang@email.unc.edu (X. Yang).

difficult to achieve high-accuracy even with a spectral method. In [11,12], the authors developed a semi-implicit Fourier-spectral moving mesh method for the Allen–Cahn and Cahn–Hilliard equations with periodic boundary conditions and showed that such a moving mesh method leads to significant savings as compared with the fixed grid Fourier-spectral method.

The purpose of this paper is to develop an efficient moving mesh spectral method for the phase-field model of two-phase flows with non-periodic boundary conditions. For the spatial discretization, we will use the fast spectral-Galerkin method for separable geometries developed in [20,22]. These methods are most efficient for solving elliptic equations with constant coefficients and with moderately varying coefficients. Therefore, a designing principle for the time discretization is to avoid, as much as possible, solving problems with non-constant coefficients at each time step, while allowing reasonably large time steps. This becomes a tricky demand as a coordinate transform renders all differential operators with constant coefficients to that with non-constant coefficients. We observe that our designing principle is very similar to that used in [11] which is however restricted to Allen–Cahn phase equations with periodic problems. The issues we address in this paper include:

- Identifying a suitable moving mesh strategy to determine the coordinate transform dynamically.
- Designing an efficient and stable time discretization scheme, for the mapped phase equation with a Lagrange multiplier, which does not suffer from the stiffness associated with the thin interfacial width and conserves the volume fraction well.
- Designing an efficient and stable time discretization scheme for the Navier–Stokes equations which does not involve the costly pressure solver.
- Combining the separate treatments for MMPDE, phase equation and Navier–Stokes equations together with the fast spectral-Galerkin method to form a robust and efficient moving mesh numerical scheme for two-phase flows, and investigate the effectiveness and accuracy compared with the fixed grid method.

The rest of the paper is organized as follows: In the next section, we first introduce the phase-field model for two-phase incompressible flows and review the framework for the MMPDE. Then, in Section 3, we describe in detail our numerical method for the coupled nonlinear system: our method consists of semi-implicit schemes for the MMPDE, the phase equation with Lagrangian multiplier and the Navier–Stokes equations based on a modified penalty formulation. In Section 4, we present several numerical results to illustrate the effectiveness and the correctness of our scheme by comparing with the usual fixed grid method (FGM). We conclude with a few remarks in the last section.

2. Phase-field model and moving mesh PDE

2.1. A phase-field model for two-phase incompressible flow

Let Ω be the physical domain filled with two incompressible fluids separated by a free moving interface. As in [18], we introduce a phase function $\phi(x)$, defined on the physical domain Ω , to label the two phases. Namely, we set $\phi(x) = 1$ in one phase and $\phi(x) = -1$ in the other phase with a thin smooth transitional layer across the interface. Then, the level set $\{x : \phi(x) = 0\}$ represents the interface. Introducing the elastic “mixing energy”:

$$W(\phi) = \int_{\Omega} \left[\frac{1}{2} |\nabla \phi|^2 + \frac{1}{4\eta^2} (\phi^2 - 1)^2 \right] dx, \tag{1}$$

where η is an artificial width for the “diffusive” interface, we determine the phase evolution through a gradient flow (cf. [18]):

$$\phi_t + u \cdot \nabla \phi = -\gamma \frac{\partial W}{\partial \phi} = \gamma \left(\Delta \phi - \frac{1}{\eta^2} \phi (\phi^2 - 1) \right), \tag{2}$$

where u is the velocity field of the flow, the parameter γ represents the elastic relaxation time scale of the fluid system. It can be expected that as $\gamma, \eta \rightarrow 0$, (2) converges to the classical two-phase fluid system. Therefore, the accuracy of the diffusive interface model (2) improves as γ, η decrease. However, the computational cost also increases as γ, η decrease. So how to choose γ, η to balance the accuracy and cost is a delicate manner. We refer to [10] for a more detailed discussion on this matter.

The phase Eq. (2) does not conserve volume fraction, so following [25] we introduce below a scalar function $\theta(t)$ which act as a Lagrange multiplier to enforce the conservation of the volume fraction:

$$\begin{aligned} \phi_t + u \cdot \nabla \phi &= \gamma \left(\Delta \phi - \frac{1}{\eta^2} \phi (\phi^2 - 1) + \theta(t) \right), \\ \frac{d}{dt} \int_{\Omega} \phi dx &= 0. \end{aligned} \tag{3}$$

The momentum equation for the flow, with a Boussinesq approximation for the variable density, takes the usual form:

$$\rho_0 (u_t + (u \cdot \nabla) u) = f - \nabla p + \nabla \cdot \sigma, \tag{4}$$

where ρ_0 is the background density (usually taken as the average of the two different densities of the two fluids), f includes the external body force and the gravity force due to the density difference in the two fluids, p is the pressure and σ is the deviatoric stress tensor which includes the viscous tensor and the induced elastic stress tensor. When we take into account the competition between the kinetic energy and the elastic energy, we find (cf. [18]):

$$\sigma = \mu(\phi)[\nabla u + (\nabla u)^t] - \lambda(\nabla\phi \otimes \nabla\phi), \tag{5}$$

where $\mu(\phi)$ is the dynamic viscosity coefficient, and the term $\nabla\phi \otimes \nabla\phi$ is the induced elastic stress due to the elastic mixing energy, and λ corresponds to the ratio between the kinetic energy and the elastic energy, and its ratio with η is proportional to the traditional surface tension energy (cf. [26]). A particular choice for $\mu(\phi)$ is the geometric average of μ_1 and μ_2 which are the dynamic viscosity of the two fluid phases, namely:

$$\frac{1}{\mu(\phi)} = \frac{1+\phi}{2\mu_1} + \frac{1-\phi}{2\mu_2}. \tag{6}$$

For more details on the phase-field model for two-phase flows, we refer to [18].

2.2. The MMPDE

Due to the nature of the phase function ϕ which has large gradients near the interfaces (cf. Fig. 1(a)), it is not efficient to approximate it in the physical domain Ω since a large number of unknowns are needed to resolve the interfaces. However, if we can find a mapping $x = x(\xi, t)$ such that the mapped function $\psi(\xi, t) := \phi(x(\xi, t), t)$ is smooth (cf. Fig. 1(b)) in the mapped domain Ω_c (the so called computational domain) with coordinate $\xi = (\xi_1, \dots, \xi_d)^t$, then, the mapped function $\psi(\xi, t)$ can be approximated efficiently in Ω_c . The question is how to find such a dynamic mapping $x = x(\xi, t)$.

Traditional moving mesh approaches [24] aim to find $x(\xi, t)$ which minimize the functional problem

$$I[x] = \int_{\Omega_c} \omega d\xi, \tag{7}$$

where ω is a suitable monitor function. In this paper, we shall follow the variational approach developed in [15,14] which aims to find $\xi = \xi(x, t)$, the inverse mapping of $x = x(\xi, t)$, that minimizes

$$E[\xi] = \int_{\Omega} \sum_{i=1}^d (\nabla_{\xi^i})^T G^{-1} \nabla_{\xi^i} dx, \tag{8}$$

where G is a suitable monitor function. Throughout the paper, we shall use $\nabla, \nabla \cdot$ and Δ to denote the gradient, divergence and Laplace operators in the original coordinates x , while $\nabla_{\xi}, \nabla_{\xi} \cdot$ and Δ_{ξ} will denote those operators in the transformed coordinates ξ .

The Euler–Lagrange equations associated with (8) are

$$\nabla \cdot (G^{-1} \nabla_{\xi^i}) = 0, \quad i = 1, \dots, d, \tag{9}$$

and the associated gradient flow are:

$$\frac{\partial}{\partial t} \xi^i(x, t) = m \nabla \cdot (G^{-1} \nabla_{\xi^i}), \quad i = 1, \dots, d, \tag{10}$$

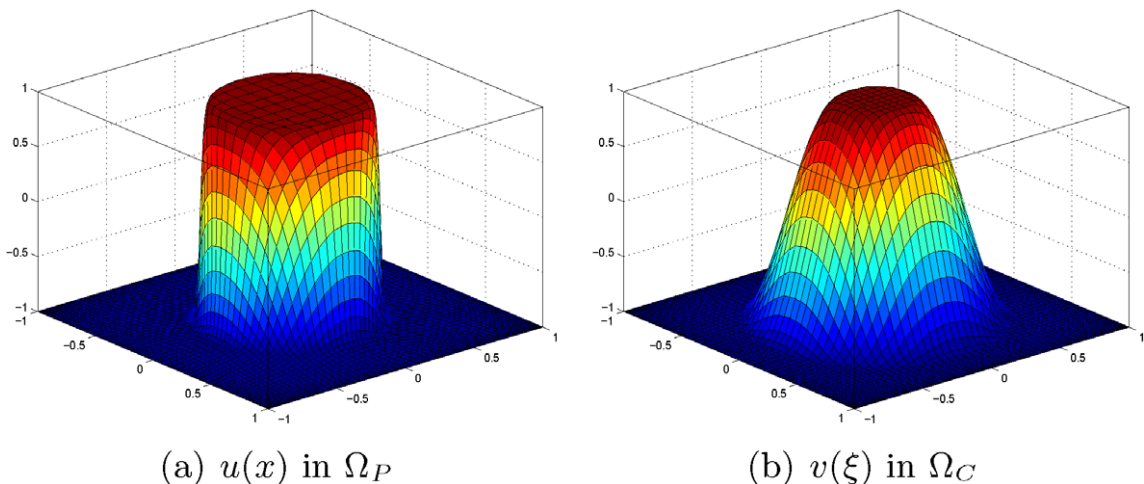


Fig. 1. Comparison of the representation of the same function in different coordinates.

where m is a suitable mobility function of our choice.

In general, the monitor function G should depend on the function ϕ , in particular, a common choice to use the Winslow's monitor function $G = wI$, where I is the identity matrix and w is a weight function. Following [15,14], we define the covariant and contravariant basis vectors

$$a_i = \frac{\partial x}{\partial \xi^i}, \quad a^i = \nabla \xi^i, \quad i = 1, \dots, d. \tag{11}$$

To fix the idea, we take $d = 3$. Then, the following relation holds:

$$a^i = \frac{1}{J} a_j \times a_k, \quad a_i = J a^j \times a^k, \quad a_i \cdot a^j = \delta_{ij} \quad (i, j, k) \text{ cyclic}, \tag{12}$$

where $J = a_1 \cdot (a_2 \times a_3)$ is the Jacobian of the coordinate transform. Thanks to (11) and (12), the gradient operator ∇ in the original coordinates can be expressed using only the covariant vector $\{a_i\}$:

$$\nabla = \sum_i a^i \frac{\partial}{\partial \xi^i} = \frac{1}{J} \sum_i \frac{\partial}{\partial \xi^i} J a^i = \frac{1}{J} \sum_i \frac{\partial}{\partial \xi^i} a_j \times a_k \quad (i, j, k) \text{ cyclic}. \tag{13}$$

Then, the divergence and Laplace operators $\nabla \cdot$ and $\Delta = \nabla \cdot \nabla$ in the original coordinates can be expressed using (13). In particular, we have

$$\Delta v = \nabla \cdot \nabla v = \frac{1}{J} \nabla_\xi \cdot (JA \nabla_\xi v), \tag{14}$$

where A is a positive definite matrix with entries $A_{ij} = a^i \cdot a^j$.

With the help of the above relations, the Eq. (10) with $G = wI$ is transformed to

$$\frac{\partial x}{\partial t} = \frac{m}{w^2} \sum_{i,j=1}^d a^i \cdot a^j \frac{\partial}{\partial \xi^i} \left(w \frac{\partial x}{\partial \xi^j} \right). \tag{15}$$

A popular choice for the monitor function is $w = \sqrt{1 + |\nabla \phi|^2}$ which is, however, not very convenient to use in the computational coordinates. Following the suggestion in [6] (see also [11]), we use

$$w = \sqrt{1 + \beta^2 |\nabla_\xi \phi(x(\xi, t), t)|^2}, \tag{16}$$

where β is a suitable scaling constant which can be used to adjust the mesh concentration. As for the mobility function m , we set $m = \tau w^2$ where τ is a parameter to control the relaxation time scale of (15).

2.3. The complete set of equations

We now write the governing equations in the new coordinates (ξ, t) through the transform $x = x(\xi, t)$. Notice that the time derivatives before and after the transform are related by

$$\partial_t \psi(\xi, t) = \partial_t \phi(x(\xi, t), t) = \dot{x} \cdot \nabla \phi + \partial_t \phi. \tag{17}$$

To simplify the notation, we shall use the same notations to denote the functions u, p, ϕ , etc. before and after the transformation. Then, the complete set of governing equations for the moving mesh phase-field model in the computational domain Ω_C are:

$$\rho_0(u_t - \dot{x} \cdot \nabla u + (u \cdot \nabla)u) = f - \nabla p + \nabla \cdot (\mu(\phi)[\nabla u + (\nabla u)^t] - \lambda(\nabla \phi \otimes \nabla \phi)), \tag{18a}$$

$$\nabla \cdot u = 0, \tag{18b}$$

$$\phi_t - \dot{x} \cdot \nabla \phi + u \cdot \nabla \phi = \gamma \left(\Delta \phi - \frac{1}{\eta^2} (\phi^2 - 1)\phi + \theta(t) \right), \tag{18c}$$

$$\frac{d}{dt} \int_{\Omega_C} \phi J d\xi = 0, \tag{18d}$$

$$\frac{\partial x}{\partial t} = \tau \sum_{i,j=1}^d (a^i \cdot a^j) \frac{\partial}{\partial \xi^i} \left(\sqrt{1 + \beta^2 |\nabla_\xi \phi|^2} \frac{\partial x}{\partial \xi^j} \right), \tag{18e}$$

where $\mu(\phi)$ is given by (6). Note that for the sake of simplicity, here and in the next section, we use two sets of differential operators (in the original and transformed coordinates) to describe the nonlinear system and its time discretizations. Since all computations will be performed in the transformed coordinates, we need to express the differential operators $\nabla, \nabla \cdot$ and Δ in the transformed coordinates through (13).

The above system is subjected to suitable initial and boundary conditions.

3. Description of the numerical scheme

The system (18e) is a coupled nonlinear system. The purpose of this section is to develop an efficient numerical scheme for (18e). Since we are mainly concerned with a spectral discretization in separable domains, our guiding principle is to avoid, as much as possible, solving problems with non-constant coefficients at each time step, leading to a simple, yet efficient and accurate numerical scheme for (18e). We shall describe our approaches for the MMPDE, phase equation and Navier–Stokes equations separately before we describe the complete time discretization scheme for (18e). Due to the complexity of the system, only first-order time discretization will be presented in detail, but a second-order version will also be presented and implemented.

Our numerical scheme is built upon the scheme presented in [25] for fixed grids. We refer to [25] for more background detail on the time discretization scheme below.

Notice that with the coordinate transform $x = x(\xi, t)$, all the differential operators $\nabla, \nabla \cdot$ and Δ in the original coordinates x depend on time when expressed in the transformed coordinates ξ . To simplify the notation, we shall omit the time dependence in the notation with the understanding that all operators $\nabla, \nabla \cdot$ and Δ in the schemes below are at the time t^{n+1} .

3.1. Time discretization of the MMPDE

Eq. (15) is a nonlinear diffusive equation. To avoid solving a nonlinear equation at each time step while allowing reasonable time step, we propose the following semi-discrete numerical algorithm

$$\begin{cases} \frac{x^{n+1} - x^n}{\Delta t} - \tau W^n \Delta_\xi (x^{n+1} - x^n) = \tau \sum_{ij} ((a^i)^n \cdot (a^j)^n) \frac{\partial}{\partial \xi^i} \left(W^n \frac{\partial x^n}{\partial \xi^j} \right), \\ (x^{n+1} - x^n)|_{\partial\Omega_C} = 0. \end{cases} \tag{19}$$

In the above, a^i and w are given by (11) and (16), W^n is a variable constant equal to the product of the maximum of w^n with the largest eigenvalue of the positive definite matrix A^n with entries $A_{ij}^n = (a^i)^n \cdot (a^j)^n$.

3.2. A stabilized and volume conserving time discretization for the phase equation

Let us consider first the Allen–Cahn phase equation without the transport term and the Lagrange multiplier. Following [25], we use the following stabilized semi-implicit scheme:

$$\begin{cases} \left(\frac{1}{\delta t} + \frac{\gamma S}{\eta^2} \right) (\phi^{n+1} - \phi^n) J^{n+1} - \gamma \kappa^{n+1} \Delta_\xi (\phi^{n+1} - \phi^n) = J^{n+1} \left(\dot{x}^n \cdot \nabla \phi^n + \gamma (\Delta \phi^n - \frac{1}{\eta^2} ((\phi^n)^2 - 1) \phi^n) \right), \\ \frac{\partial (\phi^{n+1} - \phi^n)}{\partial n} \Big|_{\partial\Omega_C} = 0, \end{cases} \tag{20}$$

where J^{n+1} are the Jacobian at time step $n + 1$, and κ^{n+1} can be taken as the maximum eigenvalue of the matrix $J^{n+1} A^{n+1}$. We recall that the ∇ and Δ operators in the above should be computed via (13) and (14) at time t^{n+1} . The same is assumed in all the schemes presented below.

Notice that in addition to the artificial diffusion term, we have also added an artificial stabilizing term $\frac{\gamma S}{\eta^2} (\phi^{n+1} J^{n+1} - \phi^n J^n)$ to alleviate the stiffness associated with small η (cf. [25]). The stabilizing parameter s is usually taken to be $O(1)$.

The above scheme does not conserve the volume fraction. So a Lagrange multiplier is introduced in [25] to enforce the volume conservation. While this process is straightforward without coordinate transform, extra care has to be taken when a coordinate transform is involved.

With the coordinate transform, it is important to conserve the volume fraction in the original coordinate x , i.e.

$$\int_{\Omega_C} (\phi^{n+1} J^{n+1} - \phi^n J^n) d\xi = 0. \tag{21}$$

To this end, we propose the following scheme which is of first-order in time:

$$\left(\frac{1}{\delta t} + \frac{\gamma S}{\eta^2} \right) (\phi^{n+1} - \phi^n) J^{n+1} - \gamma \kappa^{n+1} \Delta_\xi (\phi^{n+1} - \phi^n) - \gamma \theta^{n+1} = J^{n+1} \left(-u^n \cdot \nabla \phi^n + \dot{x}^n \cdot \nabla \phi^n + \gamma \left(\Delta \phi^n - \frac{1}{\eta^2} ((\phi^n)^2 - 1) \phi^n \right) \right), \tag{22a}$$

$$\int_{\Omega_C} (\phi^{n+1} J^{n+1} - \phi^n J^n) d\xi = 0, \tag{22b}$$

$$\frac{\partial (\phi^{n+1} - \phi^n)}{\partial n} \Big|_{\partial\Omega_C} = 0, \tag{22c}$$

where

$$\dot{x}^n := \frac{x^{n+1} - x^n}{\delta t}. \tag{23}$$

We denote $\alpha = \frac{1}{\delta t} + \frac{\gamma S}{\eta^2}$ and

$$G^n = -u^n \cdot \nabla \phi^n + \dot{x}^n \cdot \nabla \phi^n + \gamma \left(\Delta \phi^n - \frac{1}{\eta^2} ((\phi^n)^2 - 1) \phi^n \right). \tag{24}$$

Then, integrating (22a) over Ω_C and using the constraint (22b), we find that θ^{n+1} can be explicitly obtained by

$$\gamma |\Omega_C| \theta^{n+1} = \alpha \int_{\Omega_C} (J^n - J^{n+1}) \phi^n d\xi - \int_{\Omega_C} G^n J^{n+1} d\xi, \tag{25}$$

where $|\Omega_C|$ denotes the volume of Ω_C . With θ^{n+1} known, we can then determine $\phi^{n+1} - \phi^n$ by solving a Poisson-type equation with a variable coefficient αJ^{n+1} . This equation can be solved efficiently by using a preconditioned conjugate gradient method.

A second-order version of (22c) is described below:

$$\begin{aligned} & \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\delta t} J^{n+1} + \frac{\gamma S}{\eta^2} (\phi^{n+1} - 2\phi^n + \phi^{n-1}) J^{n+1} - \gamma \kappa^{n+1} \Delta_\xi (\phi^{n+1} - 2\phi^n + \phi^{n-1}) - \gamma \theta^{n+1} \\ & = 2J^{n+1} \left(-u^n \cdot \nabla \phi^n + \gamma \left(\Delta \phi^n - \frac{1}{\eta^2} ((\phi^n)^2 - 1) \phi^n \right) \right) \\ & \quad - J^{n+1} \left(-u^{n-1} \cdot \nabla \phi^{n-1} + \gamma \left(\Delta \phi^{n-1} - \frac{1}{\eta^2} ((\phi^{n-1})^2 - 1) \phi^{n-1} \right) \right) + J^{n+1} (\dot{x}^{n+1} \cdot \nabla (2\phi^n - \phi^{n-1})) \end{aligned} \tag{26a}$$

$$\int_{\Omega_C} (\phi^{n+1} J^{n+1} - \phi^n J^n) d\xi = 0, \tag{26b}$$

$$\left. \frac{\partial \phi^{n+1}}{\partial n} \right|_{\partial \Omega_C} = 0, \tag{26c}$$

where

$$\dot{x}^{n+1} = \frac{3x^{n+1} - 4x^n + x^{n-1}}{2\delta t}, \tag{27}$$

and θ^{n+1} can be determined using (26b) by a relation similar to (25).

3.3. A modified penalty method for the time-dependent Stokes equations

In order to decouple the computation of the pressure from velocity, a common strategy is to use a projection type scheme (see, for instance, a recent review in [13]) for the Navier–Stokes equations. However, such a scheme involves solving a pressure Poisson equation, which, after the mapping, becomes an elliptic equation with variable coefficients. Unlike the MMPDE and the phase equation, the pressure Poisson equation is not “dynamic” so the previous trick used for the MMPDE in (19) cannot be applied here and one has to resort to an iterative method for solving the pressure Poisson equation. It is widely recognized that, in a projection type method, except when fast Poisson solver is available, solving the pressure equation is the most challenging and time consuming part. In order to avoid the costly procedure for solving the pressure equation, we propose a different approach.

We start with a modified first-order penalty method for the time dependent Stokes equations with homogeneous Dirichlet boundary conditions (cf. [21]):

$$\begin{cases} \frac{u^{n+1} - u^n}{\delta t} - \nu \Delta u^{n+1} + \nabla p^{n+1} = f^n, & u^{n+1}|_{\partial \Omega} = 0, \\ \nabla \cdot u^{n+1} + \varepsilon (p^{n+1} - p^n) = 0. \end{cases} \tag{28}$$

It can be easily shown that the above scheme is unconditionally stable and the error for both the velocity and pressure behave like $O(\delta t)$ and the divergence of u like $O(\varepsilon \delta t)$.

Substituting

$$p^{n+1} = p^n - \frac{1}{\varepsilon} \nabla \cdot u^{n+1} \tag{29}$$

into the first equation in (28), we obtain

$$\frac{u^{n+1} - u^n}{\delta t} - \left(\nu + \frac{1}{\varepsilon} \right) \Delta u^{n+1} - \frac{1}{\varepsilon} \nabla \times \nabla \times u^{n+1} = f^n - \nabla p^n. \tag{30}$$

The above equation is elliptic but couples all components of u . In order to use the fast Poisson solver, we replace $\frac{1}{\varepsilon} \nabla \times \nabla \times u^{n+1}$ by $\frac{1}{\varepsilon} \nabla \times \nabla \times u^n$, leading to the decoupled system:

$$\frac{u^{n+1} - u^n}{\delta t} - \left(v + \frac{1}{\varepsilon} \right) \Delta u^{n+1} = f^n + \frac{1}{\varepsilon} \nabla \times \nabla \times u^n - \nabla p^n, \quad u^{n+1}|_{\partial\Omega} = 0, \tag{31a}$$

$$p^{n+1} = p^n - \frac{1}{\varepsilon} \nabla \cdot u^{n+1}. \tag{31b}$$

It is clear that this substitution introduces an extra error of $O(\delta t/\varepsilon)$. Hence to balance the various truncation errors, one should choose $\varepsilon = O(1)$, resulting an error of order δt .

A remarkable property of the scheme (31b) is that it is unconditionally stable.

Lemma 3.1. *Let (u^n, p^n) be the solution of (31b). Then for all $m \geq 0$, we have*

$$\|u^{m+1}\|_{L^2}^2 + \varepsilon \delta t \|p^{m+1}\|_{L^2}^2 + \delta t \left(v + \frac{1}{\varepsilon} \right) \sum_{n=0}^m \|\nabla u^{n+1}\|_{L^2}^2 \leq C \frac{\delta t}{v} \sum_{n=0}^m \|f^n\|_{H^{-1}}^2 + \|u^0\|_{L^2}^2 + \varepsilon \delta t \|p^0\|_{L^2}^2.$$

Proof. Take the inner product of (31a) with $2\delta t u^{n+1}$, integrating by parts and using the Cauchy–Schwarz inequality, we find

$$\begin{aligned} & \|u^{n+1}\|_{L^2}^2 - \|u^n\|_{L^2}^2 + \|u^{n+1} - u^n\|_{L^2}^2 + 2\delta t \left(v + \frac{1}{\varepsilon} \right) \|\nabla u^{n+1}\|_{L^2}^2 - 2\delta t (p^n, \nabla \cdot u^{n+1}) \\ & \leq \frac{\delta t}{\varepsilon} (\|\nabla \times u^{n+1}\|_{L^2}^2 + \|\nabla \times u^n\|_{L^2}^2) + v \delta t \|\nabla u^{n+1}\|_{L^2}^2 + \frac{C \delta t}{v} \|f^n\|_{H^{-1}}^2. \end{aligned}$$

Using (31b), we obtain

$$\begin{aligned} -2\delta t (p^n, \nabla \cdot u^{n+1}) &= 2\varepsilon \delta t (p^n, p^{n+1} - p^n) = \varepsilon \delta t (\|p^{n+1}\|_{L^2}^2 - \|p^n\|_{L^2}^2 - \|p^{n+1} - p^n\|_{L^2}^2) \\ &= \varepsilon \delta t (\|p^{n+1}\|_{L^2}^2 - \|p^n\|_{L^2}^2) - \frac{\delta t}{\varepsilon} \|\nabla \cdot u^{n+1}\|_{L^2}^2. \end{aligned}$$

Combining the above two relations and Using the identity

$$\|\nabla u\|_{L^2}^2 = \|\nabla \times u\|_{L^2}^2 + \|\nabla \cdot u\|_{L^2}^2, \quad \forall u \in H_0^1(\Omega)^d \quad (d = 2, 3),$$

we find

$$\|u^{n+1}\|_{L^2}^2 - \|u^n\|_{L^2}^2 + \|u^{n+1} - u^n\|_{L^2}^2 + \varepsilon \delta t (\|p^{n+1}\|_{L^2}^2 - \|p^n\|_{L^2}^2) + \delta t \left(v + \frac{1}{\varepsilon} \right) \|\nabla u^{n+1}\|_{L^2}^2 \leq \frac{\delta t}{\varepsilon} \|\nabla \times u^n\|_{L^2}^2 + \frac{C \delta t}{v} \|f^n\|_{H^{-1}}^2.$$

We conclude by summing up the above relation for $n = 0, \dots, m$. \square

Remark 3.1. We believe that the scheme (31b) is new and interesting as it appears to be the only unconditionally stable scheme for time-dependent Stokes equations that only needs to solve a decoupled Poisson-type equation for each of the velocity components. However, it is difficult to design a stable, second-order version of this scheme since a higher-order extrapolation for $\frac{1}{\varepsilon} \nabla \times \nabla \times u^{n+1}$ would render the scheme unstable.

When we apply the scheme to nonlinear Navier–Stokes equations, an explicit treatment of the nonlinear term will require that the time step be sufficiently small, but can usually be much larger than the standard CFL condition allows thanks to the implicit treatment of the diffusion term.

The above scheme needs to be modified to more efficiently deal with the variable viscosity case where $v \Delta u^{n+1}$ is replaced by $\nabla \cdot \mu(\phi^{n+1})[\nabla u^{n+1} + (\nabla u^{n+1})^t]$. To this end, we set $\bar{\mu} = \frac{\mu_1 + \mu_2}{2}$ and consider the scheme:

$$\frac{u^{n+1} - u^n}{\delta t} - \left(\bar{\mu} + \frac{1}{\varepsilon} \right) \Delta (u^{n+1} - u^n) = f^n - \nabla p^n \nabla \cdot \left(\mu(\phi^{n+1})[\nabla u^n + (\nabla u^n)^t] + \frac{1}{\varepsilon} \nabla u^n \right) + \frac{1}{\varepsilon} \nabla \times \nabla \times u^n, \tag{32a}$$

$$p^{n+1} = p^n - \frac{1}{\varepsilon} \nabla \cdot u^{n+1}. \tag{32b}$$

Finally, after taking into consideration of the coordinate transform, a first-order scheme for (18a) and (18b) reads:

$$\begin{aligned} \rho_0 \frac{u^{n+1} - u^n}{\delta t} - \kappa^{n+1} \left(\bar{\mu} + \frac{1}{\varepsilon} \right) \mathcal{A}_\varepsilon (u^{n+1} - u^n) &= \rho_0 (\dot{x}^n \cdot \nabla u^n + u^n \cdot \nabla u^n) + \frac{1}{\varepsilon} \nabla \times \nabla \times u^n + f^n - \nabla p^n + \nabla \\ &\cdot \left(\mu(\phi^{n+1})[\nabla u^n + (\nabla u^n)^t] + \frac{1}{\varepsilon} \nabla u^n - \lambda \nabla \phi^{n+1} \otimes \nabla \phi^{n+1} \right), \end{aligned} \tag{33a}$$

$$p^{n+1} = p^n - \frac{1}{\varepsilon} \nabla \cdot u^{n+1}, \tag{33b}$$

where κ^{n+1} is the same constant as in (22a).

It appears difficult to design a stable second-order version of the scheme (31b). It is clear that a second-order version of (30) can be obtained by replacing $\frac{u^{n+1}-u^n}{\delta t}$ in (30) with $\frac{3u^{n+1}-4u^n+u^{n-1}}{2\delta t}$, and replacing again $\frac{1}{\epsilon} \nabla \times \nabla \times u^{n+1}$ by $\frac{1}{\epsilon} \nabla \times \nabla \times (2u^n - u^{n-1})$. Unfortunately, this scheme becomes unstable due to the second-order extrapolation. It is however possible to design a second-order scheme by using one of the second-order projection type schemes (cf. [13]) if one is willing to solve a second-order equation for the pressure with variable coefficients at each time step. This option would become particularly competitive if the space variables are discretized via a finite element method. We list below such a scheme (18a) and (18b) based on the second-order standard pressure-correction scheme (cf. [13]):

$$\begin{aligned} \rho_0 \frac{3\tilde{u}^{n+1} - 4u^n + 2u^{n-1}}{2\delta t} - \nabla \cdot \mu(\phi^{n+1}) \nabla \tilde{u}^{n+1} \\ = \rho_0 (\dot{x}^{n+1} \cdot \nabla (2u^n - u^{n-1}) + 2u^n \cdot \nabla u^n - u^{n-1} \cdot \nabla u^{n-1}) + f^{n+1} - \nabla p^n - \lambda \nabla \cdot (\nabla \phi^{n+1} \otimes \nabla \phi^{n+1}), \end{aligned} \tag{34a}$$

$$\Delta(p^{n+1} - p^n) = \frac{3}{2\delta t} \nabla \cdot \tilde{u}^{n+1}, \quad \left. \frac{\partial(p^{n+1} - p^n)}{\partial n} \right|_{\partial\Omega_c} = 0, \tag{34b}$$

$$u^{n+1} = \tilde{u}^{n+1} - \frac{3\delta t}{2} \nabla(p^{n+1} - p^n), \tag{34c}$$

where \dot{x}^{n+1} is given by (27).

Note that the above system, when written in the transformed coordinates, involves solving a sequence of second-order elliptic equations with variable coefficients for \tilde{u}^{n+1} and for $p^{n+1} - p^n$.

3.4. The complete time discretization for the coupled system

We are now ready to describe the complete first-order scheme for (18e).

Given $(u^n, p^n, \phi^n, \theta^n)$, we update $(u^{n+1}, p^{n+1}, \phi^{n+1}, \theta^{n+1})$ as follows:

- (i) Find the new mapping x^{n+1} by solving (19).
- (ii) Find θ^{n+1} from (25), and then find ϕ^{n+1} by solving (22c).
- (iii) Find u^{n+1} and p^{n+1} from (33a).

Remark 3.2. A few remarks are in order:

- While the initial conditions for u and ϕ are given for any specific application, we need to find a suitable initial condition for x , namely, an initial transform which maps the initial phase function with large gradient to a smooth function. This can be achieved by finding the steady state solution $x_0(\xi)$ of (15) with $w = \sqrt{1 + \beta^2 |\nabla_\xi \phi(x, 0)|^2}$.
- At each time step, we have to solve a Poisson-type equation (with constant coefficients) for x^{n+1} and for u^{n+1} , and a Poisson-type equation with variable coefficients for ϕ^{n+1} .
- In the above scheme, the operators in the original coordinates are all applied to quantities from previous time steps and can be computed from (13).
- The above procedure leads to first-order accuracy for all concerned quantities. To improve the accuracy, we can replace (22c) by the second-order scheme (26c), and replace (33a) by the second-order scheme (34). Note that there is no need to use a higher-order scheme for the MMPDE (18e) as its purpose is only to provide a suitable mapping.

3.5. Spatial discretization

We now briefly describe the spatial discretization used in this paper, although the time discretization scheme described above is suitable for other spatial discretizations as well.

We consider in this paper the spectral-Galerkin methods developed in [20,22]. These methods are extremely efficient and accurate for solving Poisson type equations in simple separable geometries, so they are very suitable for the time discretization scheme described above. Several remarks are in order:

- Since a modified penalty method is used for the Navier–Stokes part of the system, we need to use a pair of *inf-sup* compatible spaces for the velocity and pressure, a popular choice is $P_N \times P_{N-2}$ (cf. [3]) for the spectral discretization.
- The integrals in (25) can be computed simply and exactly. In fact, let $J_0^n, J_0^{n+1}, \phi_0^n$ and G_0^n be the coefficients correspond to the constant terms in the Legendre expansions of J^n, J^{n+1}, ϕ^n and G^n . Then, we have

$$\gamma |\Omega_c| \theta^{n+1} = \alpha \int_{\Omega_c} (J^n - J^{n+1}) \phi^n d\xi - \int_{\Omega_c} G^n J^{n+1} d\xi = |\Omega_c| (\alpha (J_0^n - J_0^{n+1}) \phi_0^n - G_0^n J_0^{n+1}). \tag{35}$$

By using the above formula, the constraint (22b) is satisfied exactly.

- Besides the Poisson-type equations for the velocity and (19) for the mapping function, one also has to solve at each time step an elliptic equation for the phase function in the form

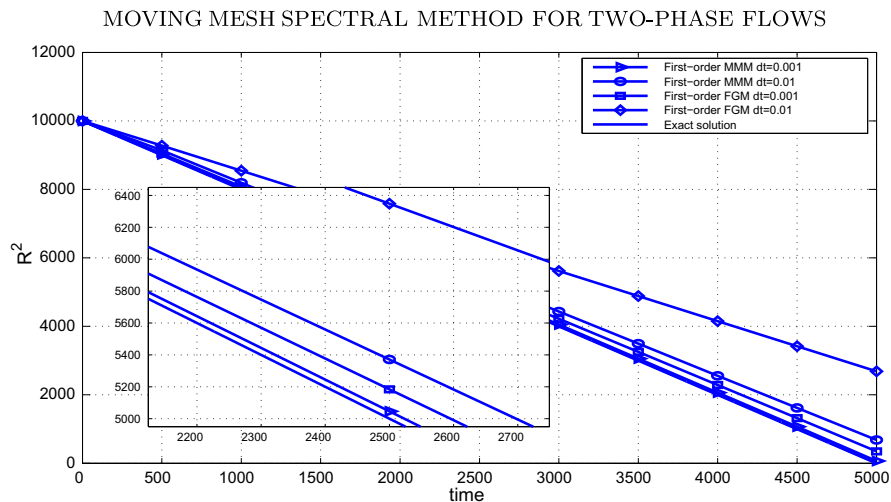
$$J\phi - \gamma \Delta_\xi \phi = g, \quad \frac{\partial \phi}{\partial n} \Big|_{\partial \Omega_c} = 0, \tag{36}$$

where J is the Jacobian of the transform. This equation can be efficiently solved by a preconditioned conjugate gradient iteration using a constant coefficient problem as the preconditioner.

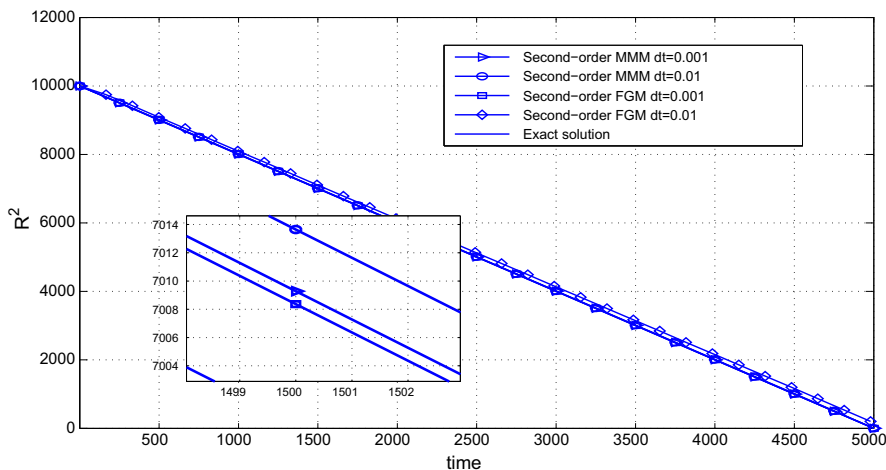
- The mapping determined from (19) is usually not very smooth in the sense that the last coefficients of its spectral expansion may not be sufficiently small. Thus, the mapping should be *smoothed* by applying a suitable filter before it is used. The choice of filter does not affect the accuracy of the scheme as the sole purpose of (15) is to provide a suitable mapping. In our computation, the raised cosine filter [23] is used.

4. Numerical simulations

In this section, we compare the numerical results obtained from our moving mesh (spectral) method (MMM) and the usual fixed grid (spectral) method (FGM) to demonstrate the effectiveness of our MMM.



(a) First-order MMM with grid size 65^2 and FGM with grid size 257^2



(b) Second-order MMM with grid size 65^2 and FGM with grid size 257^2

Fig. 2. Accuracy test.

In all examples below, the physical domain and computational domain are fixed to be $(-1, 1)^d$ ($d = 2, 3$). For all computations, we used the following computational parameters: $s = 1, \beta = 1, \tau = 1$ and $\epsilon = 1$. Other parameters are specified in the figure captions.

4.1. Allen–Cahn phase equation

We first examine the accuracy and effectiveness of the moving mesh spectral method for the Allen–Cahn phase Eq. (3) (with $u = 0$).

4.1.1. Shrinkage of a circular domain

Here, we compare the accuracy of the MMM with the FGM for the classical benchmark problem described in [8]. At the initial state, there is a circular interface boundary with a radius of $R_0 = 100$ in the rectangular domain of $[0, 256] \times [0, 256]$. Such a circular interface governed by the Allen–Cahn phase equation (with thickness $\eta = 1$) will shrink and eventually disappear. It can be shown, as the interfacial thickness goes to zero, that the velocity of the moving interface V is given by $V = \frac{dR}{dt} = -\frac{1}{R}$ where R is the radius of the circle at a given time t . We then derive that $R^2 = R_0^2 - 2t$. After we map the domain to $[-1, 1] \times [-1, 1]$, we obtain the Eq. (2) with $\gamma = 6.10351 \times 10^{-5}$ and $\eta = 0.0078$. The initial smooth phase profile is given by $\phi = -\tanh\left(\frac{\sqrt{x^2+y^2}-r_0}{\eta}\right)$ with $r_0 = 100/128$.

In Fig. 2, we plot the approximation to $R^2(t)$ obtained from the first- and second-order MMM and FGM. It can be seen that the MMM with grid size 65^2 provides better accuracy than the FGM with grid size 257^2 for the same time step. The evolution

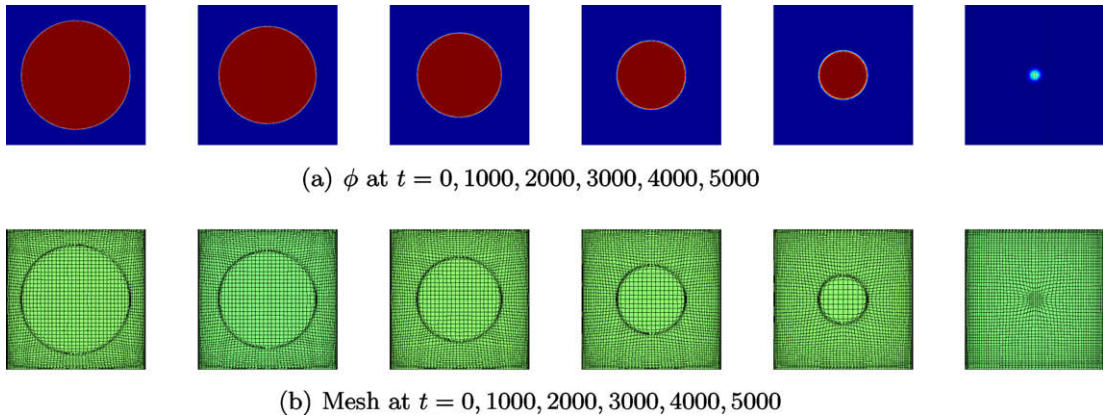


Fig. 3. The evolution of the circular domain of second-order MMM with grid size 65^2 .

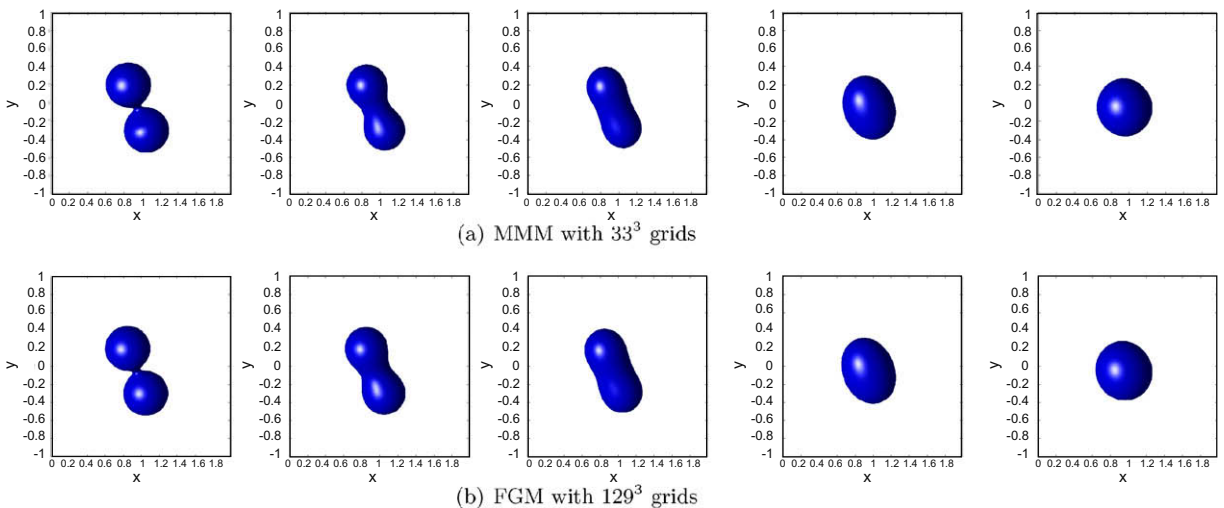


Fig. 4. Coalesce of two kissing ball: $\delta t = 0.001, \gamma = 0.02, \eta = 0.04$ at $t = 0, 0.5, 1, 3, 5$.

of the circular domain and the corresponding mesh generated by MMM at different times are shown in Fig. 3. One observes from Fig. 3(b) that the meshes generated by MMM are well concentrated near the interface.

4.1.2. Coalescence of two kissing balls in 3D

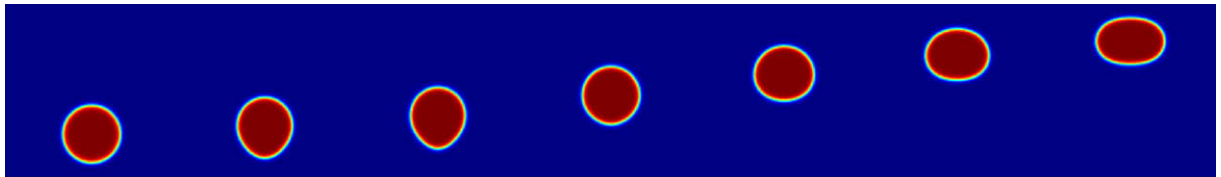
We now consider the coalescence of two kissing balls in 3D. The governing equation is the phase Eq. (3) (with $u = 0$) with the Lagrange multiplier to conserve the volume fraction. In Fig. 4, we show the snapshots of the phase function at different times obtained by the first-order MMM (33^3) and first-order FGM (129^3) with $\delta t = 0.001$. One observes that the two methods give visually identical results and the volume fraction is well conserved.

4.2. Phase-field model for two-phase flows

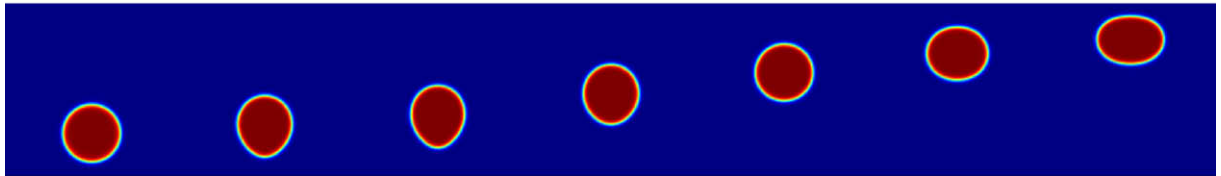
We now examine the accuracy and effectiveness of the moving mesh spectral method for the system (18).

4.2.1. A lighter bubble rising in a heavier medium

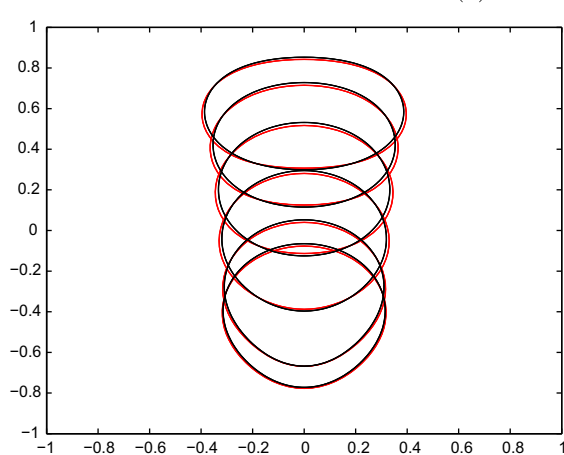
We consider a lighter bubble rising in a heavier medium. The governing equation is the phase-field model with a Boussinesq approximation for the variable density (3) and (4). We used the first-order MMM with a 75^2 grid described in Section 3 to solve the system (18e) in the computational domain. As a comparison, we also used a first-order FGM with a 257^2 grid in the original domain.



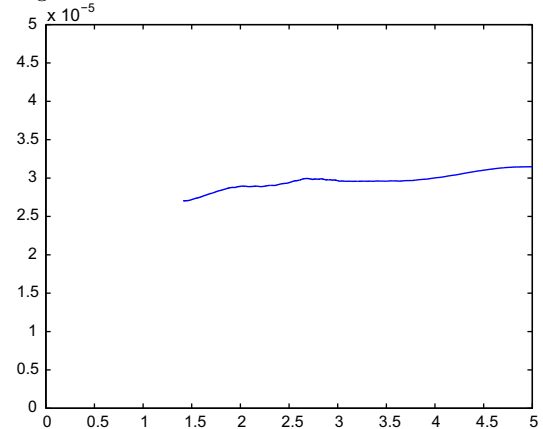
(a) MMM with a grid size of 75^2



(b) FGM with a grid size of 257^2



(c) Interfaces at different times: comparison of MMM with 75^2 (red) and FGM with 257^2 (black).



We take $\mu_1 = \mu_2 = 1, \rho_1 = \frac{1}{2}$ and $\rho_2 = \frac{3}{2}$ so $\rho_0 = \frac{\rho_1 + \rho_2}{2} = 1$. The forcing function due to the gravity is given by $f = -(1 + \phi)(\rho_1 - \rho_0) - (1 - \phi)(\rho_2 - \rho_0)g$ with the gravity constant $g = 9.8$ (cf. [18]). In Fig. 5(a) and (b)), we present snapshots of the phase function at different times obtained by the MMM and FGM. We observe that the two methods lead to essentially identical results. A more precise comparison is given in Fig. 5(c) where the level sets of $\{\phi : \phi = 0\}$ for MMM and FGM are plotted at different times. In Fig. 5(d), we plot the difference of the volume fractions at time t and the initial time. It is observed that the volume fraction is well conserved.

The simulation results in this example indicate that our MMM is capable of correctly capture the dynamics of moving interfaces in the phase-field model for two-phase flows.

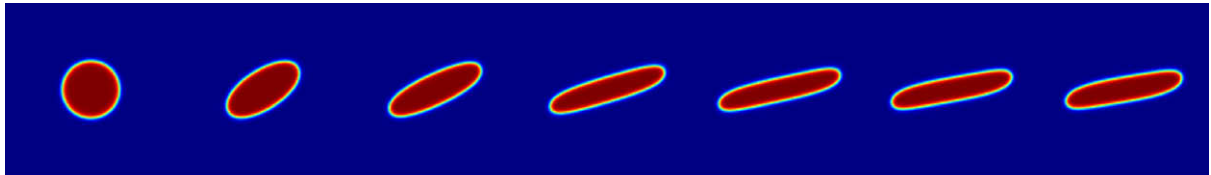
4.2.2. Bubble deformation under a shear stress

We now consider the bubble deformation under a shear stress. The governing equation is once again the phase-field model (18e) but now with $\rho_1 = \rho_2 = 1$ and $\mu_1 = \mu_2 = 1$.

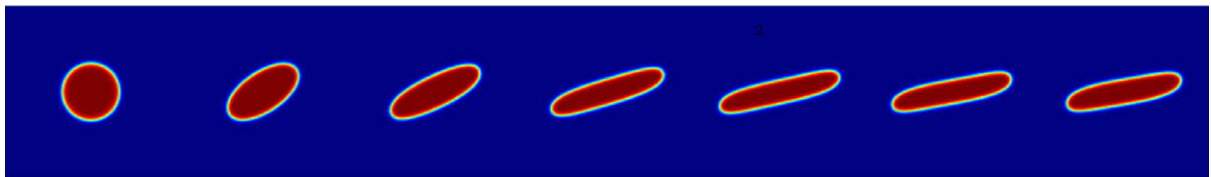
We first consider the two-dimensional case. The flow is driven by a shearing velocity boundary conditions:

$$(u, v)|_{y=\pm 1} = (\pm 2, 0) \quad \text{and} \quad (u, v)|_{x=\pm 1} = \pm 2(y, 0).$$

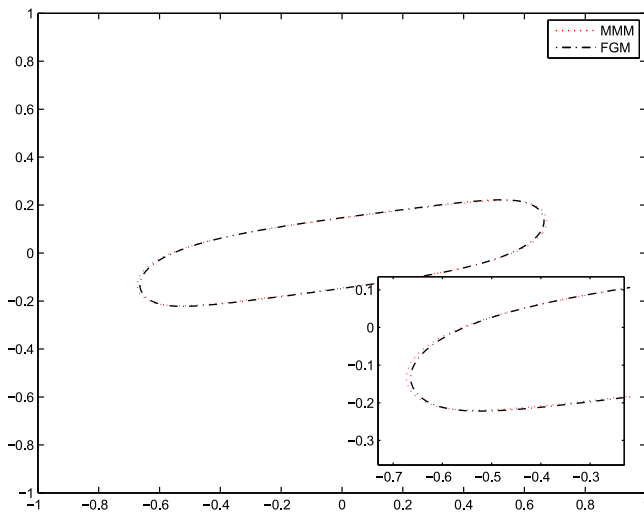
Homogeneous Dirichlet boundary conditions for the velocity are applied at other boundaries. In Fig. 6(a) and (b)), we show the snapshots of the phase function at different times by using the first-order MMM and FGM with $dt = 0.001$. No visible difference is noticeable. A more detailed comparison for the interfaces is given in Fig. 6(c).



(a) MMM with a grid size of 75^2



(b) FGM with a grid size of 257^2



(c) Interfaces at different times: comparison of MMM with 75^2 (red) and FGM with 257^2 (black).

Fig. 6. Bubble under shear: comparison of MMM and FGM with $\delta t = 0.001, \eta = 0.02, \gamma = 0.02, \lambda = 0.02$ at $t = 0, 0.5, 1, 2, 3, 4, 5$.

5. Concluding remarks

We have developed in this paper a moving mesh spectral method for the phase-field model of two-phase flows. The method is based on a variational moving mesh PDE for the phase function which determines a suitable transform that maps the phase function with large gradients to a smooth function. Then, we apply this transform to the phase-field model for two-phase flows and developed an efficient semi-implicit numerical scheme to solve the highly nonlinear and complex system (18e). Our numerical schemes are based on the following separate treatments for the moving mesh PDE, phase equation and Navier–Stokes equations:

- A semi-implicit treatment of the moving mesh PDE which only involves solving a Poisson-type equation at each time step.
- A semi-implicit stabilized scheme for the phase equation with Lagrange multiplier which requires solving an equation with variable coefficients in the form of (36) and which conserve the volume fraction exactly.
- A semi-implicit treatment of the Navier–Stokes equations based on a modified penalty formulation which only requires solving a Poisson equation for each component of the velocity and is unconditionally stable.

Then, the complete first-order scheme combines the above separate approaches by advancing consecutively x^{n+1} , $(\phi^{n+1}, \theta^{n+1})$ and u^{n+1} . A slightly more expensive second-order version of the scheme is also presented.

While the semi-implicit approach for each separate equation is stable with very mild stability constraint, the decoupled approach for the complete system leads to a somewhat restrictive time step. The cost of using smaller time step is usually more than compensated by faster solutions at each time step.

We presented several examples to illustrate the accuracy and effectiveness of our moving mesh spectral method. It is found that, at least for the examples considered in this paper, to achieve a comparable accuracy the moving mesh spectral method only needs to use about $\frac{1}{4}$ to $\frac{1}{3}$ of the points in each direction in a fixed grid spectral method. This represents a significant reduction in the number of unknowns, and consequently, in the total computational costs, particularly in three dimensions. Furthermore, with a decreased number of unknowns, the moving mesh spectral method allows much larger time steps than that is allowed by a corresponding fixed grid spectral method. This is a particular feature of the moving mesh spectral method for problems with non-periodic boundary conditions, since the time step of the fixed grid spectral method is constrained by the clustering of the collocation points near the boundary while the time step of the moving mesh spectral method will be constrained only by the spacing of grid points near the interface.

This appears to be a first successful attempt at using a moving mesh method in the context of spectral discretization for complex flows with non-periodic boundary conditions. The numerical simulations presented in the last section indicate that the moving mesh spectral method is robust and efficient for the phase-field model of two-phase flows.

References

- [1] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, *Acta Metall. Mater.* 27 (1979) 1085–1095.
- [2] D.M. Anderson, G.B. McFadden, A.A. Wheeler, Diffuse-interface methods in fluid mechanics, *Ann. Rev. Fluid. Mech.* 30 (1998) 139–165.
- [3] C. Bernardi, Y. Maday, *Approximations Spectrales de Problèmes aux Limites Elliptiques*, Springer-Verlag, Paris, 1992.
- [4] J.U. Brackbill, An adaptive grid with directional control, *J. Comput. Phys.* 108 (1) (1993) 38–50.
- [5] Weiming Cao, Weizhang Huang, Robert D. Russell, A study of monitor functions for two-dimensional adaptive mesh generation, *SIAM J. Sci. Comput.* 20 (6) (1999) 1978–1994 (Electronic).
- [6] H.D. Ceniceros, T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, *J. Comput. Phys.* 172 (2001) 609–639.
- [7] L.Q. Chen, Phase-field models for microstructure evolution, *Ann. Rev. Mater. Res.* 32 (2002) 113.
- [8] L.Q. Chen, Jie Shen, Applications of semi-implicit Fourier-spectral method to phase-field equations, *Comput. Phys. Commun.* 108 (1998) 147–158.
- [9] Yana Di, Ruo Li, Tao Tang, A general moving mesh framework in 3d and its application for simulating the mixture of multi-phase flows, *Commun. Comput. Phys.* 3 (2008) 582–602.
- [10] J.J. Feng, C. Liu, J. Shen, P. Yue, An energetic variational formulation with phase field methods for interfacial dynamics of complex fluids: advantages and challenges, *IMA Vol. Math. Appl.* 140 (2005) 1–26.
- [11] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, L.Q. Chen, Spectral implementation of an adaptive mesh method for phase-field equation, *J. Comput. Phys.* 220 (2006) 498–510.
- [12] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, L.Q. Chen, A Fourier spectral mesh method for the Cahn–Hilliard equation with elasticity, *Commun. Comput. Phys.* 5 (2009) 582–599.
- [13] J.L. Guermond, P. Mineev, Jie Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 6011–6045.
- [14] Weizhang Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, *J. Comput. Phys.* 171 (2001) 753–775.
- [15] Weizhang Huang, Variational mesh adaptation: isotropy and equidistribution, *J. Comput. Phys.* 174 (2) (2001) 903–924.
- [16] Weizhang Huang, Yuhe Ren, Robert D. Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.* 113 (2) (1994) 279–290.
- [17] Ruo Li, Tao Tang, Pingwen Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, *J. Comput. Phys.* 170 (2) (2001) 562–588.
- [18] Chun Liu, Jie Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, *Physica D* 179 (3–4) (2003) 211–228.
- [19] J. Lowengrub, L. Truskinovsky, Quasi-incompressible Cahn–Hilliard fluids and topological transitions, *Roy. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci.* 454 (1978) (1998) 2617–2654.
- [20] Jie Shen, Efficient spectral-Galerkin method I. Direct solvers for second- and fourth-order equations by using Legendre polynomials, *SIAM J. Sci. Comput.* 15 (1994) 1489–1505.
- [21] Jie Shen, On error estimates of the penalty method for the unsteady Navier–Stokes equations, *SIAM J. Numer. Anal.* 32 (1995) 386–403.
- [22] Jie Shen, Efficient Chebyshev–Legendre Galerkin methods for elliptic problems, in: A.V. Ilin, R. Scott (Eds.), *Proceedings of ICOSAHOM'95*, Houston J. Math, 1996, pp. 233–240.

- [23] Hervé Vandeven, Family of spectral filters for discontinuous problems, *J. Sci. Comput.* 6 (2) (1991) 159–192.
- [24] Alan M. Winslow, Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh, *J. Comput. Phys.* 1 (1967) 149–172.
- [25] X. Yang, J.J. Feng, C. Liu, J. Shen, Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method, *J. Comput. Phys.* 218 (2006) 417–428.
- [26] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex fluids, *J. Fluid Mech.* 515 (2004) 293–317.