Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 220 (2006) 498–510
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Spectral implementation of an adaptive moving mesh method for phase-field equations

W.M. Feng [a], P. Yu [b], S.Y. Hu [c], Z.K. Liu [a], Q. Du [a,b,*], L.Q. Chen [a]

[a] Department of Materials Science and Engineering, Penn State University, University Park, PA 16802, USA
[b] Department of Mathematics, Penn State University, 218 McAllister BLDG, University Park, PA 16802, USA
[c] Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 5 May 2006; accepted 14 July 2006
Available online 30 August 2006

Abstract

Phase-field simulations have been extensively applied to modeling microstructure evolution during various materials processes. However, large-scale simulations of three-dimensional (3D) microstructures are still computationally expensive. Among recent efforts to develop advanced numerical algorithms, the semi-implicit Fourier spectral method is found to be particularly efficient for systems involving long-range interactions as it is able to utilize the fast Fourier transforms (FFT) on uniform grids. In this paper, we report our recent progress in making grid points spatially adaptive in the physical domain via a moving mesh strategy, while maintaining a uniform grid in the computational domain for the spectral implementation. This approach not only provides more accurate treatment at the interfaces requiring higher resolution, but also retains the numerical efficiency of the semi-implicit Fourier spectral method. Numerical examples using the new adaptive moving mesh semi-implicit Fourier spectral method are presented for both two and three space dimensional microstructure simulations, and they are compared with those obtained by other methods. By maintaining a similar accuracy, the proposed method is shown to be far more efficient than the existing methods for microstructures with small ratios of interfacial widths to the domain size.

Keywords: Moving mesh; Adaptive mesh; Phase-field equations; Semi-implicit; Fourier-spectral method

1. Introduction

Phase-field method has been applied to modeling mesoscale morphological pattern formation and interface motion for many different materials processes [1]. It describes a microstructure using a set of spatially dependent field variables. The temporal evolution of the field variables is then governed by systems of time-dependent Ginzburg–Landau (TDGL) and Cahn–Hilliard (CH) equations. Numerical solutions to the phase-field

* Corresponding author. Address: Department of Mathematics, Penn State University, 218 McAllister BLDG, University Park, PA 16802, USA. Tel.: +1 814 8653674; fax: +1 814 865 3735.
E-mail address: qdu@math.psu.edu (Q. Du).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.07.013

equations yield the temporal and spatial evolution of the microstructures. However, most of the existing phase-field simulations employ the simple explicit Euler finite-difference scheme which has severe limitations on simulation time and system size. Consequently, a number of efforts have been made to develop and implement more advanced numerical algorithms for solving the phase-field equations. In general, existing algorithms are designed either to increase the numerical stability with respect to time or to achieve higher accuracy in spatial discretization. To achieve high accuracy in space, two types of approaches have been utilized. One is to employ a spectral representation of a continuous spatial profile of a field variable, e.g. using a Fourier series for a periodic system, and the other is the adaptive mesh approach in which dense grid points are used in the interfacial regions where the field variables have large gradients [2]. The spectral method and its semi-implicit implementation have proved particularly efficient for systems in which the morphologies and microstructures are dominated by long-range elastic interactions [3] while the adaptive mesh method is useful for microstructures with a very small interfacial width compared to the domain size. However, it is a technical challenge to efficiently combine the spectral method with an adaptive mesh. The main objective of this paper is to develop a FFT based spectral implementation of an adaptive mesh method for solving phase-field equations.

To achieve the adaptivity within the framework of Fourier-Spectral Semi-implicit methods, we employ the moving mesh approach while maintaining the same number of Fourier modes instead of using local refinement (either by adding extra grid points or enriching Fourier modes). The main idea of the moving mesh approach is to construct a time-dependent mapping $x(\xi,t)$ from the computational domain Ω_c (parameterized by ξ) to the physical domain Ω_p (parameterized by x), such that the representation $v(\xi,t)=u(x(\xi,t))$ of the physical solution $u(x(\xi,t))$ in the computational domain is "better behaved". The criteria for constructing the mapping are usually expressed as certain variational principles, whose solutions via gradient flow lead to the so-called moving mesh partial differential equations (MMPDEs) [4–9]. Similar domain or coordinate mapping ideas have been also used in [10,11] for the adaptive pseudo spectral approximation of reaction-diffusion and combustion problems. Other works on adaptive spectral methods can also be found in [12,13]. It turns out the semi-implicit Fourier spectral method can also be effectively used to solve the MMPDEs. Taking advantages of both the moving mesh method and the Fourier Spectral Semi-implicit scheme, larger time steps and larger system sizes can be used in phase-field simulations to gain computational efficiency without sacrificing the accuracy. In this paper, we demonstrate the performance of this new approach for the Phase-field equation in both two and three space dimensions for model problems. Similar improvement can also be expected for its application to the phase filed simulations of more realistic and complicated problems.

The rest of the paper is organized as follows: we first review the framework for the MMPDEs, and discuss its Fourier-Spectral implementation, in particular, for the Phase-field equation. We then present numerical simulation results and make comparisons with other existing methods. Some concluding remarks are given in the end.

2. Formulations of moving mesh PDEs and applications to Phase-field equations

We first present the variational formulations of the moving-mesh PDEs, then we discuss the applications to Phase-field equations and their spectral implementation.

2.1. The moving-mesh PDEs

Moving-mesh PDEs can be formulated either on a computational domain [14] or on a physical domain [15]. The former has the advantage of being simple and efficient, though bearing a lesser rigorous derivation. The latter is derived on a more rigorous basis, but the resulting MMPDE is slightly more complicated. More comparisons of the implementation based on the two different approaches are given in [16]. In this paper, we briefly discuss both approaches, although only the second approach, the physical domain variational formulation (PDVF), is implemented in the numerical simulations.

To explain the idea, we first describe the MMPDE in one dimension. Fig. 1 shows the discretization of a function in the physical domain and in the computational domain, respectively. One can achieve the high grid density in the high gradient region in the physical domain (Fig. 1, left) by smoothing the gradient in the computational domain (Fig. 1, right).

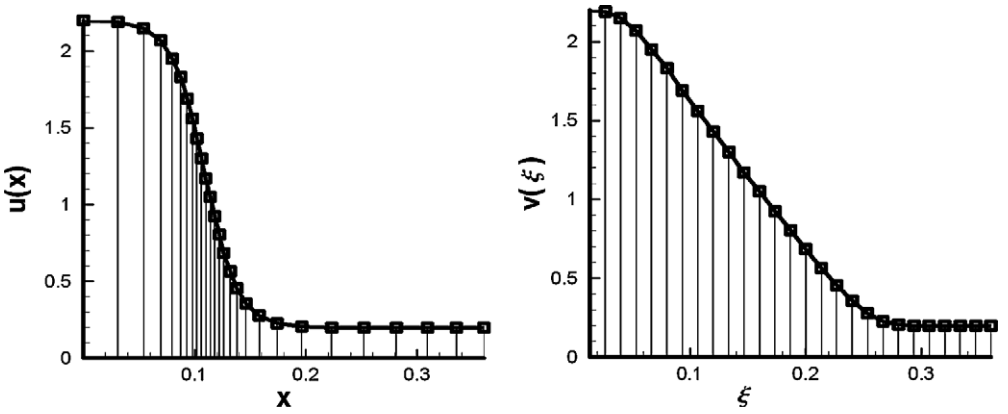

Fig. 1. The physical domain (left) and the computational domain (right).

To make $v(\xi,t) = u(x(\xi,t))$ a nicely behaved function, we seek the solution to the following minimization problem:

$$\min_{x(\xi,t)} \int_{\Omega_c} \omega \mathrm{d}\xi, \tag{1}$$

where

$$\omega = \sqrt{1+v_\xi^2} = \sqrt{1+u_x^2(x(\xi,t))x_\xi^2}. \tag{2}$$

This special functional corresponds to the arc length and its associated Euler–Lagrange equation in the computational domain is given by

$$\left(\frac{u_x^2}{\sqrt{1+u_x^2x_\xi^2}}x_\xi\right)_\xi = \frac{u_xu_{xx}x_\xi^2}{\sqrt{1+u_x^2x_\xi^2}}. \tag{3}$$

The solution to the above equation gives the transformation, $x(\xi,t)$, between the computational domain described by ξ to the physical domain by x. The PDE in Eq. (3) is difficult to solve since it involves a rather singular coefficient on the left-hand side and a stiff source term on the right-hand side. However, through numerical experiments [14], it is found that one can achieve the same goal of smoothing the physical solution in the computational space by solving a much simpler PDE,

$$\frac{\partial}{\partial\xi}(\omega x_\xi) = 0, \tag{4}$$

where $\omega = \sqrt{1+u_x^2x_\xi^2}$ or $\sqrt{1+u_\xi^2}$ is the so-called monitor function that connects the mesh with the physical solution. In general, the monitor function should be problem-dependent as this function ultimately determines the compression and deformation of the mesh. The mesh will concentrate more on places where ω is larger. Other geometric controls such as mesh orientation and orthogonality can also be built in the monitor function [17]. A common choice is $\omega = \sqrt{1+\beta^2u_\xi^2}$ where β is a scaling constant for the control of mesh concentration. Since both the MMPDE (4) and the monitor function are expressed in computational domain, the implementation is rather straightforward.

For the PDVF approach [15], the goal is to find the inverse mapping $\xi = \xi(x,t)$ of $x = x(\xi,t)$. Given an appropriate matrix monitor function G, the mapping can be obtained by solving the following variational problem

$$\min_{\xi(x,t)} = \int_{\Omega_p} \omega^{-1}\left(\frac{\partial\xi}{\partial x}\right)^2 \mathrm{d}x. \tag{5}$$

In higher dimensions, the objective function in (5) becomes

$$I[\xi] = \int_{\Omega_p} \sum_i (\nabla \xi^i)^T G^{-1} \nabla \xi^i \, dx \tag{6}$$

where ∇ is the gradient operator with respect to x, ω is the monitor function.

The Euler–Lagrange equation associated with this variational problem is

$$\frac{\delta I[\xi]}{\delta \xi^i} = 0 \quad \text{or} \quad \nabla \cdot (G^{-1} \nabla \xi^i) = 0, \quad i = 1, 2, 3. \tag{7}$$

The static PDE (7) can be converted to a time-dependent problem via gradient flow:

$$\frac{\partial}{\partial t} \xi^i(x,t) = p \nabla \cdot (G^{-1} \nabla \xi^i), \quad i = 1, 2, 3, \tag{8}$$

where the mobility p is a positive function and can vary in space.

Some complications arise when one converts (8) to a more convenient version in the computational space. This is done by interchanging the roles of dependent and independent variables in (8). Following [15], it is convenient to introduce the covariant and contravariant basis vectors

$$a_i = \frac{\partial x}{\partial \xi^i}, \quad a^i = \nabla \xi^i, \quad i = 1, 2, 3, \tag{9}$$

which are related by

$$a^i = \frac{1}{J} a_j \times a_k, \quad a_i = J a^j \times a^k, \quad a_i \cdot a^l = \delta_i^l, \quad (i, j, k \text{ being cyclic}), \tag{10}$$

where δ_i^l is the Kronecker delta function, and J is the Jacobian $J = a_1 \cdot (a_2 \times a_3)$ [15].

With the aid of the following formulae:

$$\nabla = \sum_i a^i \frac{\partial}{\partial \xi^i} = \frac{1}{J} \sum_i \frac{\partial}{\partial \xi^i} J a^i, \quad \frac{\partial x}{\partial t} = -\sum_i a_i \frac{\partial \xi^i}{\partial t} \tag{11}$$

and

$$\frac{\partial J}{\partial \xi^l} = J \sum_i a^i \cdot \frac{\partial a^i}{\partial \xi^l}, \quad \frac{\partial a^i}{\partial \xi^l} = -\sum_s \left(a^i \cdot \frac{\partial a_s}{\partial \xi^l} \right) a^s, \tag{12}$$

the Eq. (8) can be transformed into

$$\frac{\partial x}{\partial t} = p \left[\sum_{i,j} \left(a^i \cdot G^{-1} a^j \right) \frac{\partial^2 x}{\partial \xi^i \partial \xi^j} - \sum_{i,j} \left(a^i \cdot \frac{\partial G^{-1}}{\partial \xi^j} a^j \right) \frac{\partial x}{\partial \xi^i} \right]. \tag{13}$$

In the case of Winslow type of monitor function $G = \omega i$ [18], with i being the identity matrix, the above equation can be further simplified into

$$\frac{\partial x}{\partial t} = \frac{p}{\omega^2} \sum_{i,j} (a^i \cdot a^j) \frac{\partial}{\partial \xi^i} \left(\omega \frac{\partial x}{\partial \xi^j} \right). \tag{14}$$

In this paper, we focus only on the case where $G = \omega i$, though more general matrix forms can also be used in order to introduce anisotropic adaptive grids. We adopt the following variable mobility in practical calculation

$$p = \mu \omega^2 / \lambda, \tag{15}$$

where λ is the largest eigenvalue of the positive-definite matrix $A = (A_{i,j}) = (a^i \cdot a^j)$. Here, the generic constant μ introduces an artificial time scale for the MMPDE as compared with the physical time t. It should be noted that our choice of the mobility function is simpler than the form suggested in [15]. Our motivation for choosing such a mobility is to make (14) behave similarly to a simple diffusion equation, and our numerical experiences seem to indicate that this choice helps increase the stability of the MMPDE significantly.

2.2. Fourier-spectral implementation of MMPDEs

In [15], three types of boundary conditions have been suggested to supplement the moving mesh PDEs: (1) Dirichlet conditions where the boundary points are held fixed; (2) orthogonal conditions where one set of coordinate lines are required to be orthogonal to the physical boundary; and (3) boundary conditions determined by a lower dimensional MMPDE. In this paper, we discuss an implementation of the periodic boundary conditions for the MMPDEs, since periodic boundary conditions are the most commonly used in phase-field simulations. Suppose the computational domain is a unit square $[0,1]\times[0,1]$. Though periodic boundary conditions can be understood in a straightforward way on the unit square, it is not immediately clear how the resulting adaptive mesh inherits the periodic properties. In fact, it is the displacement of the adaptive grid point from its inverse image on the regular grid, i.e. $x(\xi,t)-\xi$, that satisfies the periodic boundary conditions on the unit square, i.e.,

$$x(\xi+(k,l),t)=x(\xi,t)+(k,l),\quad \text{integer pair}(k,l). \tag{16}$$

Interestingly, this condition does not require that the mapping $x(\xi,t)$ maps a unit square onto a unit square. In other words, with periodic boundary conditions, the physical domain may not turn out to be a square even though the computational domain is. This can be seen in Fig. 2 which shows a mesh adapting to a circular interface centered near the upper-right corner of the physical domain. Although the physical domain boundaries are curved, the condition (16) guarantees that the periodic copies of Ω_p (non-square) cover the whole two-dimensional space as effectively as periodic copies of the unit square. In particular, it is easy to verify that (16) implies that the area of the physical domain Ω_p is the same as that of Ω_c.

Consider the PDVF approach in [15], with periodic boundary conditions on $X=x-\xi$, Eq. (14) becomes

$$\frac{\partial X}{\partial t}=\mu\sum_{i,j}\frac{a^i\cdot a^j}{\lambda}\frac{\partial}{\partial\xi^i}\left[\omega\left(\frac{\partial X}{\partial\xi^j}+e^j\right)\right], \tag{17}$$

where e^j is the canonical unit vector (the ith column of the identity matrix). This can be solved by the semi-implicit Fourier spectral method [3] through

$$\frac{\overline{X}^i-X^i}{\Delta t}-\mu W\nabla_\xi^2\overline{X}^i=\mu\sum_{i,j}\frac{a^i\cdot a^j}{\lambda}\frac{\partial}{\partial\xi^i}\left[\omega\left(\frac{\partial X}{\partial\xi^j}+e^j\right)\right]-\mu W\nabla_\xi^2X^i, \tag{18}$$

where $\overline{X}$ is the value of X at the next time step and W is the maximum of ω on Ω_c. An equivalent representation in the Fourier space is

$$(1+\mu W\Delta t\kappa^2)\left(\widehat{\overline{X}}-\widehat{X}\right)=\mu\Delta t\left\{\sum_{i,j}\frac{a^i\cdot a^j}{\lambda}\frac{\partial}{\partial\xi^i}\left[\omega\left(\frac{\partial X}{\partial\xi^j}+e^j\right)\right]\right\}^{\wedge}, \tag{19}$$

where the $\hat{}$ represents the Fourier transform.

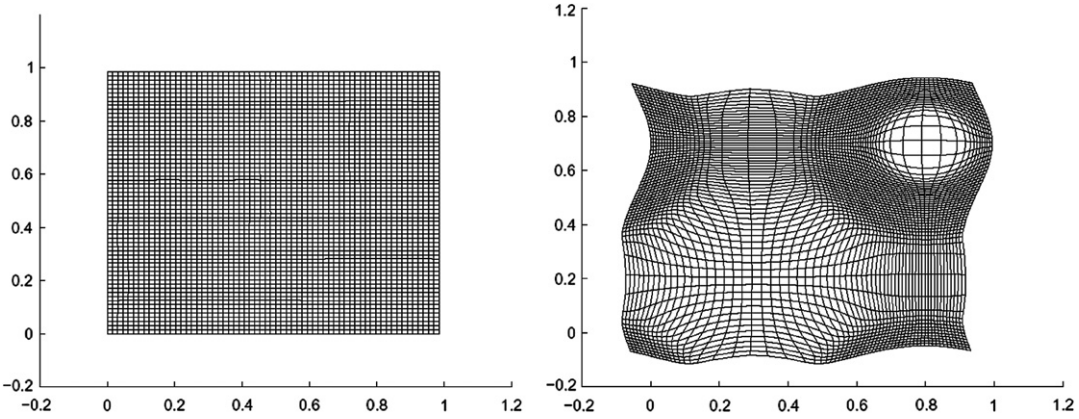

Fig. 2. A domain mapping from regular grids on the computational domain (unit square) to irregular grids on the physical domain (non-square).

2.3. Phase-field equations

In the phase-field simulation, MMPDEs are solved together with the phase-field equations alternately. We now discuss how the phase-field equations are transformed into the computational domain and their numerical solutions. As an illustration, we consider the Allen–Cahn equation with a constant mobility [1]:

$$\frac{\partial}{\partial t}\eta(x,t) = f(\eta) + \nabla_x^2\eta. \tag{20}$$

We also refer the above equation as a time-dependent Ginzburg–Landau (TDGL) equation. The nonlinear term $f = f(\eta)$ is specified later. Now, by introducing the variable transformation $x = x(\xi, t)$ and treating η as a function of ξ and t, we have

$$\frac{\partial}{\partial t}\eta(\xi,t) = \dot{x}\cdot\nabla_x\eta + f(\eta) + \nabla_x^2\eta, \tag{21}$$

where $\dot{x} = \frac{\partial x(\xi,t)}{\partial t}$ is the velocity of the moving mesh determined from the MMPDEs. The extra convection term $\dot{x}\cdot\nabla_x\eta$ represents the change of the inverse image of the field variable η on the computational domain due to the mesh motion. Using (11) and (12), one may easily verify that

$$\nabla_x\eta = \sum_i a^i\frac{\partial\eta}{\partial\xi^i} \tag{22}$$

and

$$\nabla_x^2\eta = \frac{1}{J}\nabla_\xi\cdot(JA\nabla_\xi\eta), \tag{23}$$

where $A = (A_{i,j}) = (a^i\cdot a^j)$. In order to apply the semi-implicit Fourier spectral scheme, we denote by λ^* the maximum among the eigenvalues of A over the whole domain Ω_c (maximum of λ in (15)) and introduce the splitting

$$\frac{\bar{\eta}-\eta}{\Delta t} - \lambda^*\nabla_\xi^2\bar{\eta} = \dot{x}\cdot\nabla_x\eta + f(\eta) + \nabla_x^2\eta - \lambda^*\nabla_\xi^2\eta, \tag{24}$$

whose counterpart in the Fourier space conjugate to Ω_c reads

$$(1+\lambda^*\Delta t\kappa^2)(\hat{\bar{\eta}}-\hat{\eta}) = \Delta t\{\dot{x}\cdot\nabla_x\eta + f(\eta) + \nabla_x^2\eta\}^\wedge. \tag{25}$$

3. Numerical simulations and discussion

We now report numerical simulation results using the AFSIM (adaptive Fourier-spectral semi-implicit method) for a popular bench-mark problem and some more realistic problems.

3.1. Two dimensional implementation

3.1.1. Temporal evolution of a circular domain

The first example presented here is chosen for testing the accuracy and efficiency of the proposed moving mesh based adaptive Fourier-spectral Semi-implicit method (AFSIM) since it is the most commonly used bench-mark example for which an analytical solution for the temporal evolution kinetics is known. In light of the numerical studies already carried out in [3] which demonstrated the high efficiency and accuracy of uniform Fourier-spectral semi-implicit method (UFSIM), we only need to compare the performance of AFSIM and UFSIM.

To compare the accuracy and efficiency of various schemes, we considered the TDGL Eq. (20) in two dimensions (with the system or domain size being 256×256) with various grid sizes $\mathrm{d}x$ and time step sizes $\mathrm{d}t$ (because the system size is fixed, if we for example use $\mathrm{d}x = 4$, then it implies the number of grid points used is 64×64). We use the double well free energy, $F = -1/2\eta^2 + 1/4\eta^4$, so that we have $f(\eta) = -\partial F/\partial\eta = \eta - \eta^3$ in the Eq. (20). In the implementation of AFSIM, we take $\beta^2 = 0.15$ in the monitor function

and pick $\mu = 1$ to get the mobility in the Eq. (15). These are the only extra parameters required in the computer simulation other than those already used in the standard UFSIM.

We first consider the shrinkage kinetics of a circular domain in two dimensions. Both the equilibrium profiles and the velocity of a moving interface are monitored. At $t = 0$, a circular domain with a radius of 100 is prescribed. The phase-field parameter values inside the circle are assigned -1 and $+1$ outside. Such a circular interface is unstable and the driving force for its motion is the mean curvature. Therefore, the circle will shrink and eventually disappear (Fig. 3). If the radius of the circle is much larger than the interfacial thickness, the velocity of the moving interface, V, is given by [19]

$$V = \frac{dR}{dt} = -\frac{1}{R}, \tag{26}$$

where R is the radius of the circle at a given time t. Furthermore, the area of the circle as a function of time can be described as

$$A = A_0 - 2\pi t, \tag{27}$$

where A is the area of the circle at time t and A_0 is the initial area.

The area of the circle is proportional to the time with a slope -2π. The areas as a function of time obtained from both schemes are shown in Fig. 4. All the results show a linear dependence of the area on time. However, the slopes of the lines are different. The thick solid line labeled as "theory" is a plot of Eq. (27) with a slope of -2π. We may characterize the accuracy of different schemes by comparing the slopes of the lines with the analytical solution (27). The resulting errors are shown in Table 1.

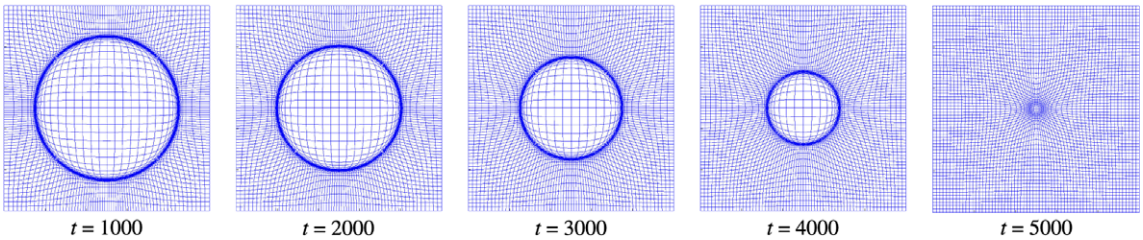

Fig. 3. Mesh distribution corresponding to the temporal evolution of a two dimensional circular domain ($dx = 4$).

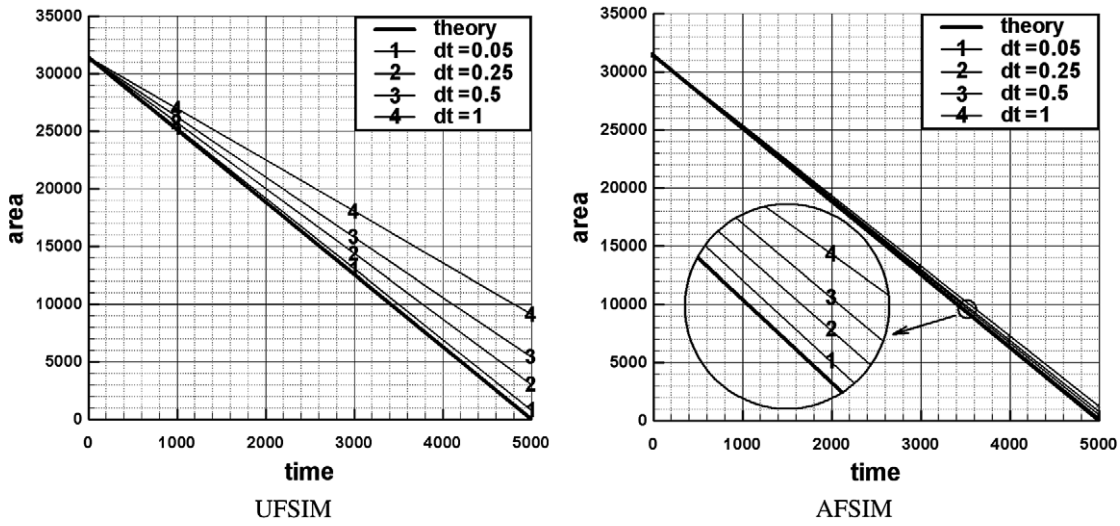

Fig. 4. Areas of a circular domain as a function of time.

Notice that the errors of UFSIM range from 2.4733% to 28.9439%, which yield large discrepancies in the solutions. On the other hand, there is no significant difference between the errors obtained from AFSIM, with different grid sizes dx and time step sizes dt, which means the newly developed AFSIM is much more accurate.

The overall computational capacity and computing time are crucial issues in large-scale simulations, especially three-dimensional simulations. AFSIM can significantly increase the spatial scale of the simulation and improve the efficiency, while ensuring the computation accuracy as well. Table 2 shows the computing time of both schemes with different grid sizes and time step sizes. One may notice that, by providing the same kinetics (Fig. 5) and a similar accuracy (2.4205% vs. 2.4733%), the AFSIM with parameter values (d$x = 4$, d$t = 0.5$) is about 18 times faster than the UFSIM with (d$x = 1$, d$t = 0.05$), in addition to potential memory savings by using fewer grid points.

3.1.2. Temporal evolution of a random domain structure

AFSIM is expected to perform well for microstructures with a small fraction of interfaces. To simulate a more realistic microstructure, we consider the evolution of a random domain structure. The same free energy

Table 1
The errors in the slopes of the area dependence (2D)

Schemes	d$t = 0.05$ (%)	d$t = 0.25$ (%)	d$t = 0.5$ (%)	d$t = 1$ (%)
UFSIM d$x = 1$	2.4733	9.5397	17.1410	28.9439
AFSIM d$x = 1$	0.6978	1.5801	2.4204	3.8661
AFSIM d$x = 2$	0.7021	1.5792	2.4220	3.8655
AFSIM d$x = 4$	0.7168	1.5743	2.4205	3.8644

Table 2
The computing time for the 2D problem (unit: s)

Schemes	d$t = 0.05$	d$t = 0.25$	d$t = 0.5$	d$t = 1$
UFSIM d$x = 1$	381	76	38	18
AFSIM d$x = 1$	8220	1656	745	362
AFSIM d$x = 2$	1390	266	133	67
AFSIM d$x = 4$	220	41	21	10

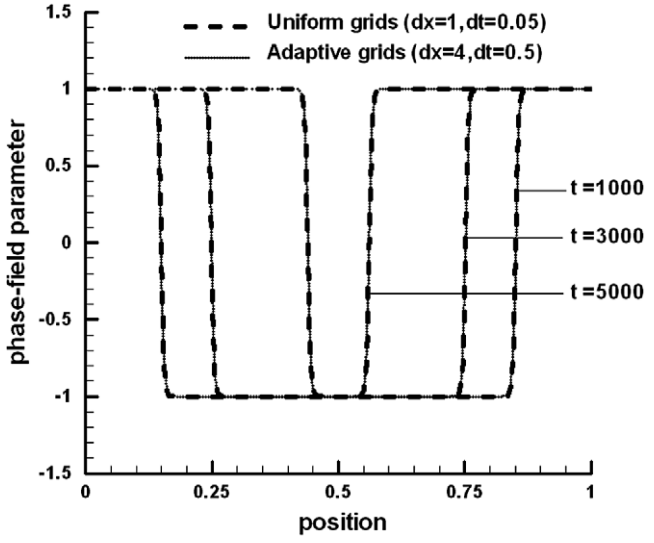

Fig. 5. Phase-field parameter profile evolution with time.

and parameters used above are employed. At $t = 0$, the initial distribution of phase-field parameter is around 0 with a spatial small random noise valued within $(-0.5, 0.5)$. Interconnected structures shown in Fig. 6 are obtained during evolution using UFSIM.

To compare, we take the distribution at $t = 50$ using UFSIM as the input for AFSIM. Fig. 7 shows the mesh distribution during evolution using AFSIM. The associated phase-field parameter distribution is plotted in Fig. 8. It can be seen that Figs. 6 and 8 agree quite well. Although the accuracy of these two methods cannot be compared quantitatively because there is no analytical solution for this case, the efficiency of the two methods are studied and it is found that AFSIM using smaller number of Fourier modes and larger time step is three times more efficient than UFSIM with more Fourier models and smaller time step.

As an additional example, we use the same parameters as in the above example, but take a tilted double well free energy: $F = 0.2\eta - 1/2\eta^2 - 0.2 * 1/3\eta^3 + 1/4\eta^4$. Thus, in Eq. (20), we have $f(\eta) = -\partial F/\partial\eta = (\eta - 0.2)(1 - \eta^2)$. At $t = 0$, the initial distribution of phase-field parameter is taken to be around 0.3 with a spatially distributed small random noise with values in $(-0.5, 0.5)$. Fig. 9(a) is obtained using UFSIM with $dx = 1$, $dt = 0.05$ and the plots correspond to solutions at $t = 100$ and $t = 150$, respectively during the microstructure evolution. To compare with the AFSIM, we take the simulation result of the UFSIM at $t = 100$ as the input for AFSIM. Fig. 9(b) shows the mesh distribution (see Fig. 9(b) left) during evolution (from $t = 100$ to $t = 150$) using the AFSIM with $dx = 2$, $dt = 0.5$ and the associated phase-field parameter distribution (Fig. 9(b) right).

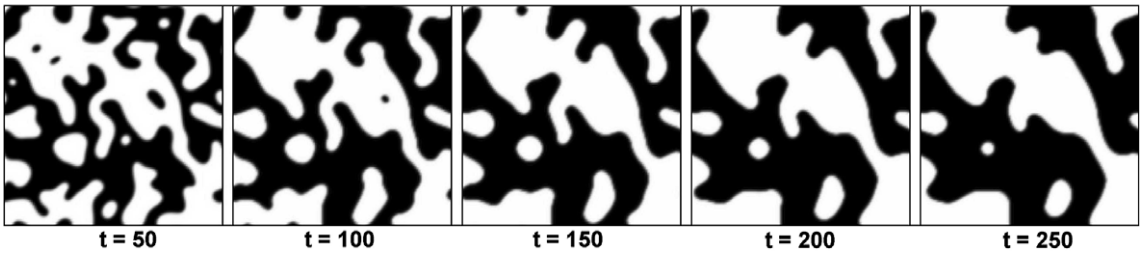

Fig. 6. Phase-field parameter distribution (UFSIM, $dx = 1$, $dt = 0.05$).

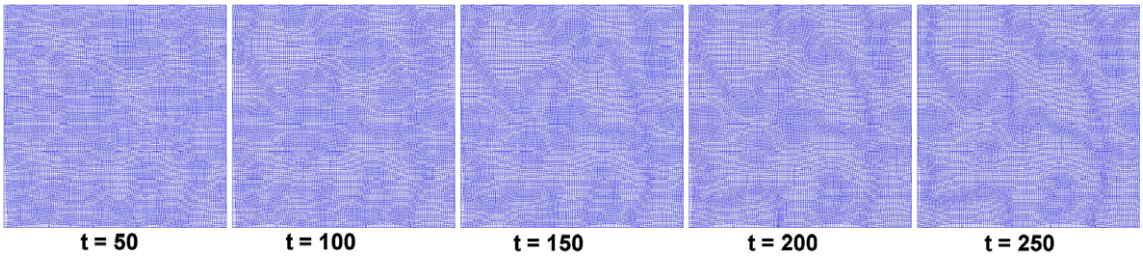

Fig. 7. Mesh distribution (AFSIM, $dx = 2$, $dt = 0.5$).

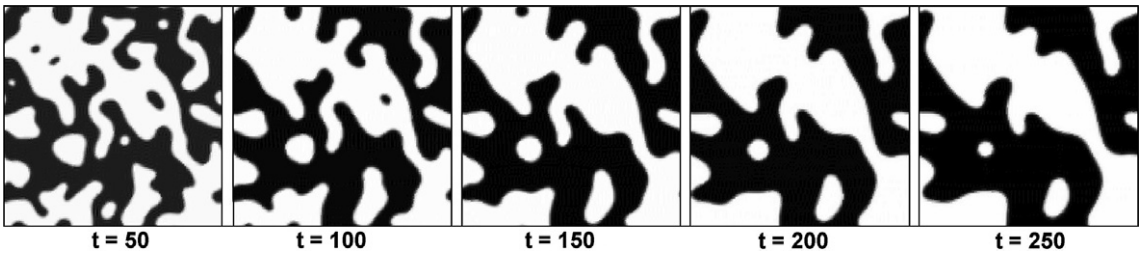

Fig. 8. Phase-field parameter distribution (AFSIM, $dx = 2$, $dt = 0.5$).

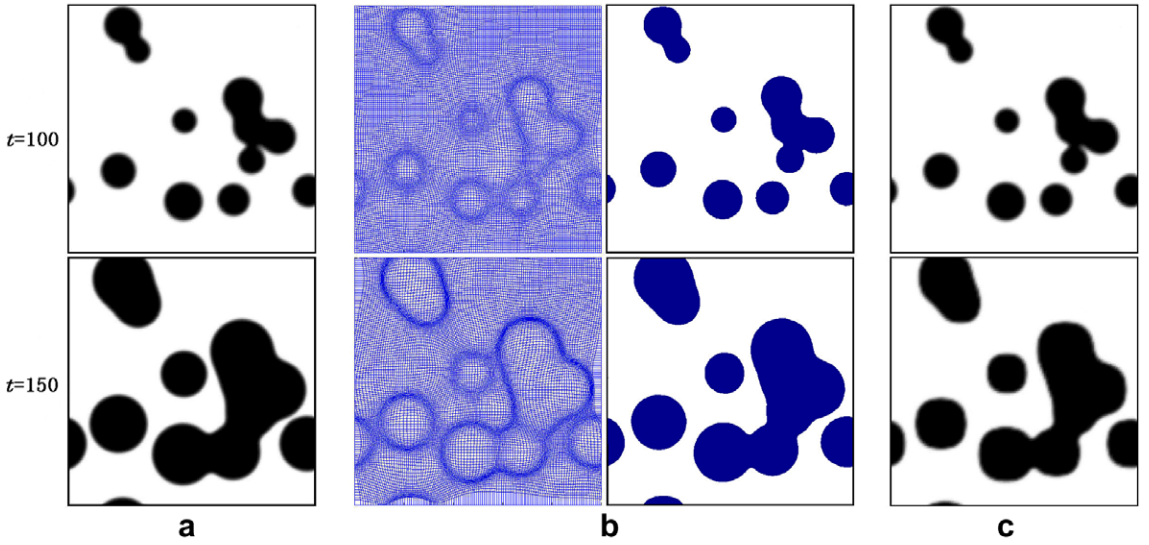

Fig. 9. Temporal evolution of a random domain structure.

It can be seen that Fig. 9(a) and (b) agree again very well, with AFSIM being again three times more efficient than to obtain than UFSIM. In comparison, Fig. 9(c) shows the phase-field parameter distribution from $t = 100$ to $t = 150$ using UFSIM with $dx = 2$, $dt = 0.5$. Notice that the microstructure at $t = 150$ displayed in Fig. 9(c) show a square like morphology instead of the circular one showing in Fig. 9(a) and (b) computed by more accurate simulations. Thus, the UFSIM with the same degree of freedom as the AFSIM produces qualitatively different results and is unable to accurately reproduce the same morphology as shown in the AFSIM and the finer UFSIM calculation. This again illustrates the enhanced resolution provided by AFSIM over the UFSIM when the same number of grid points and step size are used.

3.2. Three dimensional implementation

While a number of 1D and 2D moving mesh methods have been developed in the last 20 years, such as those in [2,4–8,20,21], few have been applied in three spatial dimensions [6,22,26]. Two and three dimensional adaptive Phase-field simulations have been implemented before but often in the finite difference and finite element contexts [23–25]. To our knowledge, the combination of moving mesh strategy with the Fourier spectral discretization in three dimensions in the context of phase field modeling has not been attempted in the literature before.

Here we discuss the three-dimensional implementation of AFSIM scheme. In the following, we considered the Allen–Cahn (or TDGL) Eq. (20) in three dimensions with a system size $256 \times 256 \times 256$ and various grid sizes dx and time step sizes dt (again, the system size is fixed, e.g. $dx = 4$ corresponds to a $64 \times 64 \times 64$ grid). Similar to the first example in the two-dimensional case, we assume that at $t = 0$, there is a spherical domain with a radius of 100. The sphere will shrink and eventually disappear as illustrated in Fig. 10. The velocity of the moving interface, V, is given by

$$V = \frac{dR}{dt} = -\frac{2}{R}, \tag{28}$$

where R is the radius of the circle at a given time t. Furthermore, the volume of the sphere as a function of time can be described as

$$\left(\frac{9\pi V^2}{16}\right)^{\frac{1}{3}} = \left(\frac{9\pi V_0^2}{16}\right)^{\frac{1}{3}} - 4\pi t, \tag{29}$$

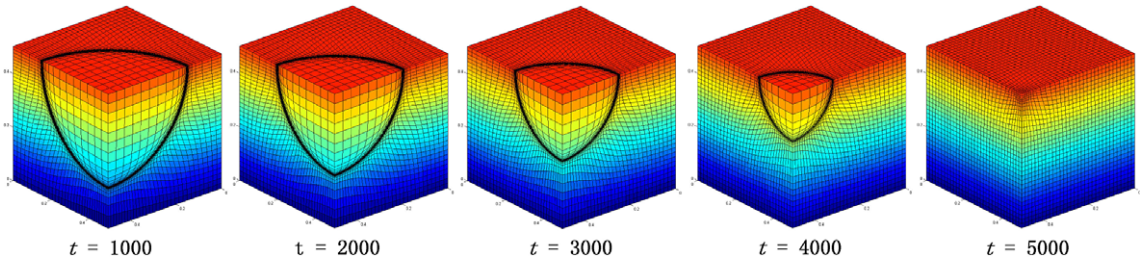

Fig. 10. Mesh distributions for the temporal evolution of a shrinking three dimensional spherical domain (cross-sections in a quadrant are shown, $dx = 4$).

where V is the volume of the sphere at time t and V_0 is the initial volume. Again, the errors of different schemes by comparing the slopes of the lines with the analytical solution (29) are shown in Table 3.

Table 4 shows the computing time of both schemes with different grid sizes and time step sizes. One may notice that, by providing a similar accuracy (2.4135% vs. 2.4642%), the AFSIM with parameter values ($dx = 4$, $dt = 0.5$) is about 18 times faster than the UFSIM with ($dx = 1$, $dt = 0.05$), in addition to potential memory savings by using fewer grid points.

3.3. Discussions

By taking advantages of the moving mesh method, larger time steps can be used to gain computational efficiency without sacrificing the accuracy, in addition to the savings of Fourier-spectral semi-implicit method. This makes sense because the stiffness of the solution is reduced in the computational domain. Also since we got high grids density in high gradient area, we can use less grid points to save the computing time, e.g. we used 64×64 grid points to calculate a system sizes of 256×256.

We should notice that the new AFSIM requires more CPU time in constructing a time-dependent mapping from the computational domain to the physical domain, which constitutes the overhead of the moving mesh strategy. Counting the cost of solving MMPDE and Allen–Cahn equation together, we need 32 Fourier transforms in 2D and 66 Fourier transforms in 3D. In contrast, solving the simplest Allen–Cahn equation on a regular uniform grid requires only three Fourier transforms. Therefore, we may need a compression ratio of four in each dimension to gain some savings by using the adaptive grids. The savings of course would be much more apparent if an Allen–Cahn equation with variable mobility had been considered. For example, in 2D, the Allen–Cahn equation with a variable mobility generally requires seven Fourier transforms on a regular uniform grid, then an adaptive grid with a compression ratio of three would already noticeably reduce the overall computational cost. As the number of Fourier modes is reduced, we also see potential savings in the memory, in spite of the extra memory overhead allocated for the additional variables, which is often another crucial concern in large scale three dimensional simulations.

Table 3
The errors in the slopes of the area dependence (3D)

Schemes	$dt = 0.05$ (%)	$dt = 0.25$ (%)	$dt = 0.5$ (%)	$dt = 1$ (%)
UFSIM $dx = 1$	2.4642	9.5292	17.0555	28.9232
AFSIM $dx = 4$	0.7023	1.5660	2.4135	3.8639

Table 4
The computing time for the 3D problem (unit: h)

Schemes	$dt = 0.05$	$dt = 0.25$	$dt = 0.5$	$dt = 1$
UFSIM $dx = 1$	41.03	8.22	4.07	2.01
AFSIM $dx = 4$	22.64	4.53	2.27	1.15

4. Conclusions

A new numerical scheme has been developed to solve the phase-field equations. It combines the adaptive moving mesh method with the semi-implicit Fourier spectral algorithm. The scheme has been implemented in both two and three dimensions for solving a single Allen–Cahn (TDGL) equation. It is demonstrated that for a prescribed accuracy in both the equilibrium profile of a phase-field parameter and the interface velocity, the new moving mesh method can potentially lead to an order of magnitude improvement in efficiency over the conventional uniform mesh spectral method, particularly for problems that the volume fraction of interfacial region is small. It is naturally possible to apply this scheme to systems of Allen–Cahn, Cahn–Hilliard, and systems involving elasticity equations (currently under development and see [16] for additional discussions). The use of a hydrodynamic approach for the moving mesh PDEs is also being explored [27].

Acknowledgments

This work is supported by the National Science Foundation Information Technology Research Project (NSF-ITR) through Grant DMR-0205232. The work of Qiang Du and Peng Yu are also supported by NSF-DMS-0409297.

References

[1] L.Q. Chen, Phase-field models for microstructure evolution, Annu. Rev. Mater. Sci. 32 (2002) 113–140.
[2] J. Mackenzie, M.L. Robertson, A moving mesh method for the solution of the one-dimensional phase-field equations, J. Comput. Phys. 181 (2002) 526–544.
[3] L.Q. Chen, J. Shen, Applications of semi-implicit Fourier-spectral method to Phase-field equations, Comput. Phys. Commun. 108 (1998) 147–158.
[4] W. Huang, Y. Ren, R.D. Russell, Moving mesh methods based on moving mesh partial differential equations, J. Comput. Phys. 113 (1994) 279–290.
[5] W. Huang, Y. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle, SIAM J. Numer. Anal. 31 (1994) 709–730.
[6] W. Huang, R.D. Russell, A high dimensional moving mesh strategy, Appl. Numer. Math. 26 (1997) 63–76.
[7] W. Huang, R.D. Russell, Moving mesh strategy based upon a gradient flow equation for two dimensional problems, SIAM J. Sci. Comput. 20 (1999) 998–1015.
[8] P. Bochev, G. Liao, G. dela Pena, Analysis and computation of adaptive moving grids by deformation, Numer. Methods Partial Diff. Equat. 12 (1998) 489–506.
[9] L. Mulholland, W. Huang, D.M. Sloan, Pseudospectral solution of near-singular problems using numerical coordinate transformations based on adaptivity, SIAM J. Sci. Comput. 19 (1998) 1261–1289.
[10] A. Bayliss, D. Gottlieb, B.J. Matkowsky, M. Minkoff, An adaptive pseudo-spectral method for reaction-diffusion problems, J. Comput. Phys. 81 (1989) 421–443.
[11] A. Bayliss, R. Kuske, B.J. Matkowsky, A two-dimensional adaptive pseudo-spectral method, J. Comput. Phys. 91 (1990) 174–196.
[12] C. Mavriplis, Adaptive mesh strategies for the spectral element method, Comput. Methods Appl. Mech. Eng. 116 (1994) 77–86.
[13] H. Guillard, J.M. Malé, R. Peyret, Adaptive spectral methods with application to mixing layer computation, J. Comput. Phys. 102 (1992) 114–127.
[14] H. Ceniceros, T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular problems, J. Comput. Phys. 172 (2001) 609–639.
[15] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, J. Comput. Phys. 171 (2001) 753–775.
[16] P. Yu, L. Chen, Q. Du, Applications of moving mesh methods to the Fourier spectral approximations of phase-field equations, DCDS, 2006, submitted for publication.
[17] W. Cao, W. Huang, R.D. Russell, A study of monitor functions for two dimensional adaptive mesh generation, SIAM J. Sci. Comput. 20 (1999) 1978–1994.
[18] A. Winslow, Numerical solution of the quasi-linear Poisson equation in a non-uniform triangle mesh, J. Comput. Phys. 1 (1967) 149–172.
[19] 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.
[20] E.A. Dorfi, L.O'C. Drury, Simple adaptive grids for 1D initial value problems, J. Comput. Phys. 69 (1987) 175–195.
[21] L.R. Petzold, Observations on an adaptive moving grid method for one-dimensional systems for partial differential equations, Appl. Numer. Math. 3 (1987) 347–360.
[22] K. Nakahashi, G.S. Deiwert, Three dimensional adaptive grid method, AIAA J. 24 (1986) 948–954.

[23] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, Phys. Rev. Lett. 80 (1998) 3308–3311.
[24] R.J. Braun, B.T. Murray, Adaptive phase-field computations of dendritic crystal growth, J. Cryst. Growth 174 (1997) 41–53.
[25] A. Karma, W. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, Phys. Rev. E 57 (1998) 4323–4349.
[26] D. Wang, X. Wang, A three-dimensional adaptive method based on the iterative grid redistribution, J. Comput. Phys. 199 (2004) 423–436.
[27] P. Yu, C. Liu, Q. Du, A hydrodynamic formulation of moving mesh methods for time dependent PDEs, preprint, 2005.