Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Computational and Applied Mathematics 190 (2006) 252–269
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

A simple moving mesh method for one- and two-dimensional phase-field equations

Zhijun Tan[a], Tao Tang[a], Zhengru Zhang[b,*]

[a] *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong*
[b] *School of Mathematical Sciences, Beijing Normal University, Beijing, PR China*

Received 21 October 2004; received in revised form 7 January 2005

Dedicated to Professor Roderick Wong on the occasion of his 60th birthday

Abstract

A simple moving mesh method is proposed for solving phase-field equations. The numerical strategy is based on the approach proposed in Li et al. [J. Comput. Phys. 170 (2001) 562–588] to separate the mesh-moving and PDE evolution. The phase-field equations are discretized by a finite-volume method, and the mesh-moving part is realized by solving the conventional Euler–Lagrange equations with the standard gradient-based monitors. Numerical results demonstrate the accuracy and effectiveness of the proposed algorithm.

MSC: 65M06; 35Q35; 80A22

Keywords: Phase-field equations; Computation of free boundaries; Moving mesh method; Finite volume method

1. Introduction

Numerical study of free boundaries can be grouped broadly into two categories. One is to solve sharp-interface problems in which one or more variables (or their derivatives) are typically discontinuous

* Corresponding author.
E-mail addresses: zjtan@math.hkbu.edu.hk (Z. Tan), ttang@math.hkbu.edu.hk (T. Tang), zrzhang@bnu.edu.cn (Z. Zhang).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.01.042

across an interface, and the other is to solve a system of parabolic equations in which the interface is specified by a level set of one of the variables. The later approach, also called phase field approach, has two appealing features: (i) a broad spectrum of distinct problems that can be studied by means of a single set of equations, and (ii) the interface in these problems does not need to be tracked explicitly.

Most numerical methods used to solve the phase-field equations have used stationary uniform meshes, see, e.g., [11,12,16,28]. However, it is important that the diffused interface is well resolved if the correct dynamics are to be reproduced. As the phase interface moves in time it is clear that an efficient numerical approach must involve some form of mesh adaptivity. There have been two approaches in doing this. One is to use the local mesh refinement method, i.e. *h*-method, see, e.g., [7,22–24]. The other is to use moving mesh methods which are simpler in coding and require less complicated data structures than the local mesh refinement methods. The moving mesh methods involve the solution of the underlying PDE in conjunction with a so-called moving mesh PDE for the mesh itself. The methods keep the total number of grid points unchanged, and can cluster more grid points to areas with singularities or large solution gradients, see, e.g., [5,25]. The moving mesh methods have been proven very useful for time-dependent problems with localized singularities. The basic idea of moving mesh method is to construct a transformation from a logical domain (or called computational domain) to the physical domain. A fixed mesh is given on the logical domain, and the transformation is realized by solving moving mesh PDEs or minimization problems for a mesh functional, see, e.g., [1,8,15,18,29]. Recently, Mackenzie and Robertson [19] have developed a simple moving mesh strategy for interface propagation problems. They also developed a moving mesh method for one-dimensional phase-change problems modeled by the phase-field equations [20]. The computational mesh is obtained by equidistributing a monitor function tailored for the functional variation of the phase field in the interfacial region. Existence and uniqueness of the discretized equations using a moving mesh are also established. Their numerical algorithms are relatively simple and are shown to be far more efficient than fixed grid methods. In a recent work, Becket, Mackenzie et al. [2] developed a moving finite element method for the solution of the two-dimensional phase-field equations.

The objective of this work is to develop a simple moving mesh method for solving one- and two-dimensional phase-field equations. Our approach is based on the strategy proposed in [17] by decoupling the mesh motion and the PDE evolution. This approach requires using an interpolation to transform the information from the old mesh to the new mesh. In 1D, at each time level we first solve the problem in the physical space. The mesh is then redistributed based on the resulting solutions, and a conservative type interpolation is used to transfer the solution information from the old mesh to the new mesh. In 2D, we first transform the governing equations into the computational domain by a local (time-independent) mapping. The mapping will be obtained by using the moving mesh approach, namely solving the Euler–Lagrange equations involving monitor functions. This approach allows fast solution solvers such as multi-grid methods to solve the resulting nonlinear system (the phase-field equations have to be solved using implicit schemes). Again, solution interpolations have to be used so that time-independent mappings at each time can be used.

This paper is organized as follows: In the next section we describe the phase-field models in 1D and the corresponding moving mesh methods. In Section 3, the 2D problems and the moving mesh methods will be described. Numerical results for 1D and 2D problems are included in Sections 2 and 3, respectively. Some conclusion remarks will be made in the final section.

2. One-dimensional phase-field problems

The modified Stefan model describes a heat conduction problem and the evolution of a sharp interface $\Gamma(t)$ within a domain $\Omega \subset \mathbf{R}^d$. The objective is to find a temperature field $u(x, t)$ and a co-dimension one interface $\Gamma(t) \subset \Omega$ that solves the nondimensionalized equations

$$\frac{\partial u}{\partial t} = K\Delta u, \quad x \in \Omega\backslash\Gamma(t), \tag{2.1a}$$

$$Lv = -K[\nabla u]_-^+, \quad x \in \Gamma(t), \tag{2.1b}$$

$$u = -\frac{\sigma}{\Delta s}(\kappa + \alpha v), \quad x \in \Gamma(t). \tag{2.1c}$$

Eqs. (2.1a) and (2.1b) describe the diffusion of heat within the domain and the release of latent heat across the phase-change interface. Here L is the latent heat per unit mass, K is the thermal diffusivity, v is the normal velocity of the interface, and $[\nabla u]_-^+$ is the jump in the normal component of the temperature gradient at the interface. Surface tension and undercooling effects are modeled by the extended Gibbs–Thompson relation (2.1c). The parameter σ is the surface tension, α denotes a kinetic undercooling coefficient, Δs is the entropy difference between the two phases (and will be set to 4 within the normalization we use), and κ is the sum of the principal curvatures at a point of the interface. The classical Stefan model is obtained from (2.1) by setting $\sigma = 0$.

Numerical approaches for solving (2.1) require some form of front tracking to determine the curvature on the interface. An alternative is to use a so-called diffuse interface model that implicitly defines the position of the interface. The phase-field equations are derived using the idea of a phase order parameter p and Landau–Ginzburg theory. A free energy functional $\mathbf{F}$ is constructed in terms of the phase order parameter and other thermodynamic variables. For example,

$$\mathbf{F}(p, u) = \int_\Omega \left[\frac{1}{2}\tau^2(\nabla p)^2 + f(p, u)\right] \mathrm{d}x, \tag{2.2}$$

where τ is a length scale and $f(p, u)$ is a free energy density. Various choices of the precise choice of f have been suggested, the most studied of which is the Caginalp potential [11]:

$$f(p, u) = \frac{1}{8a}(p^2 - 1)^2 - 2up. \tag{2.3}$$

Both parameters τ and a are length scales related to the macroscopic physics. In particular, the surface tension σ and the interfacial thickness ε are related by

$$\sigma = \tfrac{2}{3}\varepsilon/a = \tfrac{2}{3}\tau/\sqrt{a} \quad \text{and} \quad \varepsilon = \tau\sqrt{a}. \tag{2.4}$$

A kinetic equation for the phase field is obtained by requiring that $\mathbf{F}$ monotonically decreases in time. The simplest choice of this requirement leads to the phase-field equation

$$\alpha\tau^2\frac{\partial p}{\partial t} = -\frac{\delta \mathbf{F}}{\delta p}, \tag{2.5}$$

where $\alpha\tau^2$ is a relaxation time. Direct calculation from the above equation gives

$$\alpha\tau^2\frac{\partial p}{\partial t} = \tau^2\Delta p - \frac{1}{2a}(p^3 - p) + 2u. \tag{2.6}$$

The phase equation is conjoined with the heat equation, modified to take into account the liberation of latent heat by the inclusion of an appropriate source term:

$$\frac{\partial u}{\partial t} + \frac{L}{2}\frac{\partial p}{\partial t} = K\Delta u. \tag{2.7}$$

In the rest of this paper we will only consider the Caginalp free energy density, although the numerical approach proposed in this work can be applied to other model potentials formulations.

2.1. 1D PDE solvers

Let x and ξ denote the physical and computational coordinates, respectively, which are (without loss of generality) assumed to be in $[a, b]$ and $[0, 1]$, respectively. A one-to-one coordinate transformation is denoted by $x = x(\xi)$, $\xi \in [0, 1]$ together with $x(0) = a$, $x(1) = b$. It is assumed that we already have a partition $a = x_0 < x_1 < \cdots < x_N = b$ in the physical domain and denote

$$x_{j+\frac{1}{2}} = \tfrac{1}{2}(x_j + x_{j+1}), \quad I_{j+\frac{1}{2}} = [x_j, x_{j+1}],$$

$$u_{j+\frac{1}{2}} = \frac{1}{|I_{j+\frac{1}{2}}|}\int_{I_{j+\frac{1}{2}}} u(x)\,\mathrm{d}x, \quad p_{j+\frac{1}{2}} = \frac{1}{|I_{j+\frac{1}{2}}|}\int_{I_{j+\frac{1}{2}}} p(x)\,\mathrm{d}x.$$

Integrating Eqs. (2.6)–(2.7) over the cell $I_{j+\frac{1}{2}} \times [t^n, t^{n+1}]$ and using implicit time-discretization lead to the following numerical scheme:

$$\alpha\tau^2\frac{p^{n+1}_{j+\frac{1}{2}} - p^n_{j+\frac{1}{2}}}{\Delta t^n} = \frac{2\tau^2}{h^n_j}\left(\frac{p^{n+1}_{j+\frac{3}{2}} - p^{n+1}_{j+\frac{1}{2}}}{h^n_j + h^n_{j+1}} - \frac{p^{n+1}_{j+\frac{1}{2}} - p^{n+1}_{j-\frac{1}{2}}}{h^n_{j-1} + h^n_j}\right) - \frac{1}{2a}[(p^{n+1}_{j+\frac{1}{2}})^3 - p^{n+1}_{j+\frac{1}{2}}] + 2u^n_{j+\frac{1}{2}}, \tag{2.8a}$$

$$\frac{u^{n+1}_{j+\frac{1}{2}} - u^n_{j+\frac{1}{2}}}{\Delta t^n} + \frac{L}{2}\frac{p^{n+1}_{j+\frac{1}{2}} - p^n_{j+\frac{1}{2}}}{\Delta t^n} = \frac{2K}{h^n_j}\left(\frac{u^{n+1}_{j+\frac{3}{2}} - u^{n+1}_{j+\frac{1}{2}}}{h^n_j + h^n_{j+1}} - \frac{u^{n+1}_{j+\frac{1}{2}} - u^{n+1}_{j-\frac{1}{2}}}{h^n_{j-1} + h^n_j}\right), \tag{2.8b}$$

where $\Delta t^n = t^{n+1} - t^n$ and $h^n_j = x^n_{j+1} - x^n_j$. Note that the temperature term in (2.8a) are treated explicitly, whereas all other terms are treated implicitly. The resulting nonlinear equations will be solved by the Newton iteration method. If we denote

$$\mathbf{u}^n = (u^n_{\frac{1}{2}}, u^n_{\frac{3}{2}}, \ldots, u^n_{N-\frac{1}{2}})^{\mathrm{T}}, \quad \mathbf{p}^n = (p^n_{\frac{1}{2}}, p^n_{\frac{3}{2}}, \ldots, p^n_{N-\frac{1}{2}})^{\mathrm{T}}, \tag{2.9}$$

then (2.8a) can be written in the form

$$\mathsf{H}(\mathbf{p}^{n+1}) := A\mathbf{p}^{n+1} + \phi(\mathbf{p}^{n+1}) = 0, \tag{2.10}$$

where the tridiagonal matrix A is given by

$$A_{j,j-1} = -\frac{2\tau^2\Delta t^n}{h_j^n(h_{j-1}^n + h_j^n)}, \quad A_{j,j+1} = -\frac{2\tau^2\Delta t^n}{h_j^n(h_j^n + h_{j+1}^n)}, \tag{2.11a}$$

$$A_{j,j} = \alpha\tau^2 + \frac{2\tau^2\Delta t^n}{h_j^n}\left(\frac{1}{h_{j-1}^n + h_j^n} + \frac{1}{h_j^n + h_{j+1}^n}\right) - \frac{\Delta t^n}{2a} \tag{2.11b}$$

and the vector ϕ is defined by

$$\phi_j(\mathbf{p}^{n+1}) = \frac{\Delta t^n}{2a}(p_{j+\frac{1}{2}}^{n+1})^3 - \alpha\tau^2 p_{j+\frac{1}{2}}^n - 2\Delta t^n u_{j+\frac{1}{2}}^n. \tag{2.12}$$

The nonlinear system (2.10)–(2.12) is solved using Newton's method:

$$\left.\frac{\partial \mathsf{H}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}^{n+1,s}} (\mathbf{p}^{n+1,s+1} - \mathbf{p}^{n+1,s}) = -\mathsf{H}(\mathbf{p}^{n+1,s}),$$

where the initial guess is set to be $p_{j+\frac{1}{2}}^{n+1,0} = p_{j+\frac{1}{2}}^n$. For $s = 0, 1, \ldots$, we obtain a linear algebraic system whose jth equation ($1 \leqslant j \leqslant N-1$) is given by

$$\sum_{l=j-1}^{j+1} A_{jl} p_{l+\frac{1}{2}}^{n+1,s+1} + \frac{3\Delta t^n}{2a}(p_{j+\frac{1}{2}}^{n+1,s})^2 p_{j+\frac{1}{2}}^{n+1,s+1} = \frac{\Delta t^n}{a}(p_{j+\frac{1}{2}}^{n+1,s})^3 + \alpha\tau^2 p_{j+\frac{1}{2}}^n + 2\Delta t^n u_{j+\frac{1}{2}}^n.$$

Denoting $G = \partial \mathsf{H}/\partial \mathbf{p}|_{\mathbf{p}=\mathbf{p}^{n+1,s}}$, $Q = \partial \mathsf{H}/\partial \mathbf{p}|_{\mathbf{p}=\mathbf{p}^{n+1,s}} \cdot \mathbf{p}^{n+1,s} - \mathsf{H}(\mathbf{p}^{n+1,s})$, we can write the linear algebraic system in the following simple form:

$$G\mathbf{p}^{n+1,s+1} = \mathbf{Q}, \tag{2.13}$$

where the tridiagonal matrix G and the vector $\mathbf{Q}$ are given by

$$G_{j,j-1} = A_{j,j-1}, \quad G_{j,j} = A_{j,j} + \frac{3\Delta t^n}{2a}(p_{j+\frac{1}{2}}^{n+1,s})^2, \tag{2.14a}$$

$$G_{j,j+1} = A_{j,j+1}, \quad Q_j = \frac{\Delta t^n}{a}(p_{j+\frac{1}{2}}^{n+1,s})^3 + \alpha\tau^2 p_{j+\frac{1}{2}}^n + 2\Delta t^n u_{j+\frac{1}{2}}^n. \tag{2.14b}$$

The outer Newton iteration is performed until the tolerance of 10^{-12} is reached.

After obtaining $\mathbf{p}^{n+1}$ from (2.8a), we solve (2.8b) by solving the following linear system:

$$C\mathbf{u}^{n+1} = \mathbf{b}^{n+1}, \tag{2.15}$$

where the tridiagonal matrix C and the vector $\mathbf{b}$ are given by

$$C_{j,j-1} = -\frac{2K\Delta t^n}{h_j^n(h_{j-1}^n + h_j^n)}, \quad C_{j,j} = 1 + \frac{2K\Delta t^n}{h_j^n}\left(\frac{1}{h_{j-1}^n + h_j^n} + \frac{1}{h_j^n + h_{j+1}^n}\right), \tag{2.16a}$$

$$C_{j,j+1} = -\frac{2K\Delta t^n}{h_j^n(h_j^n + h_{j+1}^n)}, \quad b_j^{n+1} = u_{j+\frac{1}{2}}^n - \frac{L}{2}(p_{j+\frac{1}{2}}^{n+1} - p_{j+\frac{1}{2}}^n). \tag{2.16b}$$

2.2. Moving mesh methods

The basic idea of the moving mesh method is to *relocate* grid points in a mesh having a fixed number of nodes in such a way that the nodes remain concentrated in regions of rapid variation of the solution. The principle ingredient of the moving mesh methods is the so-called equidistribution principle proposed by de Boor [4]. In 1D, it involves selecting mesh points such that some measure of the solution such as arclength or computed error is equalized over each subinterval. This measure is often connected to an indicator function called monitor function. It is very important to choose a suitable monitor function, otherwise satisfactory adaptations cannot be obtained no matter how good a moving mesh algorithm is.

Having the numerical (2.8), we can advance the numerical solution one time step to $t = t_{n+1}$. Then the following strategy is used to do the grid restructuring [30]:

(a): Solve the mesh redistributing equation (a generalized Laplacian equation) by one Gauss–Seidel iteration, to get $\mathbf{x}^{(k),n}$;
(b): Interpolate the approximate solutions on the new grid $\mathbf{x}^{(k),n}$;
(c): Obtain a weighted average of the locally calculated monitor at each computational cell and the surrounding monitor values;
(d): The iteration procedure (a)–(c) on grid-motion and solution-interpolation is continued until there is no significant change in calculated new grids from one iteration to the next.

In 1D, the mesh generation equation in part (a) above is based on the standard equidistribution principle:

$$(\omega x_\xi)_\xi = 0, \quad \xi \in [0, 1], \tag{2.17}$$

where the function ω is the monitor function which in general depends on the underlying solution to be adapted and is an indicator of the degree of singularity. In this work, it is known that the phase solution p has large solution variation in very small regions of the solution domain. Therefore, it is natural to use p as the control function in the monitor function ω, see, e.g., (2.22).

Eq. (2.17) is solved in the computational domain [0, 1] with a uniform mesh. The solution-updating in part b above is given as follows. After obtaining the new mesh $\{\tilde{x}_j\}$ from (2.17), we need to update the numerical solution on the new points $\tilde{x}_{j+\frac{1}{2}} = (\tilde{x}_j + \tilde{x}_{j+1})/2$. We use a second-order interpolation formula proposed in [27], which is conservative and does not increase the total variation:

$$\Delta\tilde{x}_{j+\frac{1}{2}}\tilde{u}_{j+\frac{1}{2}} = \Delta x_{j+\frac{1}{2}}u_{j+\frac{1}{2}} - ((cu)_{j+1} - (cu)_j), \tag{2.18}$$

where $\Delta\tilde{x}_{j+\frac{1}{2}} = \tilde{x}_{j+1} - \tilde{x}_j$ and $c_j = x_j - \tilde{x}_j$. In the actual computation, the linear flux cu is approximated by an up-winding approach, see [27]:

$$(\widehat{cu})_j = \frac{c_j}{2}(u_{j+\frac{1}{2}} + u_{j-\frac{1}{2}}) - \frac{|c_j|}{2}\left(u_{j+\frac{1}{2}} - u_{j-\frac{1}{2}}\right). \tag{2.19}$$

Since $\tilde{x}_{j+\frac{1}{2}}$ (new mesh), $x_{j+\frac{1}{2}}$ (old mesh) and $u_{j+\frac{1}{2}}$ (numerical solution on the old mesh) are all available, the numerical solution is interpolated on the new grid via (2.18) and (2.19).

2.3. An 1D example

Consider a classical freezing problem in a semi-infinite plane. This allows us to compare the results using the phase-field model to the similarity solution of the classical Stefan model. Theoretically, the solution of the phase-field model has been shown to approach that of the Stefan model as $\tau \to 0$, $\sigma \to 0$, and α is fixed [10].

We briefly describe the 1D example, the details can be found in, e.g., [20]. The Neumann's solution of the classical Stefan problem has the following form

$$u^{(s)}(x,t) = \begin{cases} C_1 \dfrac{[\operatorname{erf}(\beta/2) - \operatorname{erf}(x/(2\sqrt{t+t_0}))]}{\operatorname{erf}(\beta/2)}, & x \leqslant s(t), \\ C_2 \dfrac{[\operatorname{erf}(\beta/2) - \operatorname{erf}(x/(2\sqrt{t+t_0}))]}{1-\operatorname{erf}(\beta/2)}, & x > s(t), \end{cases} \tag{2.20}$$

where t_0 is a starting time and C_1 and C_2 are constants. The position of the interface is given by

$$s(t) = \beta\sqrt{t+t_0},$$

where β is the solution of

$$\frac{2}{\sqrt{\pi}} e^{-\beta^2/4}[C_2/(1-\operatorname{erf}(\beta/2)) - C_1/\operatorname{erf}(\beta/2)] - \beta = 0.$$

Following [20], we choose $t_0 = 0.15$, $C_1 = -0.085$ and $C_2 = -0.015$, which correspond to $\beta = 0.396618$, and we choose $L = K = 1$. The initial and boundary conditions for the temperature are

$$u(x,0) = C_1, \quad u(0,t) = u^{(s)}(0,t), \quad \text{and} \quad u(1,t) = u^{(s)}(1,t).$$

The boundary conditions for the phase field are

$$p(0,t) = \min_p f(p, C_1), \quad \text{closest to } -1,$$

$$p(1,t) = \min_p f(p, u^{(s)}(1,t)), \quad \text{closest to } 1,$$

where $f(p,u)$ is the Caginalp potential (2.3). A smoothed initial value for the phase field is given by

$$p(x,0) = \begin{cases} p(0,0)\tanh\left(\dfrac{s(0)-x}{2\varepsilon}\right), & x \leqslant s(0), \\ p(1,0)\tanh\left(\dfrac{x-s(0)}{2\varepsilon}\right), & x > s(0). \end{cases} \tag{2.21}$$

Following [20], we use the parameters $a = 0.0625$, $\tau = 0.002$, and $\alpha = 1$. These values lead to an interface thickness $\varepsilon = 0.0005$ and a surface tension $\sigma = 0.00533$. This case corresponds to experiment 5 of Caginalp and Socolovsky [11] and is a stiff test for a fixed grid method. In fact, it can be demonstrated that an accurate solution cannot be computed with 1000 uniform gridpoints. This problem has been solved by several authors, including fixed mesh results of Caginalp and Socolovsky [11], the adaptive moving mesh results of McCarthy [21] who used the algorithm of Blom and Zegeling [3], and the moving mesh result of Mackenzie and Robertson [19,20].

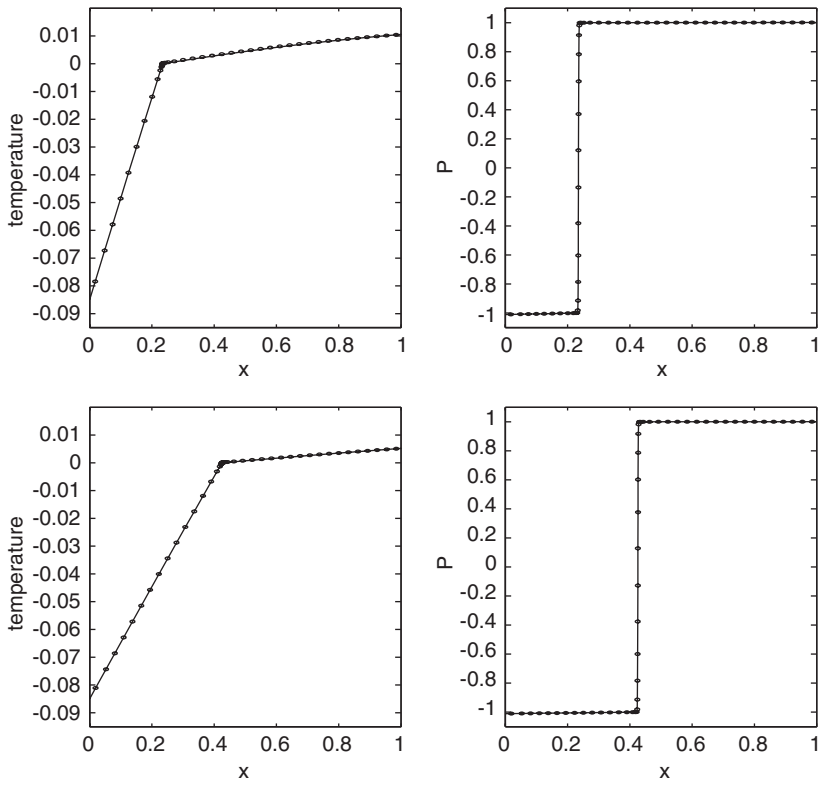

Fig. 1. The 1D example: temperature (left) and phase fields (right) at $t = 0.2$ and 1 (solid line denotes exact solution and circle denotes the moving mesh solution). $N = 50$.

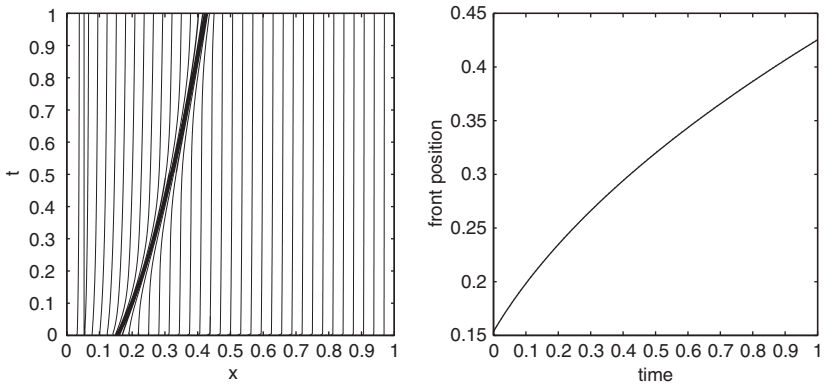

Fig. 2. the 1D example: mesh trajectories (left) and interface position (right). $N = 50$.

Fig. 1 shows the temperature profiles at $t = 0.2$ and 1 obtained by using 50 moving gridpoints. The numerical results are found satisfactory, as they are in good agreement with the previously published results, including the moving mesh results of Mackenzie and Robertson [19,20]. Fig. 2 shows the computed mesh trajectories and the interface position with $N=50$. It is observed that the mesh has followed smoothly the movement of the phase boundary and the clustering of mesh points has led to a very accurate prediction

of the position of the front. The monitor function used in the 1D computations is the standard gradient-based monitor

$$\omega = \sqrt{1 + 100 p_\xi^2}. \tag{2.22}$$

3. Two-dimensional phase-field problems

In this section, we mainly investigate the use of the moving mesh approach to solve the two-dimensional phase-field equations. Let $\Omega \subset \mathbf{R}^2$ be a bounded domain with a Lipschitz continuous boundary $\partial\Omega$. For each t we will assume we have a decomposition of Ω into subdomains $\Omega^+(t)$ and $\Omega^-(t)$ so that $\Omega = \Omega^+(t) \cup \Gamma(t) \cup \Omega^-(t)$, where the interface $\Gamma(t) = \partial\Omega^+(t) \cup \Omega^-(t)$ is smooth. Following [2], we consider the class of sharp interface problems in the following dimensionless form:

$$\theta_t = D\Delta\theta, \quad x \in \Omega^+(t) \cup \Omega^-(t), \tag{3.1a}$$

$$v = D[\nabla\theta \cdot n]_+^-, \quad x \in \Gamma(t), \tag{3.1b}$$

$$\theta = -d_0\kappa - \alpha d_0 v, \quad x \in \Gamma(t), \tag{3.1c}$$

where $\theta = c(T - T_m)/l$ is the dimensionless temperature (with T_m the equilibrium melting temperature), v is the normal velocity of the interface, D is a diffusion parameter, d_0 is a capillary length, and α is a kinetic undercooling coefficient. Using a scaling introduced in [9], we consider the corresponding phase-field model

$$\alpha\varepsilon^2 p_t = \varepsilon^2\Delta p + \frac{1}{2}(p - p^3) + \frac{\varepsilon}{3d_0}\theta, \tag{3.2a}$$

$$\theta_t + \tfrac{1}{2} p_t = D\Delta\theta, \tag{3.2b}$$

where ε is a measure of the diffuse interface thickness. The boundary conditions for the phase-field equations are the same as the sharp interface model for θ, with compatible conditions for p. For example, if Dirichlet conditions are imposed on $\theta = \theta_{\partial\pm}$, where $\pm$ denotes the liquid and solid boundaries respectively, then the corresponding values of p are the largest (p_+) and smallest (p_-) roots of

$$f(p, \theta) = \frac{1}{2}(p_\pm - p_\pm^3) + \frac{\varepsilon}{3d_0}\theta_{\partial\pm} = 0. \tag{3.3}$$

The above requirement ensures that there is no mass flow out of the system (see, e.g., [11,12,14]). Then the two phases are characterized by p taking values close to p_+ and p_- in each phase. To be more specific, consider the critical radius equilibrium example in Fig. 5. At the boundary of the square domain, we use $\theta_{\partial\pm} = -2$. The corresponding boundary conditions for p is $p = p_+$ if the boundary point is above the predicted interface and $p = p_-$ if it is below the interface.

It is noted that other phase-field models have been proposed that allow a simpler implementation of boundary conditions for p, see, e.g., Wang et al. [28].

3.1. 2D PDE solvers

Assume we have some simple mapping

$$x = x(\xi, \eta), \quad y = y(\xi, \eta) \quad \text{and} \quad \xi = \xi(x, y), \quad \eta = \eta(x, y), \tag{3.4}$$

where (x, y) and (ξ, η) are the physical and computational coordinates, respectively. We will apply the above coordinate transformation to Eq. (3.2). The cell-centered finite volume method will be employed to solve the transformed PDEs. The key point here is to obtain the transformations for Δp and $\Delta\theta$. Note that

$$\theta_{xx} = \frac{1}{J}[(J^{-1}y_\eta^2\theta_\xi)_\xi - (J^{-1}y_\xi y_\eta\theta_\eta)_\xi - (J^{-1}y_\xi y_\eta\theta_\xi)_\eta + (J^{-1}y_\xi^2\theta_\eta)_\eta], \tag{3.5}$$

$$\theta_{yy} = \frac{1}{J}[(J^{-1}x_\eta^2\theta_\xi)_\xi - (J^{-1}x_\xi x_\eta\theta_\eta)_\xi - (J^{-1}x_\xi x_\eta\theta_\xi)_\eta + (J^{-1}x_\xi^2\theta_\eta)_\eta], \tag{3.6}$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is the Jacobian of the coordinate transformation. The following symmetric discretizations will be used to approximate terms on the right-hand side of (3.5):

$$[(J^{-1}y_\eta^2\theta_\xi)_\xi]_{j,k} = (J^{-1}y_\eta^2)_{j+\frac{1}{2},k}(\theta_{j+1,k} - \theta_{j,k}) - (J^{-1}y_\eta^2)_{j-\frac{1}{2},k}(\theta_{j,k} - \theta_{j-1,k}), \tag{3.7a}$$

$$\begin{aligned}[-(J^{-1}y_\xi y_\eta\theta_\eta)_\xi]_{j,k} = & -\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j+1,k}(\theta_{j+1,k+1} - \theta_{j+1,k-1}) \\ & + \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j-1,k}(\theta_{j-1,k+1} - \theta_{j-1,k-1}),\end{aligned} \tag{3.7b}$$

$$\begin{aligned}[-(J^{-1}y_\xi y_\eta\theta_\xi)_\eta]_{j,k} = & -\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j,k+1}(\theta_{j+1,k+1} - \theta_{j-1,k+1}) \\ & + \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j,k-1}(\theta_{j+1,k-1} - \theta_{j-1,k-1}),\end{aligned} \tag{3.7c}$$

$$[(J^{-1}y_\xi^2\theta_\eta)_\eta]_{j,k} = (J^{-1}y_\xi^2)_{j,k+\frac{1}{2}}(\theta_{j,k+1} - \theta_{j,k}) - (J^{-1}y_\xi^2)_{j,k-\frac{1}{2}}(\theta_{j,k} - \theta_{j,k-1}), \tag{3.7d}$$

where for simplicity we have assumed that $\Delta\xi = \Delta\eta = 1$. The terms on the right-hand side of (3.6) can be approximated similarly. Having these approximations, we are able to approximate the Laplacian $\Delta\theta$ at (ξ_j, η_k) by

$$(\Delta\theta)_{j,k} = \sum_{l=-1}^{1}\sum_{m=-1}^{1} C_{jk}^{lm}\theta_{j+l,k+m}, \tag{3.8}$$

where

$$C_{jk}^{\pm1,-1} = \pm\tfrac{1}{4}[(J^{-1}y_\xi y_\eta)_{j\pm1,k} + (J^{-1}y_\xi y_\eta)_{j,k-1} + (J^{-1}x_\xi x_\eta)_{j\pm1,k} + (J^{-1}x_\xi x_\eta)_{j,k-1}], \tag{3.9a}$$

$$C_{jk}^{0,\pm1} = (J^{-1}y_\xi^2)_{j,k\pm\frac{1}{2}} + (J^{-1}x_\xi^2)_{j,k\pm\frac{1}{2}}, \tag{3.9b}$$

$$C_{jk}^{\pm1,0} = (J^{-1}y_\eta^2)_{j\pm\frac{1}{2},k} + (J^{-1}x_\eta^2)_{j\pm\frac{1}{2},k}, \tag{3.9c}$$

$$\begin{aligned}C_{jk}^{0,0} = & -(J^{-1}y_\eta^2)_{j+\frac{1}{2},k} - (J^{-1}y_\eta^2)_{j-\frac{1}{2},k} - (J^{-1}y_\xi^2)_{j,k+\frac{1}{2}} - (J^{-1}y_\xi^2)_{j,k-\frac{1}{2}} \\ & - (J^{-1}x_\eta^2)_{j+\frac{1}{2},k} - (J^{-1}x_\eta^2)_{j-\frac{1}{2},k} - (J^{-1}x_\xi^2)_{j,k+\frac{1}{2}} - (J^{-1}x_\xi^2)_{j,k-\frac{1}{2}},\end{aligned} \tag{3.9d}$$

$$C_{jk}^{-1,1} = \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j-1,k} + \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j,k+1} + \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{j-1,k} + \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{j,k+1}, \quad (3.9e)$$

$$C_{jk}^{1,1} = -\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j,k+1} - \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{j+1,k} - \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{j+1,k} - \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{j,k+1}. \quad (3.9f)$$

We will solve (3.2) by a finite-volume approach. Denote the control cell $[\xi_{j-\frac{1}{2}}, \xi_{j+\frac{1}{2}}] \times [\eta_{k-\frac{1}{2}}, \eta_{k+\frac{1}{2}}]$ by $A_{j,k}$ and the cell average values by

$$\bar{p}_{j,k}^n = \frac{1}{\Delta\xi\Delta\eta}\int_{A_{j,k}} p(\xi,\eta,t^n)\,\mathrm{d}\xi\,\mathrm{d}\eta, \quad \bar{\theta}_{j,k}^n = \frac{1}{\Delta\xi\Delta\eta}\int_{A_{j,k}} \theta(\xi,\eta,t^n)\,\mathrm{d}\xi\,\mathrm{d}\eta.$$

For ease of notations, below we will drop the top bar for $\bar{p}_{j,k}$ and $\bar{\theta}_{j,k}$, respectively. As done before, we will discretize the temperature term explicitly, whereas treat all other terms implicitly. This gives

$$\begin{cases} \alpha\varepsilon^2\dfrac{p_{jk}^{n+1}-p_{jk}^n}{\Delta t^n} = \dfrac{\varepsilon^2}{J}\displaystyle\sum_{l=-1}^{1}\sum_{m=-1}^{1} C_{jk}^{lm}\,p_{j+l,k+m}^{n+1} + \frac{1}{2}[p_{jk}^{n+1} - (p_{jk}^{n+1})^3] + \frac{\varepsilon}{3d_0}\theta_{jk}^n, \\ \dfrac{\theta_{jk}^{n+1}-\theta_{jk}^n}{\Delta t^n} + \dfrac{1}{2}\dfrac{p_{jk}^{n+1}-p_{jk}^n}{\Delta t^n} = \dfrac{D}{J}\displaystyle\sum_{l=-1}^{1}\sum_{m=-1}^{1} C_{jk}^{lm}\theta_{j+l,k+m}^{n+1}. \end{cases} \quad (3.10)$$

Rearranging the above scheme gives

$$\begin{aligned} &\varepsilon^2\left(J\alpha p_{jk}^{n+1} - \Delta t^n\sum_{l=-1}^{1}\sum_{m=-1}^{1} C_{jk}^{lm}\,p_{j+l,k+m}^{n+1}\right) + \frac{J\Delta t^n}{2}[(p_{jk}^{n+1})^3 - p_{jk}^{n+1}] \\ &\quad = J\left(\alpha\varepsilon^2 p_{jk}^n + \frac{\varepsilon\Delta t^n}{3d_0}\theta_{jk}^n\right), \end{aligned} \quad (3.11)$$

$$J\theta_{jk}^{n+1} - D\Delta t^n\sum_{l=-1}^{1}\sum_{m=-1}^{1} C_{jk}^{lm}\theta_{j+l,k+m}^{n+1} = J\left[\theta_{jk}^n - \frac{1}{2}(p_{jk}^{n+1} - p_{jk}^n)\right]. \quad (3.12)$$

Similar to the 1D case, we use Newton's method to solve (3.11):

$$\left.\frac{\partial \mathsf{H}}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{p}^{n+1,s}} (\mathbf{p}^{n+1,s+1} - \mathbf{p}^{n+1,s}) = -\mathsf{H}(\mathbf{p}^{n+1,s}),$$

where the initial guess is taken as $p_{jk}^{n+1,0} = p_{jk}^n$. For $s = 0, 1, \ldots,$ we obtain a linear system, whose jth equation $(1 \leqslant j \leqslant N-1)$ is given by

$$\begin{aligned} &\varepsilon^2\left(J\alpha\cdot p_{jk}^{n+1,s+1} - \Delta t^n\sum_{l=-1}^{1}\sum_{m=-1}^{1} C_{jk}^{lm}\cdot p_{j+l,k+m}^{n+1,s+1}\right) + \frac{J\Delta t^n}{2}[3(p_{jk}^{n+1,s})^2 - 1]\cdot p_{jk}^{n+1,s+1} \\ &\quad = J\Delta t^n(p_{jk}^{n+1,s})^3 + J\left(\alpha\varepsilon^2 p_{jk}^n + \frac{\varepsilon\Delta t^n}{3d_0}\theta_{jk}^n\right). \end{aligned}$$

Denoting $G = \partial\mathsf{H}/\partial\mathbf{p}|_{\mathbf{p}=\mathbf{p}^{n+1,s}}$, $Q = (\partial\mathsf{H}/\partial\mathbf{p})_{\mathbf{p}=\mathbf{p}^{n+1,s}}\cdot\mathbf{p}^{n+1,s} - \mathsf{H}(\mathbf{p}^{n+1,s})$, we can write the above linear system in the following form:

$$G\mathbf{p}^{n+1,s+1} = \mathbf{Q}, \quad (3.13)$$

where

$$Q_{jk} = J\Delta t^n (p_{jk}^{n+1,s})^3 + J\left(\alpha\varepsilon^2 p_{jk}^n + \frac{\varepsilon\Delta t^n}{3d_0}\theta_{jk}^n\right).$$

We use the above Newton's method to obtain the solution $\mathbf{p}^{n+1}$ for (3.11). Then we find the solution $\boldsymbol{\theta}^{n+1}$ of (3.12) by solving the linear system of the following form:

$$\mathrm{A}\boldsymbol{\theta}^{n+1} = \mathbf{b}^{n+1} \tag{3.14}$$

where

$$b_{jk}^{n+1} = J[\theta_{jk}^n - \tfrac{1}{2}(p_{jk}^{n+1} - p_{jk}^n)].$$

The above discretization for phase-field equations leads to two large sparse linear systems, namely (3.13) and (3.14), which are solved by using the Conjugate Gradient iteration method.

The 2D moving mesh strategy is similar to the one described in Section 2.1. The mesh generation equations are given by

$$\tilde{\nabla}\cdot(\omega\tilde{\nabla}x) = 0, \quad \tilde{\nabla}\cdot(\omega\tilde{\nabla}y) = 0 \tag{3.15}$$

where ω is a scalar monitor function, and $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^{\mathrm{T}}$. Again, the phase parameter p is chosen to be the control function in the monitor function due to its large variation behavior, see (3.19).

In the interpolation part, we use the 2D version of the conservative interpolation proposed in [27]:

$$\begin{aligned}|\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{u}_{j+\frac{1}{2},k+\frac{1}{2}} = |A_{j+\frac{1}{2},k+\frac{1}{2}}|u_{j+\frac{1}{2},k+\frac{1}{2}} &- [(c^x u)_{j+1,k+\frac{1}{2}} - (c^x u)_{j,k+\frac{1}{2}}]\\ &- [(c^y u)_{j+\frac{1}{2},k+1} - (c^y u)_{j+\frac{1}{2},k}]\end{aligned} \tag{3.16}$$

where $c_{j,k}^x = x_{j,k} - \tilde{x}_{j,k}$, $c_{j,k}^y = y_{j,k} - \tilde{y}_{j,k}$.

In order to obtain smoother transitions in the mesh, rather than merely using Eqs. (3.15), an additional filter is applied to the monitor functions. Instead of working with ω_{ij}, the smoothed values

$$\begin{aligned}\bar{\omega}_{i,j} \leftarrow{}& \tfrac{4}{16}\omega_{i,j} + \tfrac{2}{16}(\omega_{i+1,j} + \omega_{i-1,j} + \omega_{i,j+1} + \omega_{i,j-1})\\ &+ \tfrac{1}{16}(\omega_{i-1,j-1} + \omega_{i-1,j+1} + \omega_{i+1,j-1} + \omega_{i+1,j+1})\end{aligned} \tag{3.17}$$

are being used in the mesh equations.

3.2. 2D numerical examples

Planar solidification: A popular test case is to reproduce the traveling wave solution of the growth of a solid planar interface within an undercooled melt. This problem has also been considered in [2,6,12,16]. The sharp interface equations are solved with boundary conditions

$$\theta(\infty, y) = \theta_{\mathrm{cool}}, \quad \theta(-\infty, y) = \theta_{\mathrm{cool}} + 1.$$

It can be shown that there exists a traveling wave solution which takes the form

$$\theta(x, y, t) = \begin{cases}\theta_{\mathrm{cool}} + \mathrm{e}^{-v^*(x-v^*)/D}, & x > v^*t,\\ \theta_{\mathrm{cool}} + 1, & x \leqslant v^*t,\end{cases} \tag{3.18}$$

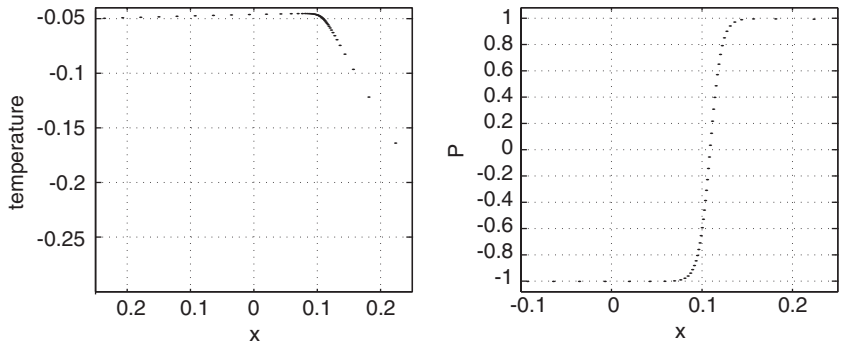

Fig. 3. Planar solidification: temperature and phase variable profiles with $\varepsilon = 0.00625$ and $d_0 = 0.05$ at $t = 0.1$.

where the shock movement velocity v^* is defined by

$$v^* = -\frac{1}{\alpha d_0}(\theta_{\text{cool}} + 1).$$

Fig. 3 shows the computed profiles for θ and p along the bottom boundary $y = -0.25$ at $t = 0.1$, obtained by using $N = 50$. Again, these are in good agreement with the moving mesh results of Beckett et al. [2]. Meshes and the corresponding front prediction at the times $t = 0$, 0.04, and $t = 0.1$ are plotted in Fig. 4. It is observed that the meshes are clustered in the neighborhood of the phase boundary. The monitor function used in (3.15) is

$$\omega = \sqrt{1 + 100|\tilde{\nabla} p|^2}. \tag{3.19}$$

This 2D example only has one-dimensional phenomena. However, it can be used to test the quality of the moving mesh algorithms. The numerical results show the straight gridlines in both x- and y-directions, which is expected theoretically.

Critical radius equilibrium: Consider a domain Ω which has no heat flux into it and within this domain the initial temperature is equal to a constant, θ_{cool}. Let us introduce an initial ball of solid of radius R_0 lying inside the undercooled liquid. It is well know that there exists a steady state solution of (3.1) where the solid ball is in equilibrium with its melt [13]. This occurs when the radius of the solid ball, R_c, is given by

$$R_c = -\frac{d_0}{\theta_{\text{cool}}}.$$

This equilibrium is unstable in that if $R_0 < R_c$ then the solid sphere will melt and the radius will decrease to zero. On the other hand if $R_0 > R_c$ then the solid will expand into the undercooled melt and the radius will increase.

Following [2], we take the initial temperature to be $\theta_{\text{cool}} = -2$ and $d_0 = 0.5$. This yields $R_c = \frac{1}{4}$. The phase-field calculations are performed with $\varepsilon = 1/(160\sqrt{2})$. The initial phase profile is given by

$$p(x, 0) = p_{bc} \tanh\left(\frac{r(\mathbf{x})}{2\varepsilon}\right),$$

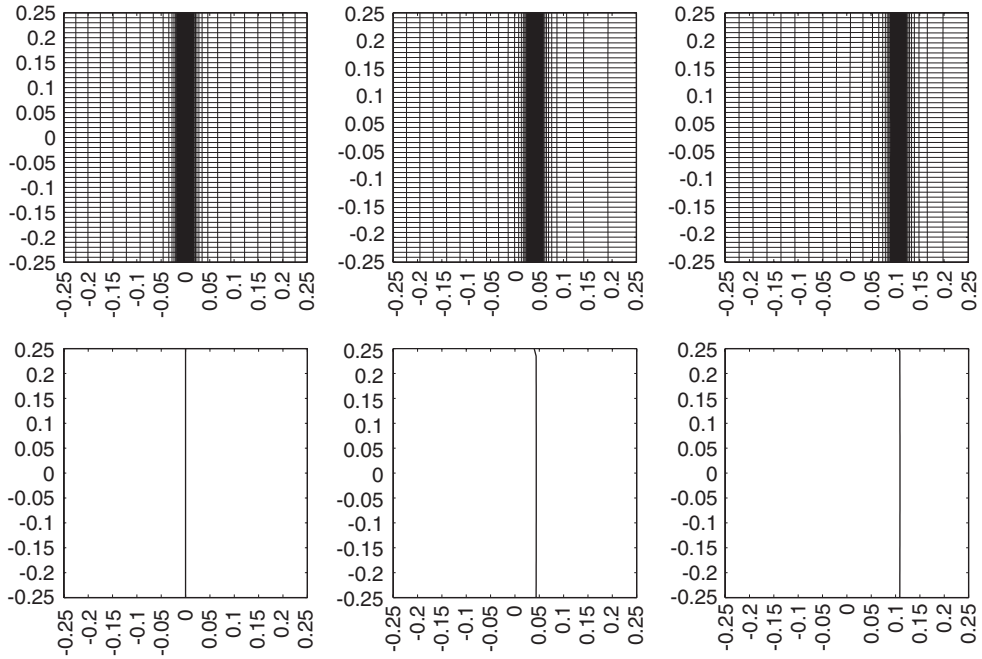

Fig. 4. Planar solidification: grid and interface predictions with $\varepsilon = 0.00625$ and $d_0 = 0.05$, at $t = 0$ (left), $t = 0.04$ (middle) and $t = 0.1$ (right).

where

$$p_{bc} = \begin{cases} -\min_p f(p, \theta_{\text{cool}}), \text{ closest to } -1, & r(\mathbf{x}) < 0, \\ \min_p f(p, \theta_{\text{cool}}), \text{ closest to } 1, & r(\mathbf{x}) \geqslant 0, \end{cases} \tag{3.20}$$

and $f(p, \theta)$ is given by (3.3). We consider one case where the initial radius is R_0=0.24, which corresponds to the unstable case of $R_0 < R_c$, in the domain $[0, 0.5]^2$.

Figs. 5 and 6 show the grids, contour of solution and front positions at times $t = 0.02$, 0.08, 0.14, and $t = 0.16$, obtained by using $N = 60$. As expected, the interface moves inwards with decreasing radius. The front positions agree very well with the moving mesh results of Beckett et al. [2] and the uniform mesh results of Elliott and Gardiner [14] who used a uniform 128^2 mesh. The monitor function used in (3.15) is again of the form (3.19).

We now briefly discuss the nonlinear iterations used in solving the system (2.10)–(3.11) or (3.11). In general, the number of Newton iterations depends on the size of the mesh points and numerical schemes used (i.e. uniform or adaptive). In the above computation (with a 60×60 grid), it takes 4–5 iterations with moving mesh, and 2–3 iterations with uniform mesh. However, with a 300×300 grid, it takes 4–5 Newton iterations for both moving mesh and uniform mesh computations.

We close this section by discussing time steps used in our computations. On the moving mesh methods for evolution, the selection of time step size is always an issue. For the 1D computations reported in the

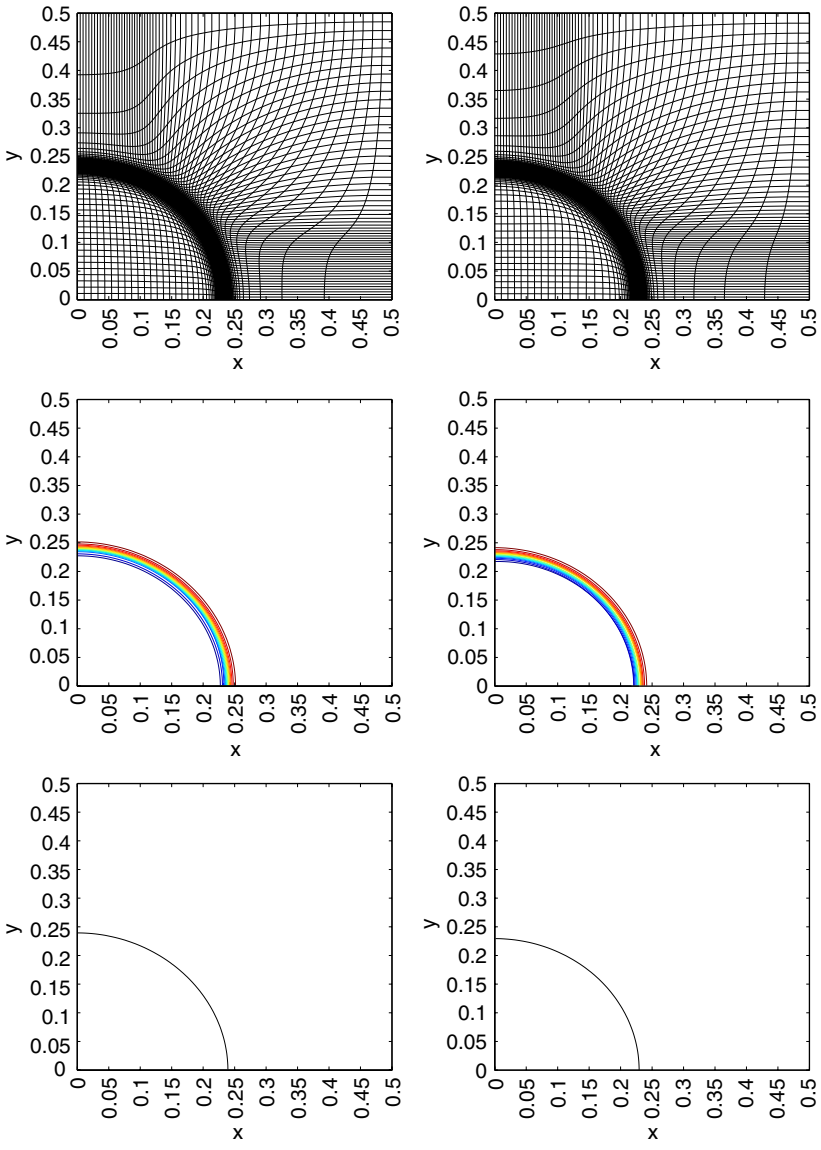

Fig. 5. Critical radius equilibrium: grid (top), solution (middle) and interface prediction (bottom) with $\varepsilon = 1/(160\sqrt{2})$, $d_0 = 0.5$ and $R_0 = 0.24$. Left: $t = 0.02$; right: $t = 0.08$.

paper, the time step used is 10^{-3}. In 2D, the time step for the planar solidification problem is 10^{-3} and for the critical radius equilibrium problem is 10^{-4}. By comparing with the uniform mesh approach, to reach the same resolutions the moving mesh method only gains in using less grid points in space but does not gain anything in time stepping. To increase the efficiency in time stepping, proper local time stepping techniques may be used [26].

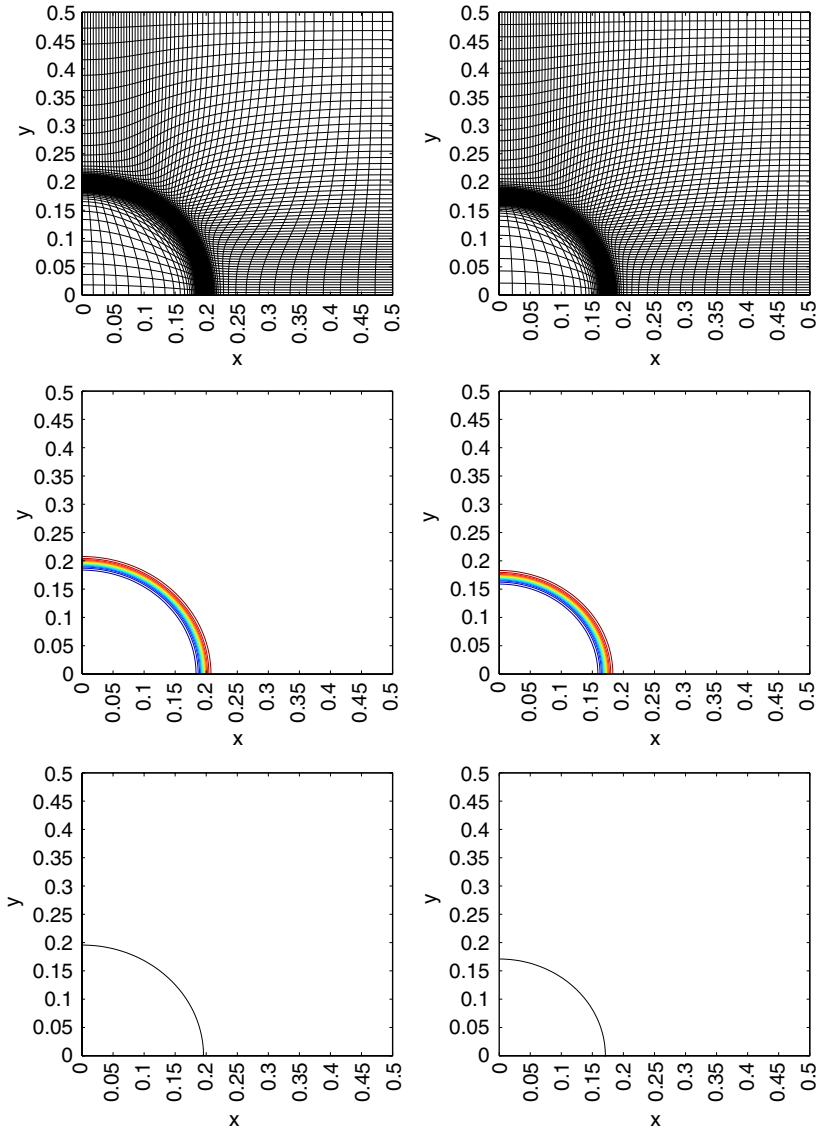

Fig. 6. Same as Fig. 5, except with $t = 0.14$ (left) and $t = 0.16$ (right).

4. Conclusions

In this paper, we have presented a simple moving mesh strategy for solving the phase-field equations. The method gives rise to smooth mesh trajectories and results in significant efficiency savings over uniform mesh methods. The numerical results are in good agreement with the recent moving mesh computations of Mackenzie et al. [2,19,20], in terms not only of accuracy but also efficiency. Our approach is based on the strategy proposed in [17] to separate the mesh moving and PDE evolution. This approach requires using an interpolation to transform the information from the old mesh to the new mesh. In this work, we

have used the second-order conservative interpolation proposed in [27]. It is noted that the approaches of Mackenzie et al. used a monitor function tailored for the functional variation of the phase field in the interfacial region. However, our approach does not require special attention to the choice of the monitor function, which simply used the standard gradient-based monitors.

Acknowledgements

The authors are grateful to Dr. John Mackenzie for many helpful discussions. The research of the second author was supported by the Hong Kong Research Grants Council and the International Research Team on Complex System of Chinese Academy of Sciences.

References

[1] M.J. Baines, M.E. Hubbard, P.K. Jimack, A Lagrangian moving finite element method incorporating monitor functions, in: Y.Y. Lu, W.W. Sun, T. Tang (Eds.), Advances in Scientific Computing and Applications, Science Press, Beijings/New York, 2004, pp. 32–44.

[2] G. Beckett, J.A. Mackenzie, M.L. Robertson, An r-Adaptive Finite Element Method for the Solution of the Two-Dimensional Phase-Field Equations, Technical Report 32, Strathclyde University, Mathematics Department, 2001, submitted for publication.

[3] J.G. Blom, P.A. Zegeling, A moving-grid interface for systems of one-dimensional time dependent partial differential equations, ACM Trans. Math. Software 20 (1994) 194.

[4] C. de Boor, Good approximation by splines with variable knots II, in: Springer Lecture Notes Series, vol. 363, Springer, Berlin, 1973.

[5] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.

[6] R.J. Braun, G.B. McFaddenn, S.R. Coriell, Morphological instability in phase-field models of solidification, Phys. Rev. E 49 (1994) 4336–4352.

[7] R.J. Braun, B.T. Murray, Adaptive phase-field computations of dendritic growth, J. Crystal Growth 174 (1997) 41–53.

[8] W.M. Cao, W.Z. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.

[9] G. Caginalp, Mathematical models of phase boundaries, in: J. Ball (Ed.), Material Instability in Continuum Problems and Related Mathematical Problems, Oxford Science Publications, 1988, pp. 35–52.

[10] G. Caginalp, Stefan and Hele-Shaw type models as asymptotic limits of the phase-field equations, Phys. Rev. A 39 (1989) 5887–5896.

[11] G. Caginalp, E.A. Socolovsky, Computation of sharp phase boundaries by spreading: the planar and spherically symmetric cases, J. Comput. Phys. 95 (1991) 85–100.

[12] G. Caginalp, E.A. Socolovsky, Phase field computations of single-needle crystals, crystal growth, and motion by mean curvature, SIAM J. Sci. Comput. 15 (1994) 106–126.

[13] B. Chalmers, Principles of Solidification, Krieger, New York, 1977.

[14] C.M. Elliott, A.R. Gardiner, Phase field equations, in: Computational Techniques and Applications, CTAC93, World Scientific, Singapore, 1993.

[15] W.Z. Huang, R.D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, SIAM J. Sci. Comput. 20 (1999) 998–1015.

[16] A. Karma, W.-J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, Phys. Rev. E 57 (1998) 4323–4349.

[17] R. Li, T. Tang, P. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.

[18] R. Li, T. Tang, P. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys. 177 (2002) 365–393.
[19] J.A. Mackenzie, M.L. Robertson, The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method, J. Comput. Phys. 161 (2000) 537–557.
[20] J.A. 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.
[21] J.F. McCarthy, One-dimensional phase field models with adaptive grids, Trans. ASME 120 (1998) 956.
[22] K. Miller, R.N. Miller, Moving finite element methods I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[23] R.H. Nochetto, M. Paolini, C. Verdi, An adaptive finite element method for two-phase Stefan problems in two space dimensions. Part II: Implementation and numerical experiments, SIAM J. Sci. Comput. 12 (1991) 1207–1244.
[24] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, Phys. Rev. Lett. 80 (1998) 3308–3311.
[25] N. Provatas, N. Goldenfeld, J. Dantzig, Adaptive mesh refinement computation of solidification microstructures using dynamic data structures, J. Comput. Phys. 148 (1999) 265–290.
[26] Z.-J. Tan, Z.-R. Zhang, T. Tang, Y.-Q. Huang, Moving mesh methods with locally varying time steps, J. Comput. Phys. 200 (2004) 347–367.
[27] H.Z. Tang, T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.
[28] S.-L. Wang, R.F. Sekerka, A.A. Wheeler, T. Murray, S.R. Coriell, R.J. Braun, G.B. McFadden, Thermodynamically-consistent phase field-models for solidification, Physica D 69 (1993) 189–200.
[29] P.A. Zegeling, On resistive MHD models with adaptive moving meshes, J. Sci. Comput., to appear.
[30] Z.-R. Zhang, T. Tang, An adaptive mesh redistribution algorithm for convection-dominated problems, Commun. Pure Appl. Anal. 1 (2002) 341–357.