INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1673–1693
Published online 17 July 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1568

An alternating Crank–Nicolson method for the numerical solution of the phase-field equations using adaptive moving meshes

Zhijun Tan[1,*,†] and Yanghong Huang[2]

[1]*Singapore-MIT Alliance, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore*
[2]*Department of Mathematics, University of California, Los Angeles, CA 90095, U.S.A.*

SUMMARY

An alternating Crank–Nicolson method is proposed for the numerical solution of the phase-field equations on a dynamically adaptive grid, which automatically leads to two decoupled algebraic subsystems, one is linear and the other is semilinear. The moving mesh strategy is based on the approach proposed by Li *et al*. (*J. Comput. Phys.* 2001; **170**:562–588) to separate the mesh-moving and partial differential equation evolution. The phase-field equations are discretized by a finite volume method in space, and the mesh-moving part is realized by solving the conventional Euler–Lagrange equations with the standard gradient-based monitors. The algorithm is computationally efficient and has been successfully used in numerical simulations. Copyright © 2007 John Wiley & Sons, Ltd.

Received 23 March 2006; Revised 29 May 2007; Accepted 1 June 2007

KEY WORDS: phase-field equations; computation of free boundaries; moving mesh method; finite volume method; Crank–Nicolson 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 across an interface, see, e.g. [1–3]. 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, see, e.g. [4–9]. 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. [6, 8–10]. It is well known that it is very important that the diffused interface is

*Correspondence to: Zhijun Tan, Singapore-MIT Alliance, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore.
†E-mail: smatz@nus.edu.sg

well resolved. As the phase interface moves in time, considering the fact that the phase solution is almost invariably away from the interface region, it is natural to use adaptive meshes, rather than uniform meshes, to compute the solution. There have been two main approaches in doing this. One is to use the local mesh refinement method, i.e. h-method, see, e.g. [5, 11–13]. The adaptive mesh is generated by adding or removing grid points to achieve a desired level of accuracy, which allows a systematic error analysis. However, the local mesh refinement approach requires complicated data structures and technically complex methods/means to communicate information among different levels of refinement. The other is to use moving mesh methods as discussed in our paper, i.e. r-method, which requires less complicated data structures than the local mesh refinement methods. The algorithm includes two independent parts: mesh-redistribution and partial differential equation (PDE) evolution. The second part is independent of the first, which can be any of the standard codes for the given PDEs. In the mesh redistribution approach, the adaptive method can keep the total number of grid points unchanged, and cluster more grid points in areas with singularities or large solution gradients, see, e.g. [14–17]. In the past two decades, the moving mesh methods have been proven very useful for time-dependent problems with localized singularities, see, e.g. [15, 16, 18–22]. 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. [20, 23–26]. Computational cost of moving mesh methods can be efficiently minimized with locally varying time steps [16]. Recently, Mackenzie and Robertson [27] have put forward a simple moving mesh strategy for interface propagation problems. They have also developed a moving mesh method for one-dimensional phase-change problems modeled by the phase-field equations [28]. 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 another recent work, Beckett *et al*. [29] developed an r-adaptive finite element method for the solution of the two-dimensional phase-field equations. They used a semi-implicit method, that is, all diffusion and source terms are treated implicitly and the convective-like terms arising from the mesh movement are treated explicitly. Tan *et al*. [15] developed a simple moving mesh method for one- and two-dimensional phase-field equations and used the implicit Euler discretization in time. One should note that they used just only the first-order numerical scheme in time.

The main objective of this work is to extend the second-order accurate, alternating Crank–Nicolson method of Mu and Huang [30] to numerical computation of the phase-field equations on adaptive moving meshes. The main contribution of this paper is to present an effective alternating Crank–Nicolson method for the phase-field equations by decoupling automatically the phase-field model to two algebraic subsystems. One is a symmetric positive-definite linear system that can be easily solved by effective iteration such as the conjugate gradient method. The other is a semilinear system that can be solved by Newton iteration method. Our moving mesh approach is based on the strategy proposed in [22] by decoupling the mesh motion and the PDE evolution. This approach requires the use of an interpolation to transform the information from the old mesh to the new mesh. We first transform the governing equations into the computational domain by a local (time-independent) mapping. The mapping is obtained *via* 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

DOI: 10.1002/fld

equations are often solved using implicit schemes in order to advance the stability). Again, solution interpolations are employed so that time-independent mappings at each time can be utilized. Two numerical experiments are conducted to demonstrate the efficiency of the proposed algorithm.

This paper is organized as follows. In the next section we describe the phase-field models in two dimension. In Sections 3 and 4, two-dimensional PDE solvers and the corresponding moving mesh methods will be described, respectively. Numerical results for two-dimensional problems are included in Section 5. Some concluding remarks will be made in Section 6.

2. 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 that 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 $\Gamma(t) = \overline{\Omega^+(t)} \cap \overline{\Omega^-(t)}$ is smooth. We use $\Omega^+(t)$ and $\Omega^-(t)$ to express the fluid phase and the solid phase regions, respectively. Following [29], we consider the class of sharp interface problems in the following dimensionless form:

$$\theta_t = D\Delta\theta, \quad \mathbf{x} \in \Omega^+(t) \cup \Omega^-(t) \tag{1a}$$

$$v = D[\nabla\theta \cdot n]_+^-, \quad \mathbf{x} \in \Gamma(t) \tag{1b}$$

$$\theta = -d_0\kappa - \alpha d_0 v, \quad \mathbf{x} \in \Gamma(t) \tag{1c}$$

where $\theta = c(T - T_m)/l$ is the dimensionless temperature, v is the normal velocity of the interface, $D = \bar{K}/\rho c$ is a diffusion parameter, $d_0 = \sigma c/(l[s]_m)$ is a capillary length, κ is the sum of the principal curvatures, $[\nabla\theta \cdot n]_+^-$ is the jump in the normal component of the temperature (from solid to liquid), and α is a kinetic undercooling coefficient. Here c, T_m, l, $\bar{K}$, ρ, σ and $[s]_m$ are physical parameters representing the specific heat, the equilibrium melting temperature, the latent heat, the thermal conductivity, the density, the surface tension and the entropy difference between phases per unit volume ($[s]_m = 4$ in the normalization used here), respectively. Using a scaling introduced in [31], we derive the corresponding phase-field model from (1):

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

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

where ε is a measure of the diffuse interface thickness. The above phase-field equations are derived using the idea of a phase-order parameter p and Landau–Ginzburg theory. Equation (2b) is an energy balance equation, and the evolution equation for p, (2a), is obtained from

$$\alpha\tau^2 \frac{\partial p}{\partial t} = -\frac{\delta \mathbf{F}}{\delta p} \tag{3}$$

Here, $\mathbf{F}$ is a free energy functional and defined by

$$\mathbf{F}(p, T) = \int_\Omega \left[\frac{1}{2}\tau^2(\nabla p)^2 + h(p, T)\right] \mathrm{d}\mathbf{x} \tag{4}$$

DOI: 10.1002/fld

where $h(p, T)$ is a free energy density of the form

$$h(p, T) = \frac{1}{8a}(p^2 - 1)^2 - \frac{\varepsilon[s]_m}{3a\sigma}(T - T_m)p \tag{5}$$

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

$$\sigma = \tfrac{2}{3}\varepsilon/a = \tfrac{2}{3}\tau/\sqrt{a} \quad \text{and} \quad \varepsilon = \tau\sqrt{a} \tag{6}$$

In [32] Caginalp gave a diagram of limits of phase-field equations and showed that if $\varepsilon \to 0$ while all other parameters are held fixed, then the asymptotic solutions of (2) are, to leading order,

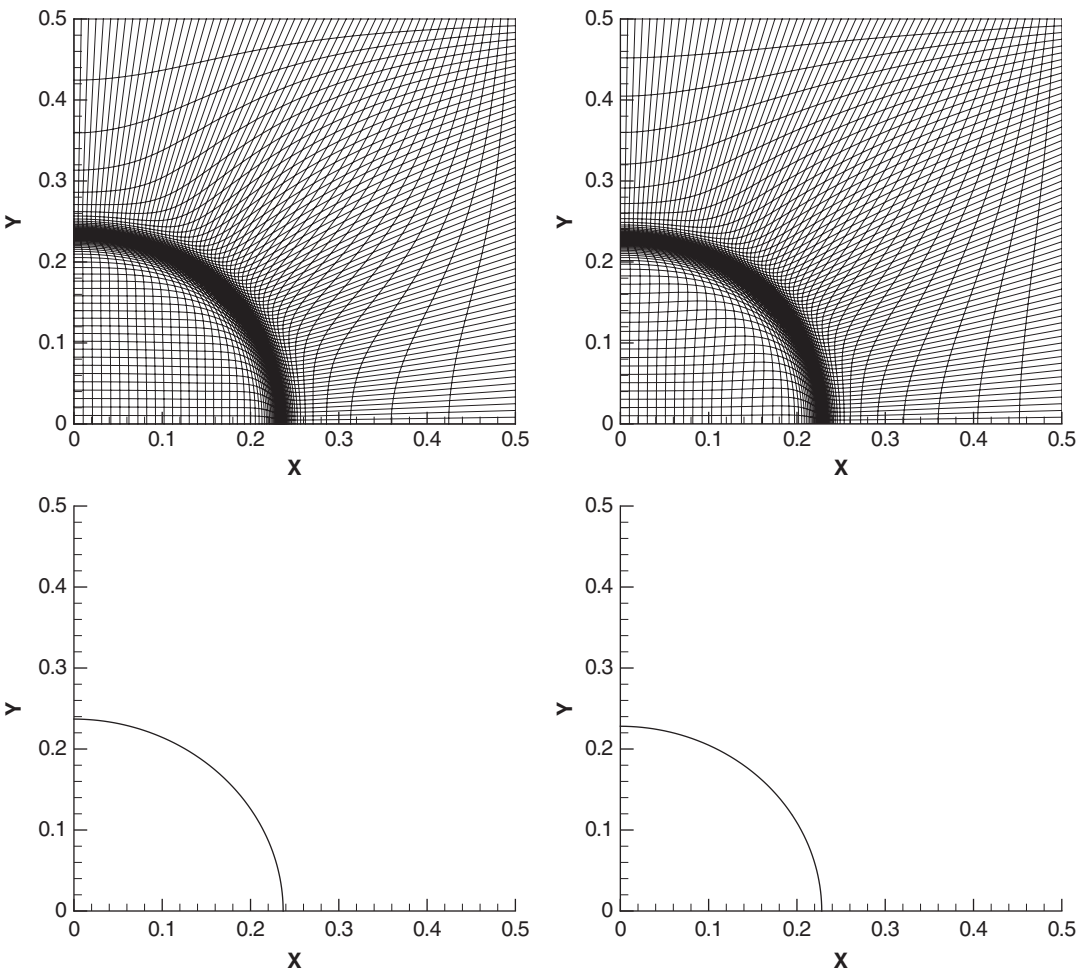

Figure 1. Critical radius equilibrium: grid (top) and interface prediction (bottom) with $\varepsilon = 1/(160\sqrt{2})$, $d_0 = 0.5$ and $R_0 = 0.24$. Left: $t = 0.04$ and right: $t = 0.08$.

DOI: 10.1002/fld

solutions of the modified Stefan problem (1). We refer the readers to [6, 10, 31, 32] and references therein for more details.

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 $\theta = \theta_{\pm}$, where $\pm$ denotes the liquid and solid boundaries, respectively, then the corresponding values of p are the largest (p_+) and smallest (p_-) roots of

$$f(p, \theta) = \frac{1}{2}(p_{\pm} - p_{\pm}^3) + \frac{\varepsilon}{3d_0}\theta_{\pm} = 0 \tag{7}$$

The above requirement ensures that there is no mass flow out of the system (see, e.g. [6, 7, 10]). 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 Figure 1. At the boundary of the square domain, we use $\theta_{\pm} = -2$. The corresponding boundary conditions for p are $p = p_+$ 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.* [9].

3. TWO-DIMENSIONAL PDE SOLVERS

Our moving mesh method is formed by two independent parts: PDE solver and the grid redistribution. The first part is discussed in this section, while the second part will be introduced in Section 4. To allow flexibility of handling complex geometry and of using fast solution solvers, we first transform the underlying PDEs (2) using the coordinate transform

$$x = x(\xi, \eta), \quad y = y(\xi, \eta) \text{ and } \xi = \xi(x, y), \quad \eta = \eta(x, y) \tag{8}$$

where (x, y) and (ξ, η) are the physical and computational coordinates, respectively, and then solve the resulting equations in the computational domain equipped with a (fixed) uniform mesh. In our computations, we restrict our attention to the structured quadrangular mesh, and the temperature θ and the phase p are defined at cell centers. The cell-centered finite volume method will be employed to solve the transformed PDEs. The ξ- and η-derivatives of x and y are approximated at the midpoints of the cell edges and the cell center points as follows:

$$(\mathscr{Z}_\xi)_{j+1/2,k} = \mathscr{Z}_{j+1,k} - \mathscr{Z}_{j,k}, \quad (\mathscr{Z}_\xi)_{j,k+1/2} = \tfrac{1}{4}(\mathscr{Z}_{j+1,k} + \mathscr{Z}_{j+1,k+1} - \mathscr{Z}_{j-1,k} - \mathscr{Z}_{j-1,k+1})$$

$$(\mathscr{Z}_\eta)_{j,k+1/2} = \mathscr{Z}_{j,k+1} - \mathscr{Z}_{j,k}, \quad (\mathscr{Z}_\eta)_{j+1/2,k} = \tfrac{1}{4}(\mathscr{Z}_{j,k+1} + \mathscr{Z}_{j+1,k+1} - \mathscr{Z}_{j,k-1} - \mathscr{Z}_{j+1,k-1})$$

$$(\mathscr{Z}_\xi)_{j+1/2,k+1/2} = \tfrac{1}{2}(\mathscr{Z}_{j+1,k} + \mathscr{Z}_{j+1,k+1} - \mathscr{Z}_{j,k} - \mathscr{Z}_{j,k+1})$$

$$(\mathscr{Z}_\eta)_{j+1/2,k+1/2} = \tfrac{1}{2}(\mathscr{Z}_{j,k+1} + \mathscr{Z}_{j+1,k+1} - \mathscr{Z}_{j,k} - \mathscr{Z}_{j+1,k}), \quad \mathscr{Z} = x \text{ or } y$$

DOI: 10.1002/fld

In order to obtain the transformed PDEs, the key point here is to obtain the transformations for Δp and $\Delta\theta$. Note that

$$W_{xx}=\frac{1}{J}[(J^{-1}y_\eta^2W_\xi)_\xi-(J^{-1}y_\xi y_\eta W_\eta)_\xi-(J^{-1}y_\xi y_\eta W_\xi)_\eta+(J^{-1}y_\xi^2W_\eta)_\eta] \tag{9}$$

$$W_{yy}=\frac{1}{J}[(J^{-1}x_\eta^2W_\xi)_\xi-(J^{-1}x_\xi x_\eta W_\eta)_\xi-(J^{-1}x_\xi x_\eta W_\xi)_\eta+(J^{-1}x_\xi^2W_\eta)_\eta] \tag{10}$$

where $J=x_\xi y_\eta-x_\eta y_\xi$ is the Jacobian of the coordinate transformation, and $W=p$ or θ. Denoting $\tilde{j}=j+1/2$ and $\tilde{k}=k+1/2$, the following symmetric discretizations at cell centers $(\xi_{j+1/2},\eta_{k+1/2})$ will be used to approximate terms on the right-hand side of (9):

$$\begin{aligned}[(J^{-1}y_\eta^2W_\xi)_\xi]_{\tilde{j},\tilde{k}}&=(J^{-1}y_\eta^2)_{\tilde{j}+1/2,\tilde{k}}(W_{\tilde{j}+1,\tilde{k}}-W_{\tilde{j},\tilde{k}})\\&\quad-(J^{-1}y_\eta^2)_{\tilde{j}-1/2,\tilde{k}}(W_{\tilde{j},\tilde{k}}-W_{\tilde{j}-1,\tilde{k}})\end{aligned} \tag{11a}$$

$$\begin{aligned}[-(J^{-1}y_\xi y_\eta W_\eta)_\xi]_{\tilde{j},\tilde{k}}&=-\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j}+1,\tilde{k}}(W_{\tilde{j}+1,\tilde{k}+1}-W_{\tilde{j}+1,\tilde{k}-1})\\&\quad+\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j}-1,\tilde{k}}(W_{\tilde{j}-1,\tilde{k}+1}-W_{\tilde{j}-1,\tilde{k}-1})\end{aligned} \tag{11b}$$

$$\begin{aligned}[-(J^{-1}y_\xi y_\eta W_\xi)_\eta]_{\tilde{j},\tilde{k}}&=-\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j},\tilde{k}+1}(W_{\tilde{j}+1,\tilde{k}+1}-W_{\tilde{j}-1,\tilde{k}+1})\\&\quad+\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j},\tilde{k}-1}(W_{\tilde{j}+1,\tilde{k}-1}-W_{\tilde{j}-1,\tilde{k}-1})\end{aligned} \tag{11c}$$

$$\begin{aligned}[(J^{-1}y_\xi^2W_\eta)_\eta]_{\tilde{j},\tilde{k}}&=(J^{-1}y_\xi^2)_{\tilde{j},\tilde{k}+1/2}(W_{\tilde{j},\tilde{k}+1}-W_{\tilde{j},\tilde{k}})\\&\quad-(J^{-1}y_\xi^2)_{\tilde{j},\tilde{k}-1/2}(W_{\tilde{j},\tilde{k}}-W_{\tilde{j},\tilde{k}-1})\end{aligned} \tag{11d}$$

where for simplicity we have assumed that $\Delta\xi=\Delta\eta=1$. The terms on the right-hand side of (10) can be approximated similarly. Having these approximations, we are able to approximate the Laplacian ΔW by

$$(\Delta_h W)_{\tilde{j},\tilde{k}}=\frac{1}{J_{\tilde{j},\tilde{k}}}\sum_{l=-1}^{1}\sum_{m=-1}^{1}C_{\tilde{j},\tilde{k}}^{lm}W_{\tilde{j}+l,\tilde{k}+m},\quad W=p\text{ or }\theta \tag{12}$$

where

$$C_{\tilde{j},\tilde{k}}^{\pm1,-1}=\pm\tfrac{1}{4}[(J^{-1}y_\xi y_\eta)_{\tilde{j}\pm1,\tilde{k}}+(J^{-1}y_\xi y_\eta)_{\tilde{j},\tilde{k}-1}+(J^{-1}x_\xi x_\eta)_{\tilde{j}\pm1,\tilde{k}}+(J^{-1}x_\xi x_\eta)_{\tilde{j},\tilde{k}-1}] \tag{13a}$$

$$C_{\tilde{j},\tilde{k}}^{0,\pm1}=(J^{-1}y_\xi^2)_{\tilde{j},\tilde{k}\pm1/2}+(J^{-1}x_\xi^2)_{\tilde{j},\tilde{k}\pm1/2} \tag{13b}$$

$$C_{\tilde{j},\tilde{k}}^{\pm1,0}=(J^{-1}y_\eta^2)_{\tilde{j}\pm1/2,\tilde{k}}+(J^{-1}x_\eta^2)_{\tilde{j}\pm1/2,\tilde{k}} \tag{13c}$$

DOI: 10.1002/fld

$$C^{0,0}_{\tilde{j},\tilde{k}} = -(J^{-1}y^2_\eta)_{\tilde{j}+1/2,\tilde{k}} - (J^{-1}y^2_\eta)_{\tilde{j}-1/2,\tilde{k}} - (J^{-1}y^2_\xi)_{\tilde{j},\tilde{k}+1/2} - (J^{-1}y^2_\xi)_{\tilde{j},\tilde{k}-1/2}$$
$$-(J^{-1}x^2_\eta)_{\tilde{j}+1/2,\tilde{k}} - (J^{-1}x^2_\eta)_{\tilde{j}-1/2,\tilde{k}} - (J^{-1}x^2_\xi)_{\tilde{j},\tilde{k}+1/2} - (J^{-1}x^2_\xi)_{\tilde{j},\tilde{k}-1/2} \quad (13d)$$

$$C^{-1,1}_{\tilde{j},\tilde{k}} = \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j}-1,\tilde{k}} + \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j},\tilde{k}+1} + \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{\tilde{j}-1,\tilde{k}} + \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{\tilde{j},\tilde{k}+1} \quad (13e)$$

$$C^{1,1}_{\tilde{j},\tilde{k}} = -\tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j},\tilde{k}+1} - \tfrac{1}{4}(J^{-1}y_\xi y_\eta)_{\tilde{j}+1,\tilde{k}} - \tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{\tilde{j}+1,\tilde{k}}$$
$$-\tfrac{1}{4}(J^{-1}x_\xi x_\eta)_{\tilde{j},\tilde{k}+1} \quad (13f)$$

Next we solve (2) by a finite volume approach. Denote the control cell $[\xi_j, \xi_{j+1}] \times [\eta_k, \eta_{k+1}]$ by $B_{j+1/2,k+1/2}$ and the cell average values by

$$\bar{p}^n_{\tilde{j},\tilde{k}} = \frac{1}{\Delta\xi\Delta\eta}\int_{B_{j+1/2,k+1/2}} p(\xi,\eta,t^n)\,\mathrm{d}\xi\,\mathrm{d}\eta, \quad \bar{\theta}^n_{\tilde{j},\tilde{k}} = \frac{1}{\Delta\xi\Delta\eta}\int_{B_{j+1/2,k+1/2}} \theta(\xi,\eta,t^n)\,\mathrm{d}\xi\,\mathrm{d}\eta$$

For ease of notations, below we will drop the top bar for $\bar{p}_{\tilde{j},\tilde{k}}$ and $\bar{\theta}_{\tilde{j},\tilde{k}}$, respectively. First, we assume that the time interval $[0, T]$ is partitioned by $0 = t^0 < t^1 < \cdots < t^{N_t} = T$ with a constant time stepping, i.e. $\Delta t = t^{n+1} - t^n$ is constant. Denoting $t^{n+1/2} = (t^{n+1} + t^n)/2$, we easily know $t^{n+1} = (t^{n+1/2} + t^{n+3/2})/2$ and $\Delta t = t^{n+3/2} - t^{n+1/2}$. We discretize the phase-field equations (2) in time using an alternating Crank–Nicolson finite difference method, i.e. we apply the Crank–Nicolson method to the first equation at $t^{n+1/2}$ and the second equation at t^{n+1} in (2), this gives

$$\alpha\varepsilon^2\frac{p^{n+1}_{\tilde{j},\tilde{k}} - p^n_{\tilde{j},\tilde{k}}}{\Delta t} = \varepsilon^2\frac{(\Delta_h p^n)_{\tilde{j},\tilde{k}} + (\Delta_h p^{n+1})_{\tilde{j},\tilde{k}}}{2} + \frac{1}{2}\overline{\overline{(p^{n+1/2}_{\tilde{j},\tilde{k}} - (p^{n+1/2}_{\tilde{j},\tilde{k}})^3)}}$$
$$+\frac{\varepsilon}{3d_0}\theta^{n+1/2}_{\tilde{j},\tilde{k}} \quad (14a)$$

$$\frac{\theta^{n+3/2}_{\tilde{j},\tilde{k}} - \theta^{n+1/2}_{\tilde{j},\tilde{k}}}{\Delta t} = D\frac{(\Delta_h\theta^{n+1/2})_{\tilde{j},\tilde{k}} + (\Delta_h\theta^{n+3/2})_{\tilde{j},\tilde{k}}}{2}$$
$$-\frac{1}{2}\frac{1}{\alpha\varepsilon^2}\left[\varepsilon^2(\Delta_h p^{n+1})_{\tilde{j},\tilde{k}} + \frac{1}{2}(p^{n+1}_{\tilde{j},\tilde{k}} - (p^{n+1}_{\tilde{j},\tilde{k}})^3)\right.$$
$$\left.+\frac{\varepsilon}{3d_0}\frac{\theta^{n+1/2}_{\tilde{j},\tilde{k}} + \theta^{n+3/2}_{\tilde{j},\tilde{k}}}{2}\right] \quad (14b)$$

For simplicity, the second term in the right-hand side of Equation (14a) is approximated by

$$\overline{\overline{(p^{n+1/2}_{\tilde{j},\tilde{k}} - (p^{n+1/2}_{\tilde{j},\tilde{k}})^3)}} = \tfrac{1}{2}(p^n_{\tilde{j},\tilde{k}} - (p^n_{\tilde{j},\tilde{k}})^3 + p^{n+1}_{\tilde{j},\tilde{k}} - (p^{n+1}_{\tilde{j},\tilde{k}})^3)$$

Given the temperature $\theta^{n+1/2}$ and the phase p^n at the time t^n, we can obtain p^{n+1} for solving the phase equation (14a). Note that in Equation (14b) p^{n+1} is already available from solving

DOI: 10.1002/fld

Equation (14a), then we can obtain $\theta^{n+3/2}$ for solving Equation (14b). Similarly, $\theta^{n+3/2}$ can be used in the next step $t^{n+3/2}$ for computing p^{n+2}. Therefore, the advantage of our alternating Crank–Nicolson method is that we automatically obtain two decoupled discrete subsystems by marching in time with p and θ alternately.

To start the time integration, one only needs to compute $\theta^{1/2}$ separately with the accuracy of $O(\Delta t)^2$, which can be done in many ways. In our computation, we can compute $\partial_t \theta(0)$ by

$$\partial_t \theta(0) = D\Delta_h \theta^0 - \frac{1}{2}\frac{1}{\alpha\varepsilon^2}\left[\varepsilon^2 \Delta p^0 + \frac{1}{2}(p^0 - (p^0)^3) + \frac{\varepsilon}{3d_0}u^0\right]$$

Substituting this in Taylor's expansion

$$\theta(t^{1/2}) = \theta(0) + \partial_t \theta(0)\Delta t/2 + O(\Delta t)^2$$

leads to an approximation to $\theta(t^{1/2})$ with second-order accuracy. An alternative is to use the predictor–corrector method for calculating $\theta(t^{1/2})$, as in [33].

The alternating approach described above solves a much smaller nonlinear system compared with larger system solved by Beckett *et al.* [29] and Tan *et al.* [15]. Equation (14a) is a small semilinear system, which is solved by Newton's method. At each time step, the initial guess for p^{n+1} is obtained from the previous time step, i.e. $p^{n+1,0} = p^n$. The resulting linear algebra system from each Newton iteration and the linear SPD systems from Equation (14b) are solved by using the conjugate gradient iteration method.

4. 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 principal ingredient of the moving mesh methods is the so-called equidistribution principle. In one dimension, 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.

With the numerical scheme (14), we can advance the numerical solution one time step to $t = t_{n+1}$. Then the following strategy is employed to carry out the grid restructuring [15]:

(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 calculating the new grid from one iteration to the next.

4.1. Mesh generation

The mesh is generated *via* variational approach. Let $\mathbf{x} = (x, y)$ and $\boldsymbol{\xi} = (\xi, \eta)$ denote the physical and computational coordinates. A coordinate mapping from the computational domain Ω_c to the

DOI: 10.1002/fld

physical domain Ω_p is given by

$$x = x(\xi, \eta), \quad y = y(\xi, \eta) \tag{15}$$

and the inverse map is

$$\xi = \xi(x, y), \quad \eta = \eta(x, y) \tag{16}$$

The specific map is obtained by minimizing of a mesh adaptation functional of the following form:

$$E[\xi, \eta] = \frac{1}{2} \int_{\Omega_p} (\nabla \xi^{\mathrm{T}} G_1^{-1} \nabla \xi + \nabla \eta^{\mathrm{T}} G_2^{-1} \nabla \eta) \, \mathrm{d}x \, \mathrm{d}y \tag{17}$$

where G_1 and G_2 are given symmetric positive-definite matrices called monitor functions. In general, monitor functions depend on the underlying solution to be adapted and its derivatives. More terms can be added to the functional (17) to control other aspects of the adaptive mesh such as orthogonality and mesh alignment with a given vector field [14, 34].

In this work, the adaptive mesh is determined by the corresponding Euler–Lagrange equations:

$$\nabla \cdot (G_1^{-1} \nabla \xi) = 0, \quad \nabla \cdot (G_2^{-1} \nabla \eta) = 0 \tag{18}$$

One of the simplest choices of monitor function is $G_1 = G_2 = \omega I$, where I is the identity matrix and ω is a positive weight function. One typical choice of the weight function is $\omega = \sqrt{1 + |\nabla u|^2}$, where u is a solution of the underlying PDEs. This choice of the monitor function corresponds to Winslows variable diffusion method [35]:

$$\nabla \cdot \left(\frac{1}{\omega} \nabla \xi \right) = 0, \quad \nabla \cdot \left(\frac{1}{\omega} \nabla \eta \right) = 0 \tag{19}$$

Equation (18) gives the coordinate transformation in mesh generation and adaptation. Grid generation is basically to obtain the curvilinear coordinate system (15) from the above elliptic system (18). Usually, after solving the system (18) for $\boldsymbol{\xi}(\mathbf{x})$, we find the inverse map to obtain $\mathbf{x}(\boldsymbol{\xi})$, which is expensive. Certainly, we can directly solve the corresponding equations on the computational domain Ω_c by interchanging the dependent and independent variables in (18). However, the obtained equations are complicated and massive computations are required. An alternative approach, as suggested by Ceniceros and Hou [36], is to consider a functional defined in the computational domain directly:

$$\tilde{E}[x, y] = \frac{1}{2} \int_{\Omega_c} (\tilde{\nabla}^{\mathrm{T}} x G_1 \tilde{\nabla} x + \tilde{\nabla}^{\mathrm{T}} y G_2 \tilde{\nabla} y) \, \mathrm{d}\xi \, \mathrm{d}\eta \tag{20}$$

to replace the convectional (17), where G_1 and G_2 are again the monitor functions and $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^{\mathrm{T}}$. The corresponding Euler–Lagrange equations are then of the form:

$$\tilde{\nabla} \cdot (G_1 \tilde{\nabla} x) = 0, \quad \tilde{\nabla} \cdot (G_2 \tilde{\nabla} y) = 0 \tag{21}$$

If we take the monitor function with the simplest form $G_1 = G_2 = \omega I$, then Equation (21) is reduced to

$$\tilde{\nabla} \cdot (\omega \tilde{\nabla} x) = 0, \quad \tilde{\nabla} \cdot (\omega \tilde{\nabla} y) = 0 \tag{22}$$

DOI: 10.1002/fld

In our computations, we use the Gauss–Seidel-type iteration method [17] to solve the mesh-moving Equation (22). For example, the iteration method for (22) is written as follows:

$$\alpha_{j+1/2,k}(\mathbf{x}_{j+1,k}^{[v]} - \mathbf{x}_{j,k}^{[v+1]}) - \alpha_{j-1/2,k}(\mathbf{x}_{j,k}^{[v+1]} - \mathbf{x}_{j-1,k}^{[v+1]}) + \beta_{j,k+1/2}(\mathbf{x}_{j,k+1}^{[v]} - \mathbf{x}_{j,k}^{[v+1]}) - \beta_{j,k-1/2}(\mathbf{x}_{j,k}^{[v+1]} - \mathbf{x}_{j,k-1}^{[v+1]}) = 0 \tag{23}$$

for $1 \leqslant j \leqslant N_\xi$ and $1 \leqslant k \leqslant N_\eta$, $v = 0, 1, \ldots,$ where N_ξ and N_η are the number of mesh points in the x- and y-direction, respectively, and

$$\alpha_{j+1/2,k} = \omega(u_{j\pm 1/2,k}^{[v]}) = \omega(\tfrac{1}{2}(u_{j\pm 1/2,k+1/2}^{[v]} + u_{j\pm 1/2,k-1/2}^{[v]}))$$

$$\beta_{j+1/2,k} = \omega(u_{j,k\pm 1/2}^{[v]}) = \omega(\tfrac{1}{2}(u_{j+1/2,k\pm 1/2}^{[v]} + u_{j-1/2,k\pm 1/2}^{[v]}))$$

The iteration is continued until there is no significant change in calculating new grids from one iteration to the next. In practice, a few iterations are required at each time level; hence, the cost for generating new mesh is not too expensive.

Let $\mathbf{x}_{j,k}$ and $\tilde{\mathbf{x}}_{j,k}$ be coordinates of the old and new grid points, respectively, and at the same time $A_{j+1/2,k+1/2}$ and $\tilde{A}_{j+1/2,k+1/2}$ denote the quadrangles with four vertices $(\mathbf{x}_{j+p,k+q})$ and $(\tilde{\mathbf{x}}_{j+p,k+q})$, $p, q \in \{0, 1\}$, respectively. After generating the new mesh at each iterative step according to the monitor function, we need to remap the approximate solutions onto the newly resulting mesh $\{\mathbf{x}_{j,k}\}$ from the old mesh $\{\tilde{\mathbf{x}}_{j,k}\}$. In our computations, the remapping procedure of the temperature θ and the phase p can be realized by using the conservative interpolation technique proposed by Tang and Tang [17], which is

$$|\tilde{A}_{j+1/2,k+1/2}|\tilde{U}_{j+1/2,k+1/2} = |A_{j+1/2,k+1/2}|U_{j+1/2,k+1/2} - [(c_{\mathbf{n}}^2 U)_{j+1,k+1/2} + (c_{\mathbf{n}}^4 U)_{j,k+1/2}] - [(c_{\mathbf{n}}^3 U)_{j+1/2,k+1} + (c_{\mathbf{n}}^1 U)_{j+1/2,k}] \tag{24}$$

where $U = \theta$ or p, and $c_{\mathbf{n}}^l := c^x n_x^l + c^y n_y^l$ with mesh velocity $(c^x, c^y) = (x - \tilde{x}, y - \tilde{y})$ and the unit outward normal direction $\mathbf{n}^l = (n_x^l, n_y^l)$ on the corresponding surface of the control volume $A_{j+1/2,k+1/2}$ for $l = 1, 2, 3, 4$. More detailed explanation can be found in [17]. The above formula is obtained using the classical perturbation theory. It is obvious that the discretization form (24) satisfies the mass conservation in the following discrete sense:

$$\sum_{j,k} |\tilde{A}_{j+1/2,k+1/2}|\tilde{U}_{j+1/2,k+1/2} = \sum_{j,k} |A_{j+1/2,k+1/2}|U_{j+1/2,k+1/2}, \quad U = \theta \text{ or } p$$

where $|A_{j+1/2,k+1/2}|$ and $|\tilde{A}_{j+1/2,k+1/2}|$ means the areas of the corresponding control cells. Some theoretical properties of this conservative interpolation can be found in [17]. Recently, Zhang [37] presented a new conservative interpolation, which may be more accurate and robust than the Tang and Tang's interpolation scheme [17]. However, it remains to be seen the implementation for higher dimensions.

In order to obtain smoother transitions in the mesh, rather than merely using equations (22), an additional filter is applied to the monitor functions. Instead of working with ω_{ij}, the smoothed

DOI: 10.1002/fld

values

$$\bar{\omega}_{i,j} \leftarrow \tfrac{4}{16}\omega_{i,j} + \tfrac{2}{16}(\omega_{i+1,j} + \omega_{i-1,j} + \omega_{i,j+1} + \omega_{i,j-1}) + \tfrac{1}{16}(\omega_{i-1,j-1} + \omega_{i-1,j+1} + \omega_{i+1,j-1} + \omega_{i+1,j+1}) \tag{25}$$

are being used in the mesh equations.

5. NUMERICAL EXAMPLES

In this section, two numerical experiments will be carried out to demonstrate the effectiveness of our algorithm proposed in this work.

5.1. Critical radius equilibrium

Consider a domain Ω that 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 known that there exists a steady-state solution of (1) where the solid ball is in equilibrium with its melt [38]. This occurs when the radius of the solid ball, R_c, is given by

$$R_c = -\frac{d_0}{\theta_{\text{cool}}} \tag{26}$$

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 [29], we take the initial temperature to be $\theta_{\text{cool}} = -2$ and $d_0 = 0.5$ with parameters $D = 1$ and $\alpha = 1$. The other physical parameters are chosen as $\rho = 1$, $c = 1$, $l = 1$, $\bar{K} = 1$, $\tau = 0.1151$, $a = 0.001473$ and $\sigma = 2.0$. It follows from (26) that $R_c = 0.25$. The phase-field calculations are performed with $\varepsilon = 1/(160\sqrt{2})$. Let $r(\mathbf{x})$ denote a signed normal distance from the point $\mathbf{x}$ to the interface Γ (i.e. $r(\mathbf{x})$ is the distance to the interface if it is in the liquid region and minus the distance if the point is in the solid region). The initial phase profile is given by

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

where

$$p_{bc} = \begin{cases} -\min\limits_{p} f(p, \theta_{\text{cool}}), \text{ closest to } -1, & r(\mathbf{x}) < 0 \\ \min\limits_{p} f(p, \theta_{\text{cool}}), \text{ closest to } 1, & r(\mathbf{x}) \geqslant 0 \end{cases} \tag{27}$$

and $f(p, \theta)$ is given by (7). We consider two cases where the initial radius is $R_0 = 0.24$ and $R_0 = 0.26$, which corresponds to the unstable case of $R_0 < R_c$ and $R_0 > R_c$, respectively, in the domain $[0, 0.5]^2$.

Figures 1 and 2 show the grids and front positions at times $t = 0.04, 0.08, 0.14$ and $t = 0.16$ when $R_0 = 0.24$, obtained by using a 60^2 moving grid. As expected, the interface moves inwards

DOI: 10.1002/fld

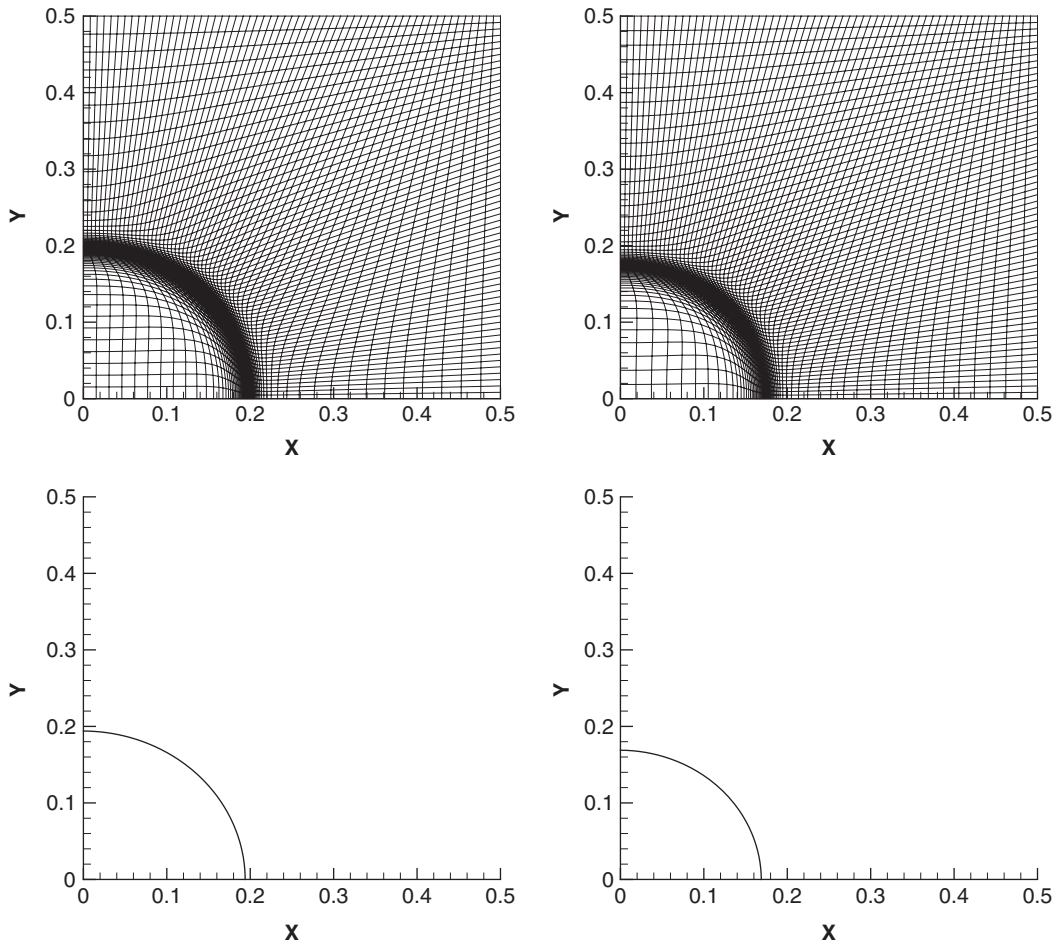

Figure 2. Same as Figure 1, except with $t = 0.14$ (left) and $t = 0.16$ (right).

with decreasing radius, and the interface radius *versus* time is given in Figure 3. The front positions agree very well with the moving mesh results of Beckett *et al*. [29] and the uniform mesh results of Elliott and Gardiner [7] who used a 128^2 uniform grid. For comparison, we also plot in Figure 3 the radial positions with uniform meshes, the dotted line denoting the solution using a 200^2 uniform grid and the dash dot line denoting the solution using a 300^2 uniform grid. The monitor function used in (22) is

$$\omega = \sqrt{1 + 100|\tilde{\nabla} p|^2} \tag{28}$$

There is no exact analytical solution for this problem. To test the accuracy, we use the result of the numerical solution computed with the same method on a 1024^2 uniform grid as the exact solution in the error analysis. Table I shows the computed l^1-errors and convergence rates at time $t = 0.1$, and the second-order accuracy can be obtained.

DOI: 10.1002/fld

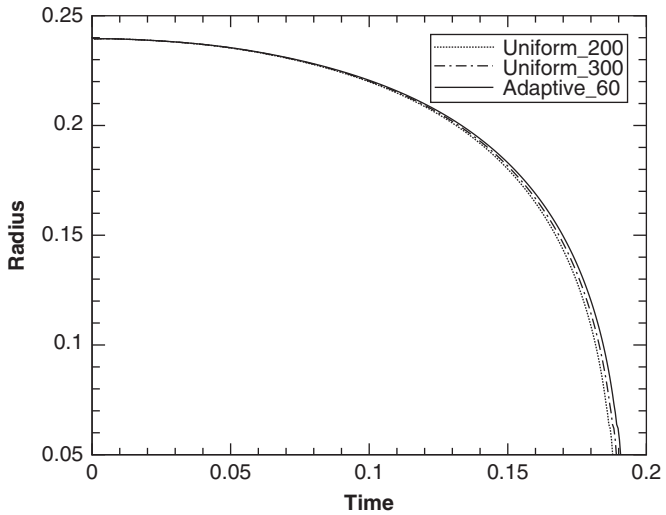

Figure 3. Critical radius equilibrium: the interface radius *versus* time. The solid line denotes the solution obtained using a 60^2 moving grid, the dotted line denotes the solution using a 200^2 uniform grid and the dash dot line denotes the solution using a 300^2 uniform grid.

Table I. Numerical errors and convergence rates for the critical radius equilibrium problem with $R_0 = 0.24$ at time $t = 0.1$.

N	l^1-error (θ)	θ-order	l^1-error (p)	p-order
30	3.172e−2	—	2.684e−2	—
60	8.825e−3	1.846	7.633e−3	1.814
120	2.341e−3	1.915	2.051e−3	1.896
240	5.958e−4	1.974	5.264e−4	1.962

θ and p represent the temperature and the phase, respectively.

For the case of $R_0 = 0.26$, the results obtained were also as expected, with the solid ball growing outwards and the supercooled liquid solidifying. The grids and front positions at times $t = 0.04, 0.08, 0.16$ and $t = 0.18$ are presented in Figures 4 and 5, the same number of grids and monitor function are taken in this case. In Figure 6, we present the radial positions for both interfaces, the solid line denotes the solution obtained with $R_0 = 0.24$ and the dashed denotes the solution with $R_0 = 0.26$.

5.2. Scaled viscous Chan–Hilliard problem

The viscous Cahn–Hilliard equation [4] arises from the phase-field model by the eliminating θ_t from (2b). Let us consider a circular domain Ω of radius R and that homogeneous Neumann data for θ is imposed on the fixed boundary $r = R$. Initially we have two circular interfaces with $R_I = 0.15$ and $R_O = 0.30$. If we assume radial symmetry then the sharp interface problem is equivalent to

DOI: 10.1002/fld

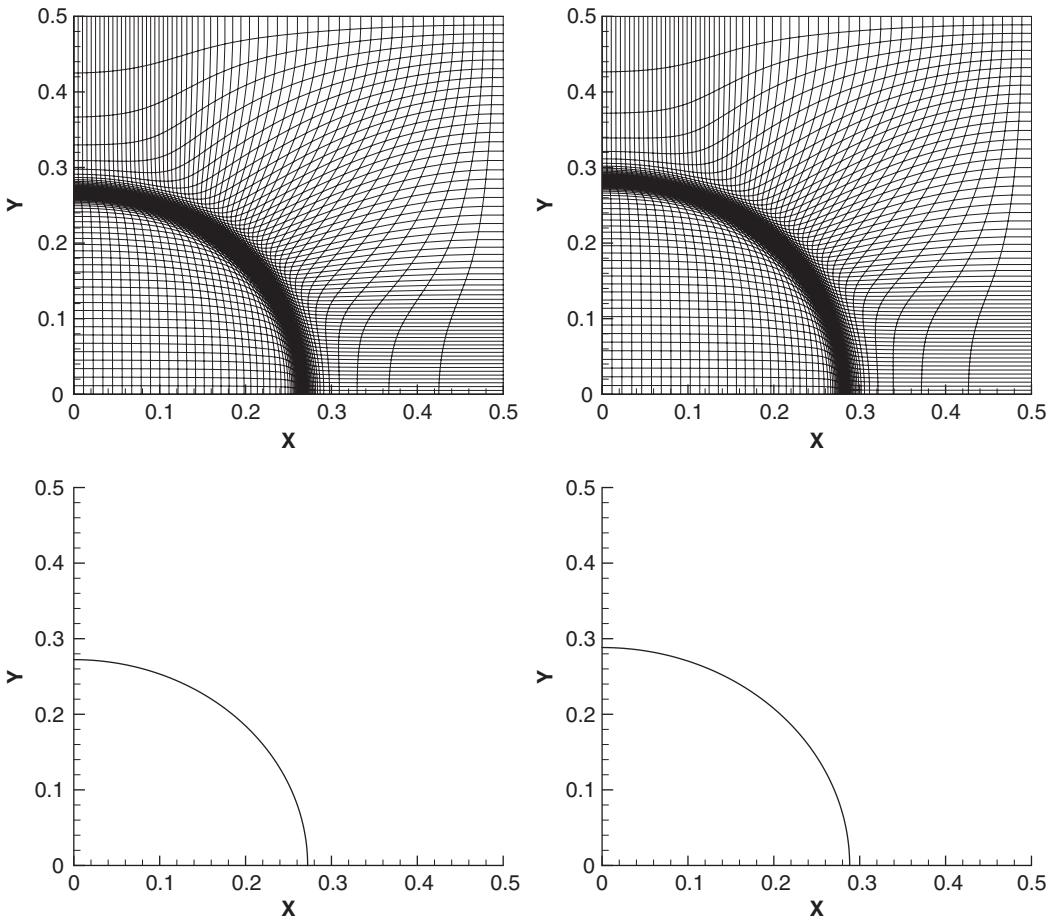

Figure 4. Critical radius equilibrium: grid (top) and interface prediction (bottom) with $\varepsilon = 1/(160\sqrt{2})$, $d_0 = 0.5$ and $R_0 = 0.26$. Left: $t = 0.04$ and right: $t = 0.08$.

solving Laplace's equation

$$\frac{1}{r}(r\theta_r(r,t))_r = 0$$

which has solution

$$\theta(r,t) = \begin{cases} B(t) & 0 < r < R_{\mathrm{I}}(t) \\ A^+(t)\ln r + b^+(t) & R_{\mathrm{I}}(t) < r < R_{\mathrm{O}}(t) \\ B^-(t) & r > R_{\mathrm{O}}(t) \end{cases} \tag{29}$$

DOI: 10.1002/fld

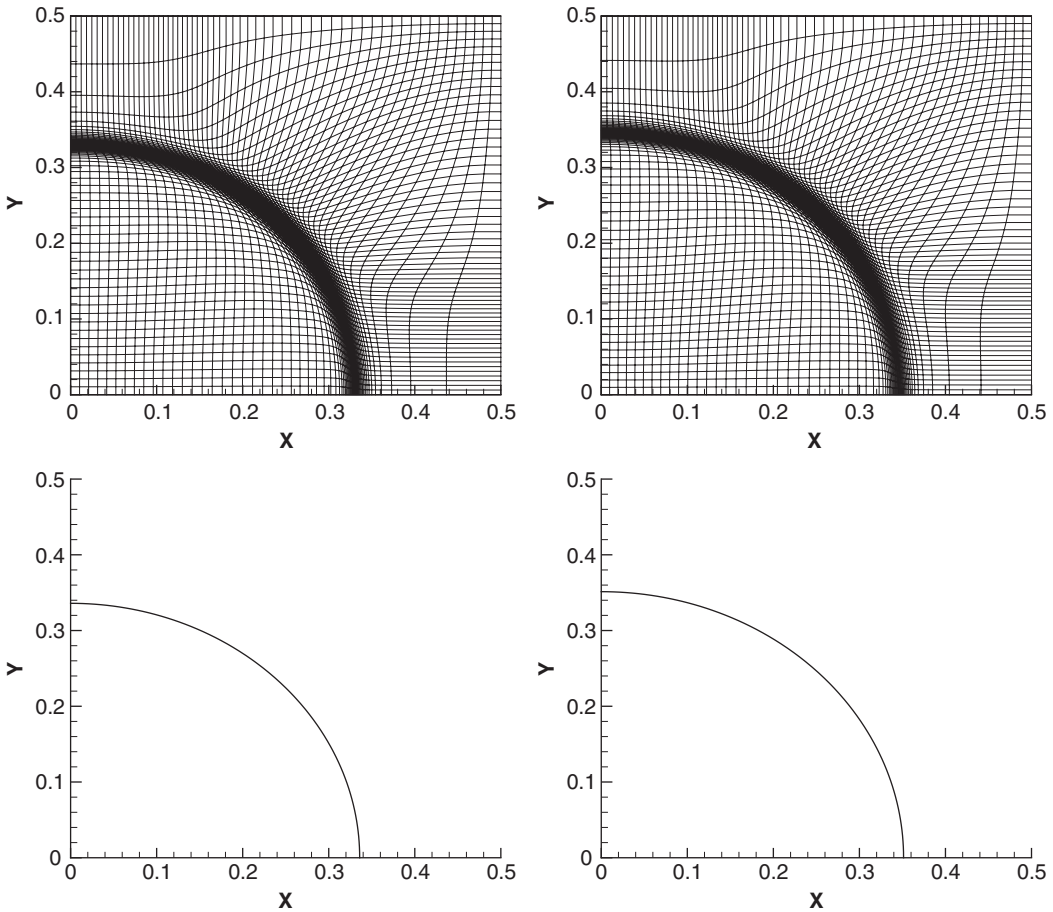

Figure 5. Same as Figure 4, except with $t = 0.16$ (left) and $t = 0.18$ (right).

At $r = R_{\mathrm{I}}(t)$ and $r = R_{\mathrm{O}}(t)$, we obtain from the jump condition (1b) that

$$R_{\mathrm{I}} \frac{\mathrm{d}R_{\mathrm{I}}(t)}{\mathrm{d}t} = R_{\mathrm{O}} \frac{\mathrm{d}R_{\mathrm{O}}(t)}{\mathrm{d}t} = -A^{+}(t) \tag{30}$$

and to ensure conservation of mass we require

$$R_{\mathrm{O}}^2(t) = R_{\mathrm{I}}^2(t) + \delta$$

where δ is a constant determined by the initial values of the radius.

From condition (1c) we have

$$B(t) = \theta(R_{\mathrm{I}}(t), t) = -\frac{d_0}{R_{\mathrm{I}}(t)} - d_0 \alpha \frac{\mathrm{d}R_{\mathrm{I}}(t)}{\mathrm{d}t}$$

DOI: 10.1002/fld

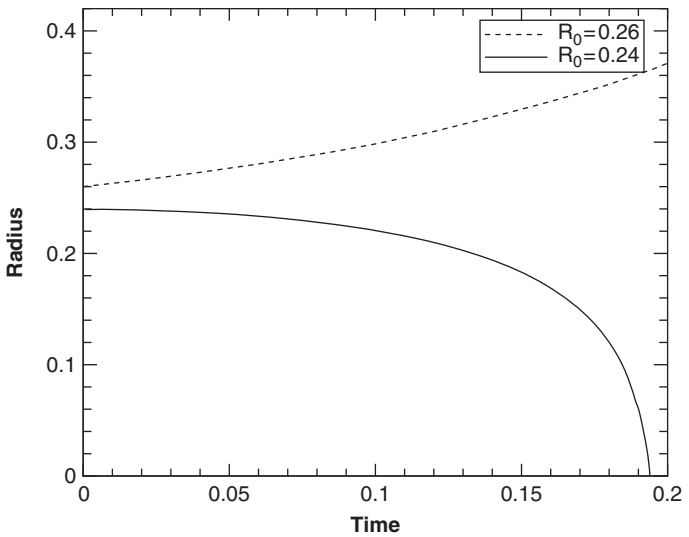

Figure 6. Critical radius equilibrium: the interface radius *versus* time. The solid line denotes the solution obtained with $R_0 = 0.24$ and the dashed line denotes the solution with $R_0 = 0.26$.

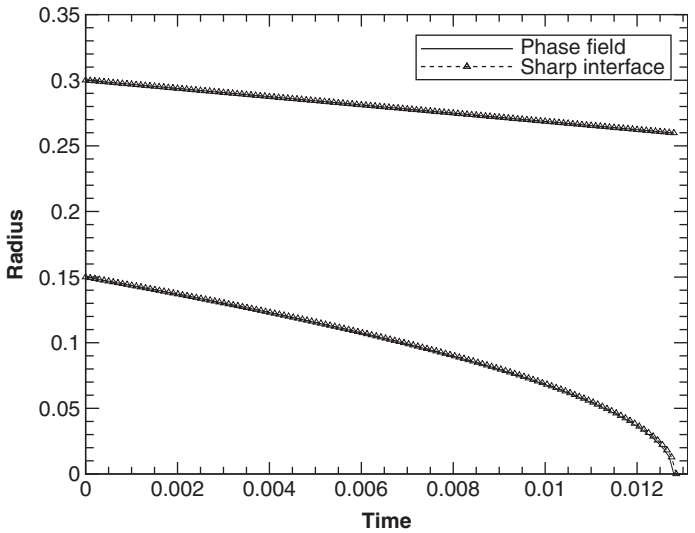

Figure 7. Scaled viscous Cahn–Hilliard problem: the interface radius *versus* time. The solid lines denote the phase-field solution, the symbolized dashed lines denote the sharp interface solution.

and

$$B^{-}(t) = \theta(R_{\mathrm{O}}(t), t) = \frac{d_0}{R_{\mathrm{O}}(t)} + d_0 \alpha \frac{\mathrm{d}R_{\mathrm{O}}(t)}{\mathrm{d}t}$$

DOI: 10.1002/fld

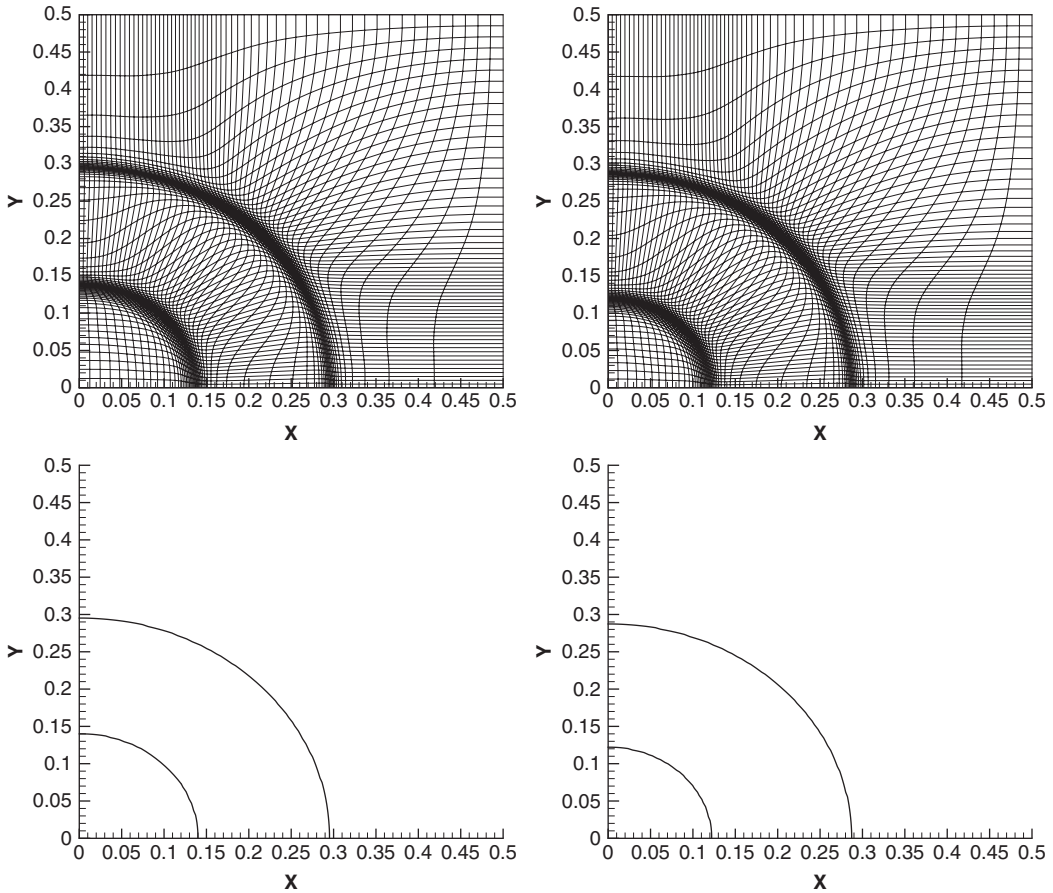

Figure 8. Scaled viscous Chan–Hilliard problem: grid (top) and interface prediction (bottom) with $\varepsilon = 1/(320\sqrt{2})$, $d_0 = 0.5$, $R_{\mathrm{I}} = 0.15$ and $R_{\mathrm{O}} = 0.3$. Left: $t = 0.0013$ and right: $t = 0.0039$.

Using the above information we find that

$$\frac{\mathrm{d}R_{\mathrm{I}}(t)}{\mathrm{d}t} = \frac{-d_0\left(\dfrac{1}{R_{\mathrm{O}}(t)} + \dfrac{1}{R_{\mathrm{I}}(t)}\right)}{d_0\alpha\left(1 + \dfrac{R_{\mathrm{I}}(t)}{R_{\mathrm{O}}(t)}\right) + R_{\mathrm{I}}(t)\ln\left(\dfrac{R_{\mathrm{O}}(t)}{R_{\mathrm{I}}(t)}\right)} \tag{31}$$

By introducing (31), we can compare the analytical interface solution to the sharp interface problem with the phase-field solution by using moving mesh method later. Equation (31) is an ODE system, and we solve (31) using a Runge–Kutta method by Shu and Osher [39] with $\alpha = 1$ and $d_0 = 0.5$. More precisely, for the ODE system $u'(t) = L(u)$, where $u = R_{\mathrm{I}}$, we use

$$u^{(1)} = u^n + \Delta t L(u^n)$$

DOI: 10.1002/fld

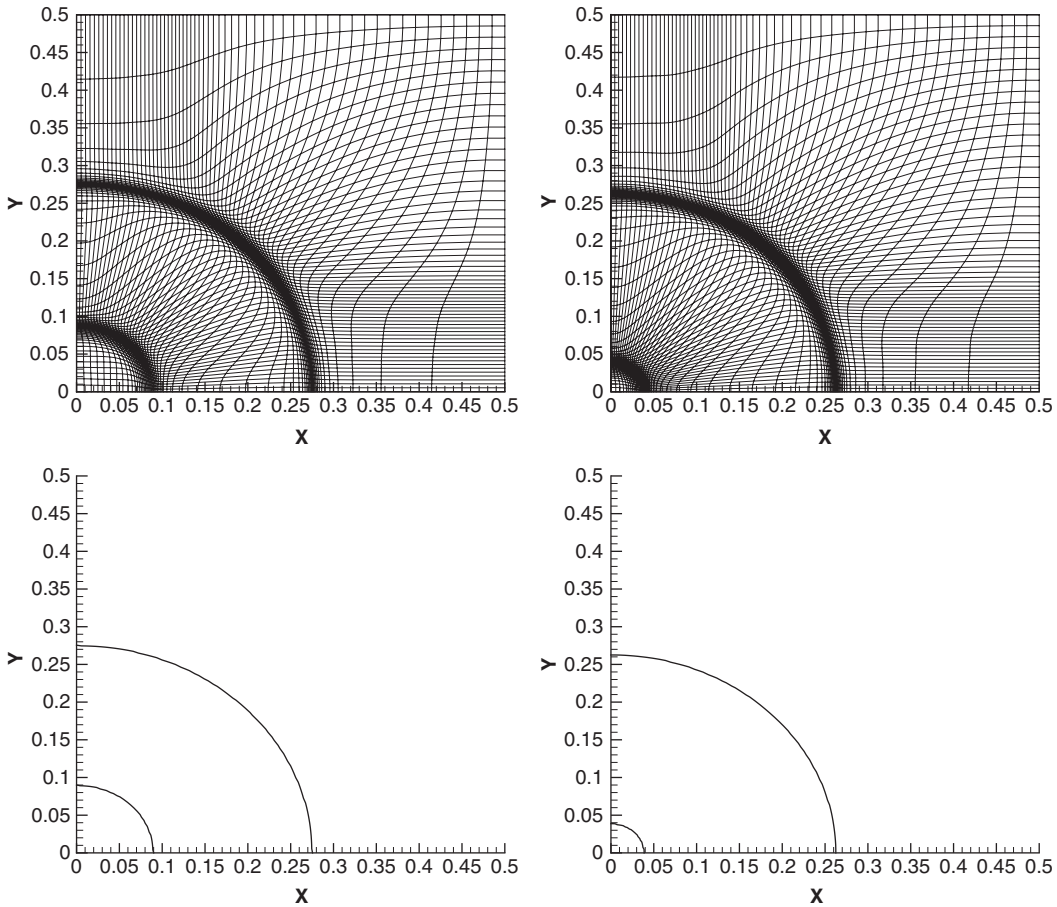

Figure 9. Same as Figure 8, except with $t = 0.0078$ (left) and $t = 0.0117$ (right).

$$u^{(2)} = \tfrac{3}{4}u^n + \tfrac{1}{4}[u^{(1)} + \Delta t L(u^{(1)})]$$

$$u^{n+1} = \tfrac{1}{3}u^n + \tfrac{2}{3}[u^{(2)} + \Delta t L(u^{(2)})]$$

The results for $R_{\mathrm{I}}(t)$ and $R_{\mathrm{O}}(t)$ as symbolized dashed lines are shown in Figure 7. It can be seen that both balls decrease in size with the inner ball shrinking faster than the outer ball.

In our computations, we set the physical parameters appeared in Section 2 to be $D = 1$, $\rho = 1$, $c = 0$, $l = 1$, $\bar{K} = 1$, $\tau = 0.08142$, $a = 0.0007366$ and $\sigma = 2.0$. We solve the phase-field equations with $\varepsilon = 1/(320\sqrt{2})$ by using a 60^2 moving grid in the domain $[0, 0.5]^2$. The monitor function used in (22) is again of the form (28). Meshes and the corresponding front prediction at the times $t = 0.0013$, 0.0039, 0.0078 and $t = 0.0117$ are plotted in Figures 8 and 9. It is observed that the meshes are clustered in the neighborhood of the two interfaces. As expected, both the interfaces move inwards with decreasing radius of the inner ball. The radial positions *versus* time for both interfaces are presented in Figure 7. We can see that the computed interface positions agree very

DOI: 10.1002/fld

well with the reference solution. Again, these are in good agreement with the moving mesh results of Beckett *et al*. [29] and the uniform mesh results of Elliott and Gardiner [7]. The monitor function used in (22) is again of the form (28).

We now briefly discuss the nonlinear iterations used in solving the system (14). 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 60^2 grid), it takes 3–5 iterations with moving mesh, and 4–5 iterations with a 300^2 grid on uniform mesh. Finally, we want to briefly discuss about the 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 computations in the paper, the time step for the critical radius equilibrium problem is 10^{-3} and for the scaled viscous Chan–Hilliard 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 [16].

6. CONCLUDING REMARKS

In this paper, we have presented an effective alternating Crank–Nicolson method of Mu and Huang [20] for solving two-dimensional phase-field equations on adaptive moving meshes. Its main advantage is that the two discrete unknown variables θ and p are decoupled, and governing by one semi-linear and one linear algebraic systems, respectively. Two algebraic systems are solved by effective iteration such as the Newton's method and the conjugate gradient method. 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*. [27–29], not only in terms of accuracy but also efficiency. Our moving mesh approach, same as that in the literatures e.g. [22], is formed by two independent parts: the grid redistribution and PDE solver. 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 [17]. 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. Numerical experiments demonstrate the efficiency of the proposed algorithm.

ACKNOWLEDGEMENTS

The authors would like to thank Professor Yunqing Huang for useful discussions and the referees for valuable comments and suggestions.

REFERENCES

1. Udaykumar HS, Mao L, Mittal R. A finite-volume sharp interface scheme for dendritic growth simulations: comparison with microscopic solvability theory. *Numerical Heat Transfer B* 2002; **42**:389–409.
2. Udaykumar HS, Marella S, Krishnan S. Sharp-interface simulation of dendritic growth with convection: benchmarks. *International Journal of Heat and Mass Transfer* 2003; **46**:2615–2627.

DOI: 10.1002/fld

3. Yang Y, Udaykumar HS. Sharp interface Cartesian grid method III: solidification of pure materials and binary solutions. *Journal of Computational Physics* 2005; **210**:55–74.
4. Bai F, Elliott CM, Gardiner A, Spence A, Stuart AM. The viscous Cahn–Hilliard equation. Part I: computation. *Nonlinearity* 1995; **8**:131–160.
5. Braun RJ, Murray BT. Adaptive phase-field computations of dendritic growth. *Journal of Crystal Growth* 1997; **174**:41–53.
6. Caginalp G, Socolovsky EA. Phase field computations of single-needle crystals, crystal growth, and motion by mean curvature. *SIAM Journal on Scientific Computing* 1994; **15**:106–126.
7. Elliott CM, Gardiner AR. Phase field equations. *Computational Techniques and Applications*, *CTAC93*. World Scientific: Singapore, 1993.
8. Karma A, Rappel W-J. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Physical Review E* 1998; **57**:4323–4349.
9. Wang S-L, Sekerka RF, Wheeler AA, Murray T, Coriell SR, Braun RJ, McFadden GB. Thermodynamically-consistent phase field-models for solidification. *Physica D* 1993; **69**:189–200.
10. Caginalp G, Socolovsky EA. Computation of sharp phase boundaries by spreading: the planar and spherically symmetric cases. *Journal of Computational Physics* 1991; **95**:85–100.
11. Nochetto RH, Paolini M, Verdi C. An adaptive finite element method for two-phase Stefan problems in two space dimensions, Part II: implementation and numerical experiments. *SIAM Journal on Scientific Computing* 1991; **12**:1207–1244.
12. Provatas N, Goldenfeld N, Dantzig J. Efficient computation of dendritic microstructures using adaptive mesh refinement. *Physical Review Letters* 1998; **80**:3308–3311.
13. Provatas N, Goldenfeld N, Dantzig J. Adaptive mesh refinement computation of solidification microstructures using dynamic data structures. *Journal of Computational Physics* 1999; **148**:265–290.
14. Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics* 1982; **46**:342–368.
15. Tan Z-J, Tang T, Zhang Z-R. A simple moving mesh method for one- and two-dimensional phase-field equations. *Journal of Computational and Applied Mathematics* 2006; **190**:252–269.
16. Tan Z-J, Zhang Z-R, Huang Y-Q, Tang T. Moving mesh methods with locally varying time steps. *Journal of Computational Physics* 2004; **200**:347–367.
17. Tang HZ, Tang T. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 2003; **41**:487–515.
18. Azarenok BN, Ivanenko SA, Tang T. Adaptive mesh redistribution method based on Godunov's scheme. *Communication on Mathematical Science* 2003; **1**:152–179.
19. Azarenok BN, Tang T. Second-order Godunov-type scheme for reactive flow calculations on moving meshes. *Journal of Computational Physics* 2005; **206**:48–80.
20. Cao WM, Huang WZ, Russell RD. An r-adaptive finite element method based upon moving mesh PDEs. *Journal of Computational Physics* 1999; **149**:221–244.
21. Di Y, Li R, Tang T, Zhang P. Moving mesh finite element methods for the incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing* 2005; **26**:1036–1056.
22. Li R, Tang T, Zhang P. Moving mesh methods in multiple dimensions based on harmonic maps. *Journal of Computational Physics* 2001; **170**:562–588.
23. Huang WZ, Russell RD. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM Journal on Scientific Computing* 1999; **20**:998–1015.
24. Li R, Tang T, Zhang P. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *Journal of Computational Physics* 2002; **177**:365–393.
25. Tan Z-J. Adaptive moving mesh methods for two-dimensional resistive magneto-hydrodynamic PDE models. *Computers and Fluids* 2007; **36**:758–771.
26. Tan Z-J, Lim KM, Khoo BC. An adaptive mesh redistribution method for the incompressible mixture flows using phase-field model. *Journal of Computational Physics* 2007, in press.
27. Mackenzie JA, Robertson ML. The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method. *Journal of Computational Physics* 2000; **161**:537–557.
28. Mackenzie JA, Robertson ML. A moving mesh method for the solution of the one-dimensional phase-field equations. *Journal of Computational Physics* 2002; **181**:526–544.
29. Beckett G, Mackenzie JA, Robertson ML. An r-adaptive finite element method for the solution of the two-dimensional phase-field equations. *Communications in Computational Physics* 2006; **1**:805–826.

DOI: 10.1002/fld

30. Mu M, Huang Y. An alternating Crank–Nicolson method for decoupling the Ginzburg–Landau equations. *SIAM Journal on Numerical Analysis* 1998; **35**:1740–1761.
31. Caginalp G. Mathematical models of phase boundaries. In *Material Instability in Continuum Problems and Related Mathematical Problems*, Ball J (ed.). Oxford Science Publications: Oxford, 1988; 35–52.
32. Caginalp G. Stefan and Hele–Shaw type models as asymptotic limits of the phase-field equations. *Physical Review A* 1989; **39**:5887–5896.
33. Mu M. A linearized Crank–Nicolson–Galerkin method for the Ginzburg–Landau model. *SIAM Journal on Scientific Computing* 1997; **18**:1028–1039.
34. Brackbill JU. An adaptive grid with direction control. *Journal of Computational Physics* 1993; **108**:38–50.
35. Winslow A. Numerical solution of the quasi-linear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics* 1967; **1**:149–172.
36. Ceniceros HD, Hou TY. An efficient dynamically adaptive mesh for potentially singular solutions. *Journal of Computational Physics* 2001; **172**:609–639.
37. Zhang Z-R. Moving mesh method with conservative interpolation based on L^2-projection. *Communications in Computational Physics* 2006; **1**:930–944.
38. Chalmers B. *Principles of Solidification*. Krieger: New York, 1977.
39. Shu C-W, Osher S. Efficient implement of essentially non-oscillatory shock-wave schemes, II. *Journal of Computational Physics* 1989; **83**:32–78.

DOI: 10.1002/fld