Journal of Computational Physics **168,** 500–518 (2001)
doi:10.1006/jcph.2001.6721, available online at http://www.idealibrary.com on IDEAL®

A Moving Mesh Finite Element Method for the Solution of Two-Dimensional Stefan Problems

G. Beckett, J. A. Mackenzie, and M. L. Robertson

Department of Mathematics, University of Strathclyde, Livingstone Tower,
26 Richmond Street, Glasgow, Scotland G1 1XH

Received February 1, 2000; revised October 20, 2000

An r-adaptive moving mesh method is developed for the numerical solution of an enthalpy formulation of two-dimensional heat conduction problems with a phase change. The grid is obtained from a global mapping of the physical to the computational domain which is designed to cluster mesh points around the interface between the two phases of the material. The enthalpy equation is discretised using a semi-implicit Galerkin finite element method using linear basis functions. The moving finite element method is applied to problems where the phase front is cusp shaped and where the interface changes topology. © 2001 Academic Press

Key Words: enthalpy; phase change; equidistribution; Stefan problem; moving meshes; adaptive method; moving finite elements.

1. INTRODUCTION

In this paper we consider the development of a simple moving mesh method to solve two-dimensional phase change problems. These arise in a number of important physical and industrial contexts such as process engineering and geophysics. A convenient formulation for the numerical solution of these problems is obtained by writing the governing heat conduction equations in terms of the enthalpy, which is the sum of the sensible and latent heats. The main advantage of this approach is that no explicit tracking of the phase front is needed and that this information can be deduced from the numerical solution a posteriori. A number of fixed grid methods have been proposed including [10, 19, 23, 27]. However, it is well known that unphysical behaviour, such as spurious temperature plateaus and oscillatory phase front movement, often occur using these methods [9, 25]. Alternatively, one can attempt to track the moving phase front by deforming the underlying mesh. Normally, this is done in such a way that element boundaries or mesh coordinate lines coincide with the phase front [17, 24]. Improved accuracy is afforded by these methods at an increased cost.

0021-9991/01 $35.00

The main disadvantage of this approach however is the complication involved when the phase front changes topology.

A number of researchers have found it possible to improve the solution process using adaptive mesh techniques. Within a finite element context this is usually achieved using the h-method of adaptation, where the mesh is locally refined or coarsened by adding or deleting points [21, 22]. A less popular approach is to use the so-called r-refinement method where mesh points are moved throughout the domain while the connectivity of the mesh is kept fixed. The main reason for the lack of popularity of this approach is the difficulty involved in controlling the geometry of the mesh elements. If this is not done with care then mesh tangling and elements with negative areas can easily arise. However, the development of a robust r-adaptive method is attractive in that it intuitively should be able to accurately resolve and follow important solution features. The coding involved in an r-adaptive method is also simpler than that involved in an h-method, which requires a considerably more complicated data structure.

The development of r-adaptive methods has come from a number of different directions. The moving finite element method of Miller and coworkers [18a] uses equations describing the mesh movement derived from the minimisation of the residual of the finite element approximation over an enlarged test space. Early versions of the method required care to prevent mesh tangling although recent formulations have led to a more robust method [6]. Since the mesh equations are coupled to the equations for the physical PDE, the resulting nonlinear systems are often very large. The fact that the mesh is often of secondary importance suggests that a more efficient decoupled procedure would be of interest. Recently, Huang and Russell [15] and Cao *et al.* [4] have developed moving mesh methods to solve time-dependent problems with steep solution fronts. The meshes are obtained from a mapping of a computational domain to the physical domain that minimises a functional related to grid smoothness, orthogonality, and adaptivity. Similar minimisation techniques have been used to generate adaptive grids for steady state problems [1, 2, 26]. For time-dependent problems the computational mesh is obtained from a gradient flow equation which is driven by a regularised form of the Euler–Lagrange equation describing the functional minimum. The regularisation in time allows the mesh to smoothly track significant solution features.

The moving mesh method of Huang and Russell [15] has been used with a finite volume approximation of the enthalpy equation by Lang [16]. A classical jump discontinuity is assumed in the temperature–enthalpy relationship and the mesh is moved towards large gradients in the enthalpy. Numerical experiments using this approach show that accurate predictions of the phase front can only be achieved using high-order upwind techniques to solve the physical PDE on the moving mesh. Upwinding is required since significant convective terms are introduced to the governing equations when the mesh points are allowed to move. At the phase change interface these convective terms dominate as no dissipation is present in the classical formulation.

The main aim of this paper is to apply the moving mesh approach of Huang and Russell [14] to the solution of a regularised enthalpy formulation of Stefan problems. The regularisation removes the need for any sophisticated upwinding and instead we apply a standard Galerkin finite element discretisation. The regularisation also allows Newton's method to be used to efficiently solve the nonlinear algebraic systems that arise from the finite element discretisation at each time step. Rather than adapt the grid towards large gradients in the enthalpy, we instead consider a very simple adaptivity criterion based on the distance from the numerical approximation of the phase front. This allows the mesh to be highly clustered where the material is changing phase and allows for the accurate determination of its position. A similar approach was used successfully as part of the one-dimensional moving mesh method of Mackenzie and Robertson [18].

The layout of this paper is as follows: in the next section we present a regularised enthalpy formulation of the heat conduction equations. In Section 3 we discuss how the moving mesh is generated along with specific adaptivity criteria for phase change problems. In Section 4 we describe a semi-implicit moving finite element discretisation of the enthalpy equation. Finally, we apply the moving mesh method to four test cases in Section 5.

2. GOVERNING EQUATIONS

Let $\Omega \in \mathbb{R}^2$ be a bounded polygonal domain and $T > 0$. Set $Q := \Omega \times (0, T)$. It is well known that a substance of constant conductivity and unit density satisfies the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + f(x, y, t), \tag{2.1}$$

where θ is the temperature, $u(\theta)$ is the enthalpy, and $f(x, y, t)$ represents any body heating or cooling sources. If a pure substance with constant specific heats c_1 and c_2 undergoes a change of phase at the temperature $\theta = \theta_m$ then the enthalpy may be written as

$$u(\theta) = \begin{cases} c_1(\theta - \theta_{ref}), & \theta < \theta_m \\ u(\theta_m^-) + \lambda + c_2(\theta - \theta_m), & \theta_m \le \theta, \end{cases} \tag{2.2}$$

where $u(\theta_m^-) = \lim_{\delta \to 0^-} u(\theta_m + \delta)$, θ_{ref} is any reference temperature below θ_m, and λ is the latent heat.

For the reasons outlined in the Introduction, various attempts have been made to regularise the discontinuity in $u(\theta)$. We consider a continuously differentiable relationship suggested by Egolf and Manz [12] which takes the form

$$u(\theta) = \begin{cases} u_{ref} + c_1(\theta - \theta_m) + \frac{\lambda}{2} \exp\left(-\frac{|\theta - \theta_m|}{\varepsilon^-}\right), & \theta < \theta_m \\ u(\theta_m) + \frac{\lambda}{2} + c_2(\theta - \theta_m) - \frac{\lambda}{2} \exp\left(-\frac{|\theta - \theta_m|}{\varepsilon^+}\right), & \theta \ge \theta_m, \end{cases} \tag{2.3}$$

where ε^- and ε^+ determine the rates at which the temperature–enthalpy function asymptotes to the linear relationship away from the phase change temperature θ_m (see Fig. 1). For $u(\theta)$ to be continuously differentiable at θ_m we require that

$$c_1 - c_2 = \frac{\lambda}{2}\left(\frac{1}{\varepsilon^+} - \frac{1}{\varepsilon^-}\right). \tag{2.4}$$

If $\varepsilon = \varepsilon^- + \varepsilon^+$ then we can define a modified Stefan number

$$St^* = \frac{(c_2 - c_1)\varepsilon}{\lambda}. \tag{2.5}$$

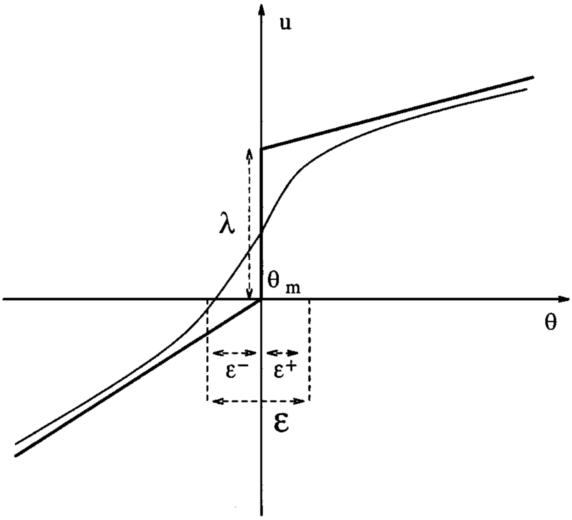

FIG. 1. Regularised temperature–enthalpy function.

The simultaneous satisfaction of (2.4) and (2.5) gives rise to quadratic equations for ε^- and ε^+ which have physically relevant solutions given by

$$\varepsilon^- = \frac{\varepsilon}{2St^*}(1 + St^* - \sqrt{1 + St^*}) \tag{2.6}$$

and

$$\varepsilon^+ = \frac{\varepsilon}{2St^*}(St^* - 1 + \sqrt{1 + St^*}). \tag{2.7}$$

In the limit $c_2 \rightarrow c_1$ we have $\varepsilon^+ \rightarrow \varepsilon^- = \varepsilon/2$.

The original motivation for this model was to describe mixtures and glassy substances that have a continuous enthalpy transition from a pure solid to a pure liquid phase. However, here we primarily use this model to regularise the temperature–enthalpy relationship. In [20], Nochetto considered the effect of regularisation on the L_2 error in the temperature using a continuous piecewise linear function. If there are no mushy regions then the error in the temperature is $O(\varepsilon)$. Numerical experiments in one dimension [18] suggest that we also introduce an $O(\varepsilon)$ error due to the exponential-based regularisation. The main idea using an adaptive moving mesh is to use smaller values of ε than would be possible using a fixed mesh so that the regularisation error is much smaller than the discretisation error.

3. A MOVING MESH STRATEGY

To generate an adaptive mesh it is useful to regard the physical domain Ω_p as the image of a computational (logical) domain Ω_c under the invertible maps

$$x = x(\xi, \eta), \quad y = y(\xi, \eta) \quad \text{and} \quad \xi = \xi(x, y), \quad \eta = \eta(x, y), \tag{3.1}$$

where $\boldsymbol{x} = (x, y)$ and $\boldsymbol{\xi} = (\xi, \eta)$ are the physical and computational coordinates, respectively. A mesh covering Ω_p is obtained by applying the mapping given in (3.1) to a partitioning of Ω_c.

A popular way to choose the coordinate transformation for steady problems is to require that it minimises a functional of the form

$$F(\xi,\eta)=\frac{1}{2}\int_{\Omega_p}(\nabla\xi^T G^{-1}\nabla\xi+\nabla\eta^T G^{-1}\nabla\eta)\,\mathrm{d}x\,\mathrm{d}y, \tag{3.2}$$

where $\nabla=(\partial/\partial x,\partial/\partial y)$ and $G(x,y)$ is a 2×2 symmetric positive definite (SPD) matrix, often referred to as a monitor matrix. The idea in adaptive mesh generation is to choose G to concentrate mesh points in Ω_p where the PDE is difficult to solve. However, constructing a suitable monitor matrix is not an easy task. Given a 2×2 SPD matrix it can be written in terms of its eigendecomposition

$$G=\lambda_1\boldsymbol{v}_1\boldsymbol{v}_1^T+\lambda_2\boldsymbol{v}_2\boldsymbol{v}_2^T, \tag{3.3}$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are normalised eigenvectors with corresponding eigenvalues λ_1 and λ_2. A study by Cao *et al.* [5] shows that if the mesh Ω_c is uniformly distributed then the adaptive mesh generated by minimising $F(\xi,\eta)$ is concentrated in regions where λ_1 and λ_2 change rapidly. The analysis therefore allows a direct way of defining a monitor matrix by specifying a suitable normalised direction $\boldsymbol{v}_1$ and setting $\boldsymbol{v}_2=\boldsymbol{v}_1^\perp$. Thereafter λ_1 is chosen to have a suitable variation in the direction given by $\boldsymbol{v}_1$. This leaves the choice of λ_2 and generally the smaller the ratio λ_1/λ_2 the more the grid adapts in the direction $\boldsymbol{v}_1$ and hence two-dimensional effects become less pronounced.

A number of choices of monitor matrices are discussed in [5]. For example, one possibility is to set

$$\boldsymbol{v}_1=\frac{\nabla u}{|\nabla u|},\quad \boldsymbol{v}_2=\boldsymbol{v}_1^\perp,\quad \lambda_1=\sqrt{1+|\nabla u^2|},$$

which adapts the mesh to large gradient changes in the function $u(\boldsymbol{x})$. The eigenvalue λ_2 can be chosen as a function of λ_1. For example, if $\lambda_1=\lambda_2$ then $G=\lambda_1 I$ and this results in minimising Winslow's functional [26]. If $\lambda_2=1/\lambda_1$ then $G=M/\sqrt{\det(M)}$, where $M=I+(\nabla u)(\nabla u)^T$ and we arrive at a method based on harmonic mappings [11]. If $\lambda_2=1$ then $G=(I+(\nabla u)(\nabla u)^T)^{1/2}$, which is a generalisation of the well-known arc-length monitor function used in equidistribution schemes in one dimension.

For Stefan problems the main numerical difficulty occurs at the phase change interface which can be detected by a large local gradient in the enthalpy. The moving mesh scheme of Lang [16] uses the arc-length monitor matrix based on gradients of the enthalpy. However, it is not uncommon to have regions of the domain that are far from the phase front where the gradient of the enthalpy is significant enough to affect the clustering of the grid. For example, problems with a large Stefan number are characterised by the fact that the latent heat jump is small in relation to the temperature, and hence enthalpy, difference across the domain. Numerical experiments in Section 5.1 show that this can lead to unnecessary mesh clustering away from the phase change interface.

In an attempt to focus the mesh adaption towards the phase front we instead consider a monitor matrix of Winslow-type with

$$G=\left(1+\frac{\mu_1}{\sqrt{\mu_2^2|\boldsymbol{x}-\boldsymbol{x}_*|^2+1}}\right)I. \tag{3.4}$$

Here $\boldsymbol{x}_*$ is the closest point to $\boldsymbol{x}$ on the numerical estimate of any phase front, and μ_1 and μ_2 are user-chosen parameters. The one-dimensional equivalent of this monitor matrix has been used successfully in the moving mesh method of Mackenzie and Robertson [18]. The experience gained in one dimension shows that the parameter μ_1 controls the minimum mesh spacing whereas μ_2 controls the rate at which mesh clustering occurs.

The calculation of $\boldsymbol{x}_*$ is achieved as follows. At time t^n we have a piecewise linear temperature field $\Theta(\boldsymbol{x}, t^n)$ obtained from a finite element discretisation which is described in the next section. A piecewise linear representation of a phase interface is then obtained from the ordered list of points $\{\boldsymbol{x}_i^c\}_{i=1}^{N_c}$ such that $\Theta(\boldsymbol{x}_i^c) = \theta_m$. These points are obtained from a plotting routine used to display the numerical results. This routine can also easily detect if more than one phase front is present. A smooth representation of an interface I^n is then obtained by an arc-length parameterised spline passing through the points $\{\boldsymbol{x}_i^c\}_{i=1}^{N_c}$. This curve is then partitioned by a set of points $\{\boldsymbol{x}_i^s\}_{i=1}^{N_s}$ which is uniformly distributed along I^n with $\boldsymbol{x}_1^s = \boldsymbol{x}_1^c$ and $\boldsymbol{x}_{N_s}^s = \boldsymbol{x}_{N_c}^c$. Finally, we set $\boldsymbol{x}_* = \min_{1\le i\le N_s} |\boldsymbol{x}_i^s - \boldsymbol{x}|$. In all the calculations performed in Section 5 we have set $N_s = 100$, $\mu_1 = 100$, and $\mu_2 = 40$.

Once a monitor matrix is decided upon, the computational mesh is then found by solving the Euler–Lagrange equations

$$\nabla \cdot (G^{-1}\nabla\xi) = 0 \quad \text{and} \quad \nabla \cdot (G^{-1}\nabla\eta) = 0. \tag{3.5}$$

In practice we solve for $\boldsymbol{x}(\boldsymbol{\xi})$ as this defines the physical mesh used with the finite element discretisation. Therefore, by interchanging the roles of the dependent and independent variables we find that (3.5) takes the form

$$\frac{\partial}{\partial\xi}\left(\frac{\boldsymbol{x}_\eta^T G \boldsymbol{x}_\eta}{Jg}\right) - \frac{\partial}{\partial\eta}\left(\frac{\boldsymbol{x}_\xi^T G \boldsymbol{x}_\eta}{Jg}\right) = 0 \tag{3.6}$$

and

$$-\frac{\partial}{\partial\xi}\left(\frac{\boldsymbol{x}_\eta^T G \boldsymbol{x}_\xi}{Jg}\right) + \frac{\partial}{\partial\eta}\left(\frac{\boldsymbol{x}_\xi^T G \boldsymbol{x}_\xi}{Jg}\right) = 0, \tag{3.7}$$

where $J = x_\xi y_\eta - y_\xi x_\eta$ and $g = \det(G)$.

For time-dependent problems we follow the approach of Huang and Russell [14] where the coordinate mapping solves the gradient flow equations

$$\frac{\partial\boldsymbol{\xi}}{\partial t} = \frac{1}{\tau}\nabla \cdot (G^{-1}\nabla\boldsymbol{\xi}). \tag{3.8}$$

Here $\tau > 0$ is a temporal relaxation parameter that determines the rate at which the computational mesh attempts to minimise the functional F. Writing the gradient flow equations in terms of the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ results in the coupled set of parabolic equations

$$\frac{\partial\boldsymbol{x}}{\partial t} = \frac{1}{\tau}\left[\sum_{i,j} a_{i,j}\frac{\partial^2\boldsymbol{x}}{\partial\xi^i\partial\xi^j} + \sum_i b_i\frac{\partial\boldsymbol{x}}{\partial\xi^i}\right], \tag{3.9}$$

where

$$a_{i,j} = \boldsymbol{a}^i \cdot G^{-1}\boldsymbol{a}^j, \quad b_i = -\sum_j \boldsymbol{a}^i\frac{\partial G^{-1}}{\partial\xi^j}\boldsymbol{a}^j, \tag{3.10}$$

and the contravariant base vectors $\boldsymbol{a}^i = \nabla\xi^i$, $i = 1, 2$. The temporal discretisation of (3.9) is achieved using a semi-implicit approach where

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t\left(a_{11}^n \boldsymbol{x}_{\xi\xi}^{n+1} + a_{12}^n \boldsymbol{x}_{\xi\eta}^{n+1} + a_{22}^n \boldsymbol{x}_{\eta\eta}^{n+1} + b_1^n \boldsymbol{x}_{\xi}^{n+1} + b_2^n \boldsymbol{x}_{\eta}^{n+1}\right). \tag{3.11}$$

The spatial discretisation of (3.11) is performed using second-order central finite differences on an $N \times N$ uniform partition of $\Omega_c = (0, 1) \times (0, 1)$. The effect of evaluating the coefficients $a_{i,j}$, and b_i at time t^n is to decouple the solution of the mesh from that of the physical PDE. This has the advantage that different iterative techniques and different tolerances can be used for each system. In particular, the discretisation of the moving mesh equations (3.9) results in a linear algebraic system which we solve using an ILU-preconditioned BICGstab routine with a tolerance of 10^{-8}.

Dirichlet boundary conditions for the above system are obtained by solving a one-dimensional moving mesh PDE. If $\Gamma \in \partial\Omega$ and $\Gamma_c \in \partial\Omega_c$ denote the physical and computational boundary segments with arc-lengths l and l_c respectively, then the mesh on Γ is the solution of

$$\frac{\partial s}{\partial t} = \frac{1}{\tau}\left(M\frac{\partial s}{\partial \zeta}\right)^{-2} \frac{\partial}{\partial \zeta}\left(M\frac{\partial s}{\partial \zeta}\right), \quad \zeta \in (0, l_c), \tag{3.12}$$

with $s(0) = 0$ and $s(l_c) = l$. Here M is the one-dimensional projection of the two-dimensional monitor function along the boundary.

4. A MOVING FINITE ELEMENT DISCRETISATION

We will assume that $[0, T]$ is partitioned by uniform time intervals $\Delta t = T/Nt$ such that

$$0 = t^0 < t^1 < \cdots < t^{Nt-1} < t^{Nt} = T.$$

Using the procedure described in the previous section we will assume that at t^{n+1} we have a triangular mesh $\mathcal{S}^{n+1}$ that has the same connectivity as the mesh at the previous time step $\mathcal{S}^n$. Therefore, each element $K(t^{n+1})$ of $\mathcal{S}^{n+1}$ corresponds to a unique element $K(t^n)$ of $\mathcal{S}^n$.

We consider approximations of the form

$$U(x, y, t) = \sum_j U_j(t)\phi_j(x(t), y(t)), \quad \Theta(x, y, t) = \sum_j \Theta_j(t)\phi_j(x(t), y(t)),$$

where $\phi_j(x(t), y(t))$ is the usual piecewise linear basis function associated with the node $(x_j(t), y_j(t))$, and $U_j(t) = u(\Theta_j(t))$ where $u(\theta)$ is given by (2.3). While spatial derivatives of U and Θ retain their usual form the temporal derivatives must take account of the motion of the mesh. For example,

$$\begin{aligned} U_t &= \frac{\partial}{\partial t}\sum_j U_j(t)\phi_j(x(t), y(t)) = \sum_j \{\dot{U}_j\phi_j + U_j\dot{\phi}_j\} \\ &= \sum_j \left\{\dot{U}_j\phi_j + U_j\left(\frac{D\phi_j}{Dt} - \frac{\partial\phi_j}{\partial x}\dot{x}_j - \frac{\partial\phi_j}{\partial y}\dot{y}_j\right)\right\}, \end{aligned}$$

where D/Dt is the derivative along the path given by $(x_j(t), y_j(t))$. Since ϕ_j is constant along this path we have

$$U_t = \sum_j \{\dot{U}_j \phi_j - U_j \dot{x}_j (\phi_j)_x - U_j \dot{y}_j (\phi_j)_y\}. \tag{4.1}$$

The additional terms appearing in (4.1) can be viewed as a correction for the convective effects of the mesh motion.

Let $\mathcal{J}$ denote the set of indices of the mesh points and $\mathcal{J}_0$ the subset of $\mathcal{J}$ excluding those corresponding to Dirichlet boundary conditions. The finite element solution satisfies the weak formulation

$$(U_t, \phi_i) - (\Theta_{xx} + \Theta_{yy}, \phi_i) = (f, \phi_i), \quad \forall i \in \mathcal{J}_0,$$

where $(\cdot, \cdot)$ denotes the L_2 inner product over Ω. Using integration by parts and the fact that $\phi_i = 0$ on $\partial\Omega$ we have

$$(U_t, \phi_i) + (\Theta_x, (\phi_i)_x) + (\Theta_y, (\phi_i)_y) = (f, \phi_i) \quad \forall i \in \mathcal{J}_0. \tag{4.2}$$

Substituting (4.1) into (4.2) we get

$$\sum_j \dot{U}_j (\phi_j, \phi_i) - \sum_j U_j (\dot{x}_j (\phi_j)_x \phi_i + \dot{y}_j (\phi_j)_y \phi_i) + \sum_j \Theta_j ((\phi_j)_x (\phi_i)_x + (\phi_j)_y (\phi_i)_y)$$
$$= (f, \phi_i).$$

This is a system of ODEs of the form

$$M\dot{\boldsymbol{U}} + B\boldsymbol{U} - K\boldsymbol{\Theta} = \boldsymbol{f},$$

where $\boldsymbol{U}$, $\boldsymbol{\Theta}$ are the unknown nodal values of the enthalpy and temperature, $M(t)$ is the mass matrix, $B(t)$ is a matrix associated with the movement of the mesh, $K(t)$ is the stiffness matrix, and $\boldsymbol{f}(t)$ is the load vector. The integrations involved in the inner products for the matrices M, B, and K can all be calculated exactly. The load vector $\boldsymbol{f}$ is calculated using one-point Gaussian quadrature.

Analogous to the approach followed by Mackenzie and Robertson [18] in one dimension, we use a semi-implicit discretisation

$$M^{n+1}\boldsymbol{U}^{n+1} - \Delta t K^{n+1}\boldsymbol{\Theta}^{n+1} = M^{n+1}\boldsymbol{U}^n - \Delta t (B^n \boldsymbol{U}^n - \boldsymbol{f}^{n+1}). \tag{4.3}$$

Note that the terms introduced by the mesh motion are treated explicitly whereas the mass and stiffness matrix terms are treated implicitly. The main motivation behind this approach is to try to obtain a set of nonlinear equations for $\boldsymbol{\Theta}^{n+1}$ that can be shown to possess a unique solution. For the one-dimensional finite difference scheme considered by Mackenzie and Robertson [18], an existence and uniqueness result followed from the fact that the $\boldsymbol{U}^{n+1}$ above was multiplied by a diagonal matrix and the one-dimensional equivalent of the stiffness matrix was easily shown to be an M-matrix. For the two-dimensional case we could diagonalize the mass matrix by mass lumping but the main difficulty comes in showing that the stiffness matrix K^{n+1} is an M-matrix. Restrictions on the mesh that guarantee this property include the well-known weakly acute condition that all the angles occurring in each triangular element are less than $\pi/2$ (see Ciarlet and Raviart [18]). Unfortunately, this condition is far too restrictive when the mesh elements are allowed to deform in an r-adaptive method. Therefore, at present we have no theoretical result to show that a unique solution of (4.3) exists. However, in practice we find that we have no difficulty in finding solutions using the Newton iteration

$$\left(M^{n+1}\left(\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\Theta}}\right)^{[n+1,s]} - \Delta t K^{n+1}\right)\left(\boldsymbol{\Theta}^{[n+1,s+1]} - \boldsymbol{\Theta}^{[n+1,s]}\right) = \boldsymbol{r}^{[n+1,s]}, \tag{4.4}$$

where

$$\boldsymbol{r}^{[n+1,s]} = M^{n+1}\left(\boldsymbol{U}^n - \boldsymbol{U}^{[n+1,s]}\right) + \Delta t\left(K^{n+1}\boldsymbol{\Theta}^{[n+1,s]} - B^n\boldsymbol{U}^n + \boldsymbol{f}^{n+1}\right).$$

The Newton iteration is well defined since the regularised temperature–enthalpy function is continuously differentiable. The linear algebraic system (4.4) is solved iteratively using an ILU-preconditioned BICGstab routine to a tolerance of 10^{-10}. The outer Newton iteration is performed until the tolerance is also below 10^{-10}.

5. NUMERICAL RESULTS

5.1. *Solidification in a Wedge*

The first test case we consider is the solidification of a material in an infinite wedge. For numerical purposes we use the domain $\Omega = (0, 1)^2$ and $T = 0.1$. The initial condition $\theta(x, y, 0) = 0.3$ is prescribed throughout the domain. Thereafter, the Dirichlet condition $\theta = -1$ is imposed on $x = 0$ and $y = 0$, and homogeneous Neumann conditions are imposed on the remaining two edges. The latent heat for this example is $\lambda = 0.25$. This test case has also been used in the numerical work of [10] and [13].

A semi-analytical solution of this problem was proposed by Budhia and Kreith [3]. Their solution is the linear superposition of solutions to two separate problems. The first is a heat conduction problem, without a phase change, where the medium is initially above the melting temperature and the temperature of the wedge boundaries is held at a constant below the melting temperature for $t \geq 0$. The second problem is that of a moving heat source at the interface in a medium initially at the melting temperature. The moving source at the interface replaces the latent heat due to the phase change. The interface position is given by the solution of a nonlinear integro-differential equation. To simplify the solution process the authors assume that the interface is a one-parameter hyperbola and the free parameter is chosen so that the integro-differential equation is satisfied at one point on the interface. For this test case we have taken this point along the line $y = x$.

Figure 2 shows the computed moving meshes with $N = 30$, $Nt = 80$, $\tau = 0.1$, and $\varepsilon = 0.005$. We can see clearly that the mesh is clustered around the phase change interface and it follows its movement across the domain. Figure 2 also compares the computed interface positions, which are denoted by dashed lines, with the interfaces computed using a uniform fine grid with $N = 180$, $Nt = 120$, and $\varepsilon = 0.005$. The accuracy of the interface predictions is very good. Figures 3a and 3b show the computed temperature and enthalpy along the top boundary $y = 1$ which show excellent agreement with the fine grid solution.

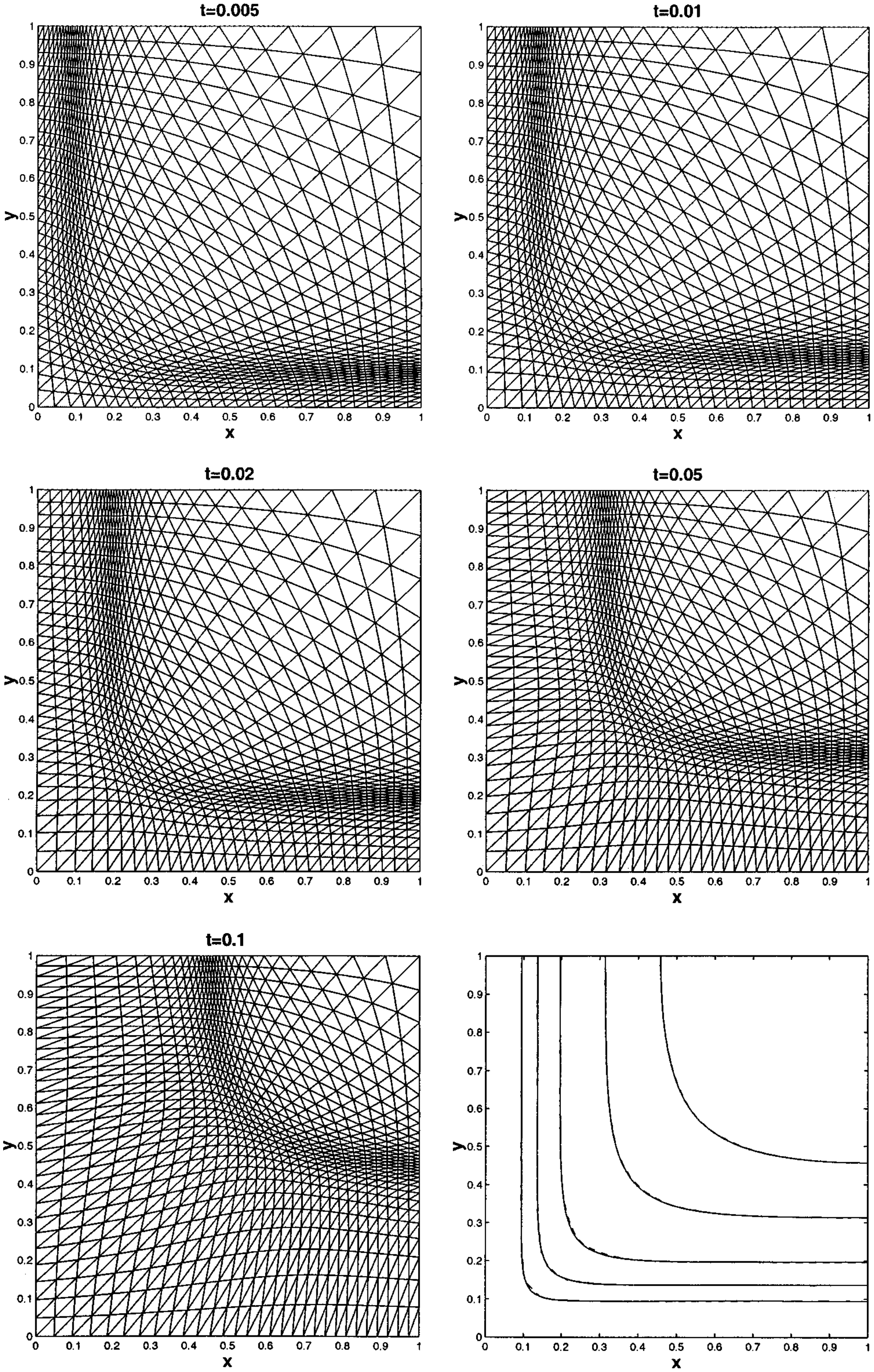

FIG. 2. Results for wedge solidification problem.

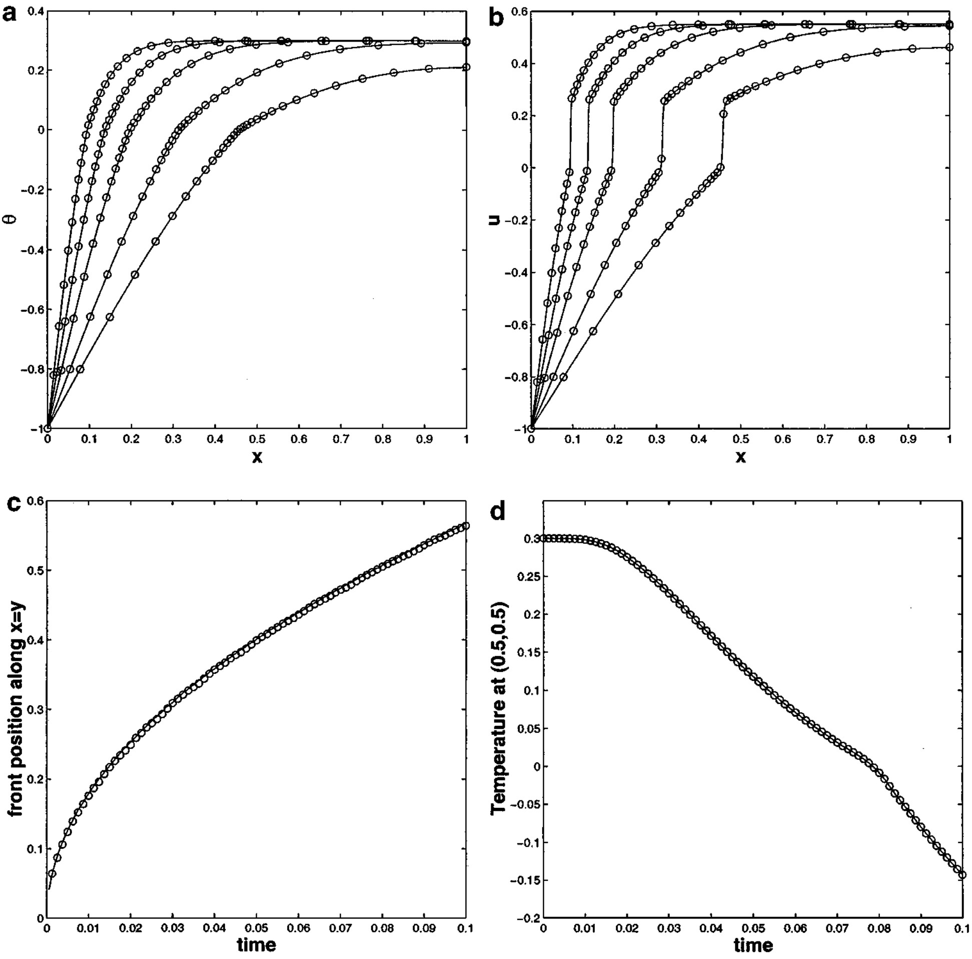

FIG. 3. Results for wedge solidification problem.

Figure 3c shows that there is very good agreement between the computed front position and the semi-analytic solution of [3] which is highly accurate along the line $y = x$. The temperature history of the point (0.5, 0.5) is shown in Fig. 3d which is again predicted accurately compared to the fine grid solution.

Figure 4 compares the grids obtained using the monitor matrices mentioned in Section 3. The three approaches based on the gradient of the enthalpy all exhibit more grid clustering within the solid phase than in the liquid phase. The reason for this is clear from Fig. 3b where we see that there is a significant gradient of the enthalpy in the solid phase. However, the variation is almost linear in the solid phase and there are no good approximation reasons for the grid being more clustered here. By contrast, the grid obtained using the monitor matrix (3.4) is symmetrically graded around the phase front.

5.2. *An Oscillating Circle*

The second test case we consider is the movement of an oscillating circular interface. This problem was originally devised by Nochetto *et al.* [21] and has also been considered as a test case for a level set method [7]. We have $\Omega = (0, 5) \times (-1, 4)$ and $T = \pi/1.25$.

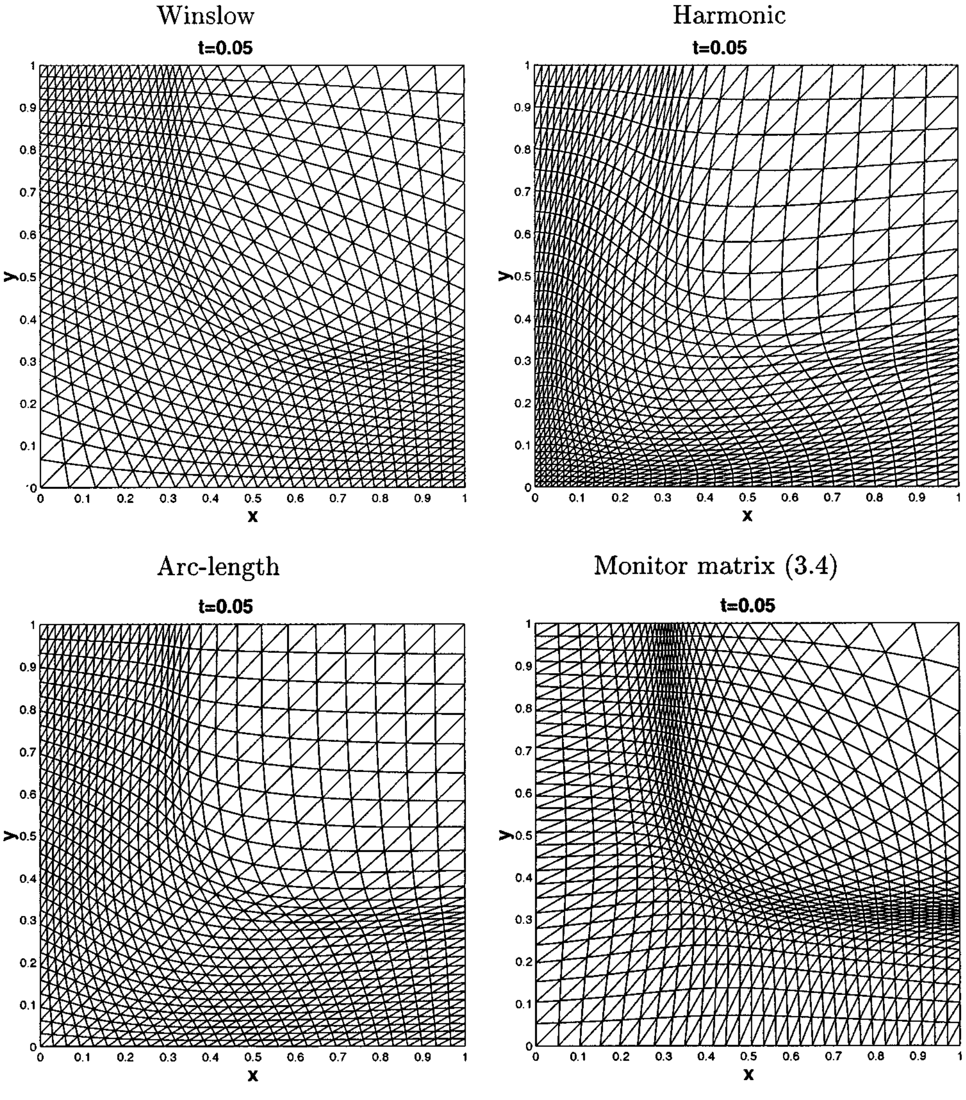

FIG. 4. Comparison of monitor functions for wedge solidification problem.

The forcing function is chosen such that the exact temperature is given by

$$\theta(x, y, t) = \begin{cases} 0.75(r^2 - 1), & r < 1, \\ (1.5 - \dot{\alpha}(t)\sin\varphi)(r - 1), & r \geq 1, \end{cases} \tag{5.1}$$

where $r = (x^2 + (y - \alpha(t))^2)^{1/2}$, $\alpha(t) = 0.5 + \sin(1.25t)$, and $\sin\varphi = (y - \alpha(t))/r$. Dirichlet boundary conditions are imposed on the sides $y = -1$, $y = 4$, and $x = 5$, whereas a homogeneous Neumann condition is prescribed on $x = 0$. The latent heat for this example is $\lambda = 1$. The exact interface, I, is a unit circle with centre $(0, \alpha(t))$ that moves up and down.

Figure 5 shows the computed moving meshes and interface predictions with $N = 30$, $Nt = 80$, $\tau = 0.1$, and $\varepsilon = 0.025$. Again the numerically computed phase interface is denoted by a dashed line. Clearly, the grid does a good job of following the movement

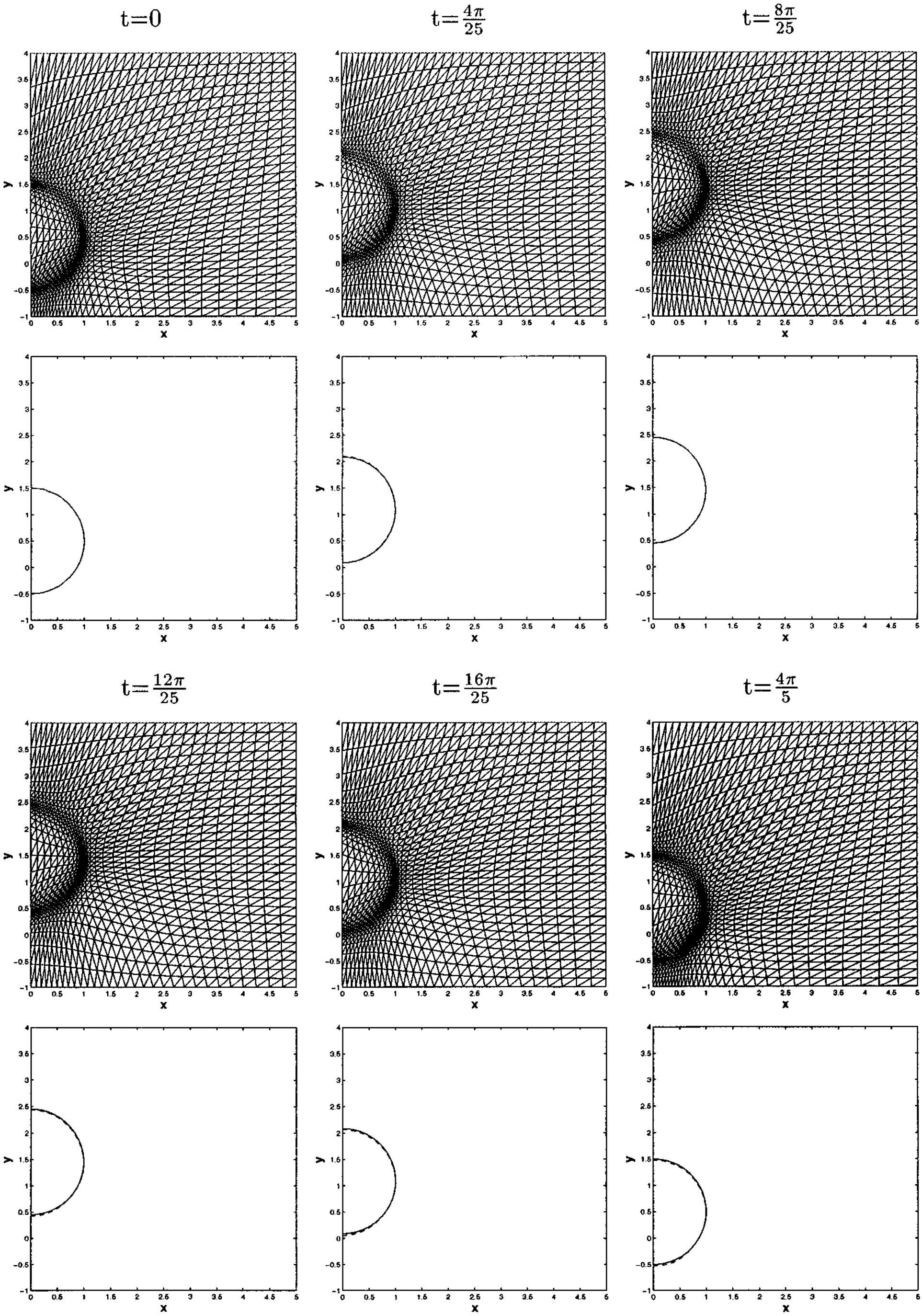

FIG. 5. Grids and interface predictions for oscillating circle problem.

of the interface. The interpolation errors using an initial uniform and adapted mesh with the same value of N are shown in Fig. 6, where the darker shaded areas correspond to larger errors. It is clear that the uniform mesh has significant errors around the position of the initial interface and that these are reduced by a factor of three by adapting the initial

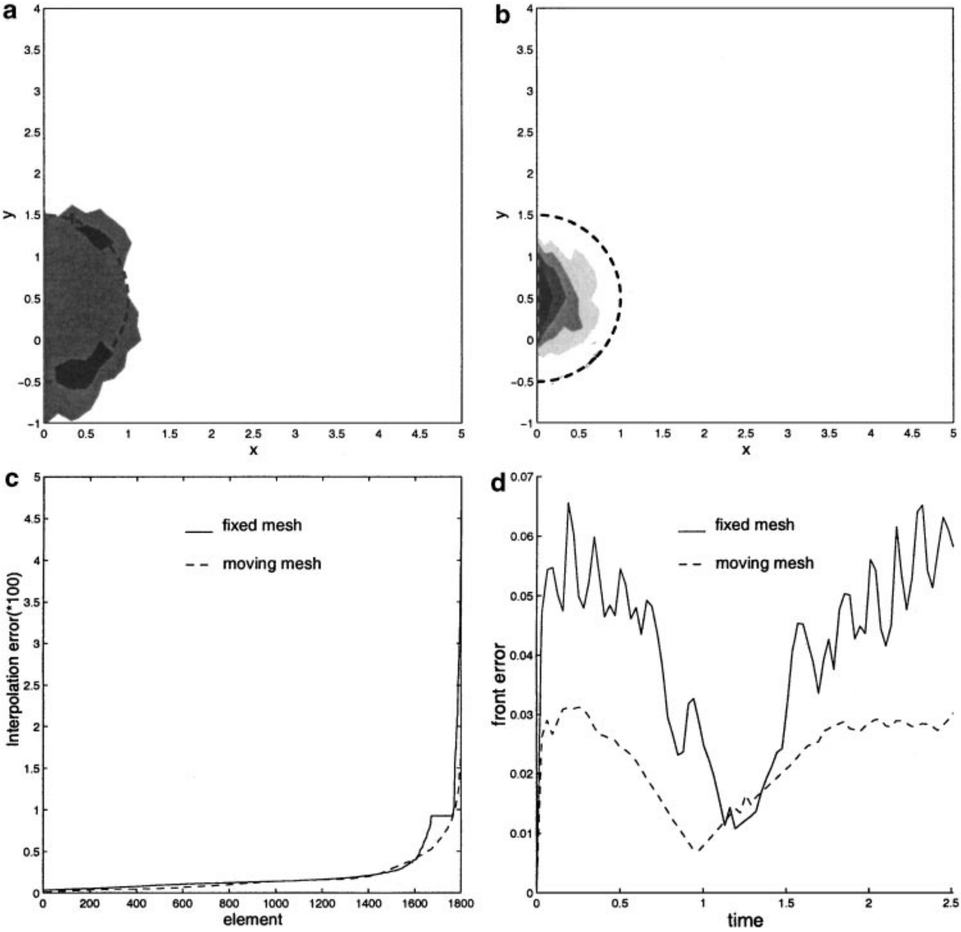

FIG. 6. Interpolation errors for oscillating circle problem with $N = 30$: (a) fixed and (b) adapted initial meshes and (c) distribution of errors. (d) shows the front error over time.

mesh. The evolution of the front error is also shown in Fig. 6 where we can see that the improvement over the fixed mesh is maintained throughout the simulation. Table I compares the performance of the moving mesh method with that of a fixed grid method. The notation

TABLE I

Errors for Moving and Fixed Grid Solutions of Oscillating Circle Problem

Moving mesh method				Fixed mesh method			
$Nt \times N$	E_θ^∞	E_I^∞	CPU	$Nt \times N$	E_θ^∞	E_I^∞	CPU
40 × 18	0.1040	0.0785	1.0	75 × 32	0.103	0.0814	2.2
60 × 25	0.0638	0.0404	2.9	100 × 43	0.0798	0.0545	5.9
80 × 30	0.0494	0.0312	5.8	150 × 65	0.0487	0.0390	26.4

TABLE II
Errors for *h*-adaptive and Fixed Grid Solutions of Oscillating Circle Problem Using the Methods of Nochetto *et al.* [21]

h-adaptive method			Fixed mesh method		
$Nt \times J$	E_θ^∞	E_I^∞	$Nt \times J$	E_θ^∞	E_I^∞
40 × 339	0.1070	0.0583	100 × 1812	0.1249	0.0814
60 × 592	0.0742	0.0501	150 × 4107	0.0778	0.0545
80 × 818	0.0517	0.0326	200 × 7361	0.0631	0.0390

used in the table is

$$E_\theta^\infty = \max_{j,n} \left|\Theta_j^n - \theta\left(\boldsymbol{x}_j, t^n\right)\right|, \quad \text{and} \quad E_I^\infty = \max_n(\operatorname{dist}(I(n\Delta t), I^n)).$$

The CPU times have been normalised such that the coarsest moving grid method corresponds to one unit. To obtain the same level of accuracy it is clear that the moving mesh method is considerably more efficient. Table II shows the results obtained using the h-adaptive and fixed grid methods of Nochetto *et al.* [21]. Here J denotes the average number of nodes used in the triangular meshes and these should be compared with N^2 for our moving mesh method. Although a strict comparison is difficult, in terms of equivalent degrees of freedom it is clear that the moving mesh solutions in most cases are more accurate than the h-adaptive solutions. As mentioned above, the moving mesh approach does not require any complicated data structure and is considerably simpler to apply than [21].

5.3. *The Formation of a Cusp*

For our third example, the initial condition is chosen such that the phase front forms a cusp. The domain $\Omega = (-2, 4) \times (0, 5)$ and $T = 1$. The initial temperature is given by

$$\theta_0(x, y, 0) = \begin{cases} 0.25(r^2 - 1), \quad r \le 1, y \ge 2; \quad 0.25(x^2 - 1), & |x| < 1, y < 2; \\ (r - 1), \quad r > 1, y \ge 2; \quad 5(|x| - 1), & |x| > 1, y < 1; \\ (|x| - 1)(3 - 2\cos\pi(y - 2)), & |x| > 1, 1 \le y < 2, \end{cases} \tag{5.2}$$

where $r = (x^2 + (y - 2)^2)^{1/2}$. A Dirichlet condition $\theta_D = \theta_0(1 + t)$ is imposed on $x = -2$, $x = 4$, $y = 5$, and a homogeneous Neumann condition is prescribed on $y = 0$. The exact solution to this problem is unknown but calculations performed by [7] and [21] suggest that the phase front forms a cusp at (0, 0) and eventually the solid phase disappears.

Figure 7 shows the computed grids and interface locations with $N = 30$, $Nt = 80$, $\tau = 0.1$, and $\varepsilon = 0.025$ which show excellent agreement with a uniform grid simulation with $N = 120$ and $Nt = 160$. Although the phase change interface forms a closed contour after $t = 0.65$ we see that this does not pose any difficulty with the mesh movement. Thereafter, the interface forms a cusp and again this potentially difficult interface shape is predicted well. After the solid phase disappears the grid automatically relaxes back towards a uniform mesh.

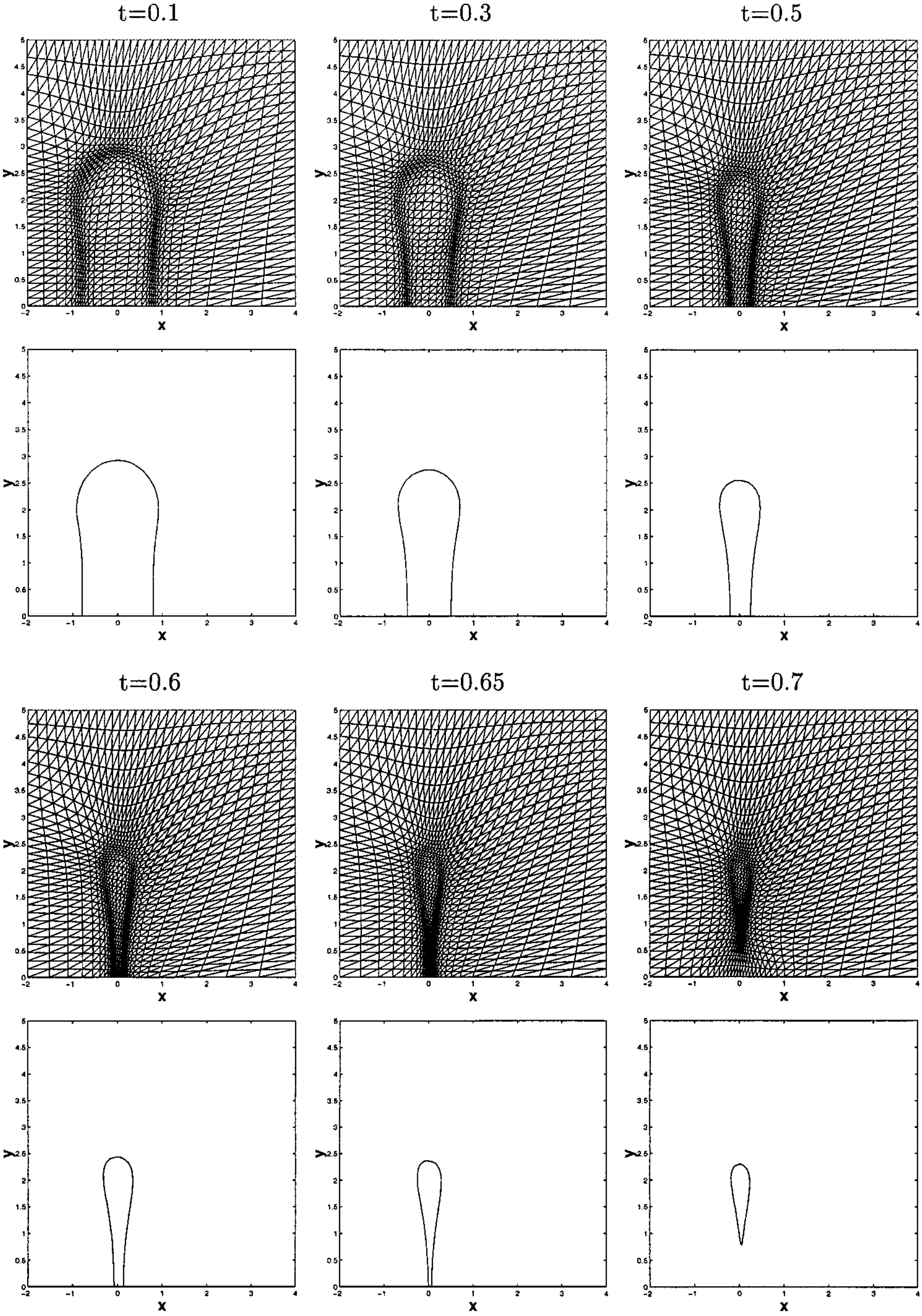

FIG. 7. Grids and interface predictions for cusp problem.

5.4. *An Oscillating Source*

For our final example the domain $\Omega = (-1, 1)^2$ and $T = 12$. The initial temperature $\theta(x, y, 0) = y/10$, the boundary conditions are $\theta(x, y, t) = y/10$ for the three sides $y > -1$, and a homogeneous Neumann condition is specified on the bottom side $y = -1$.

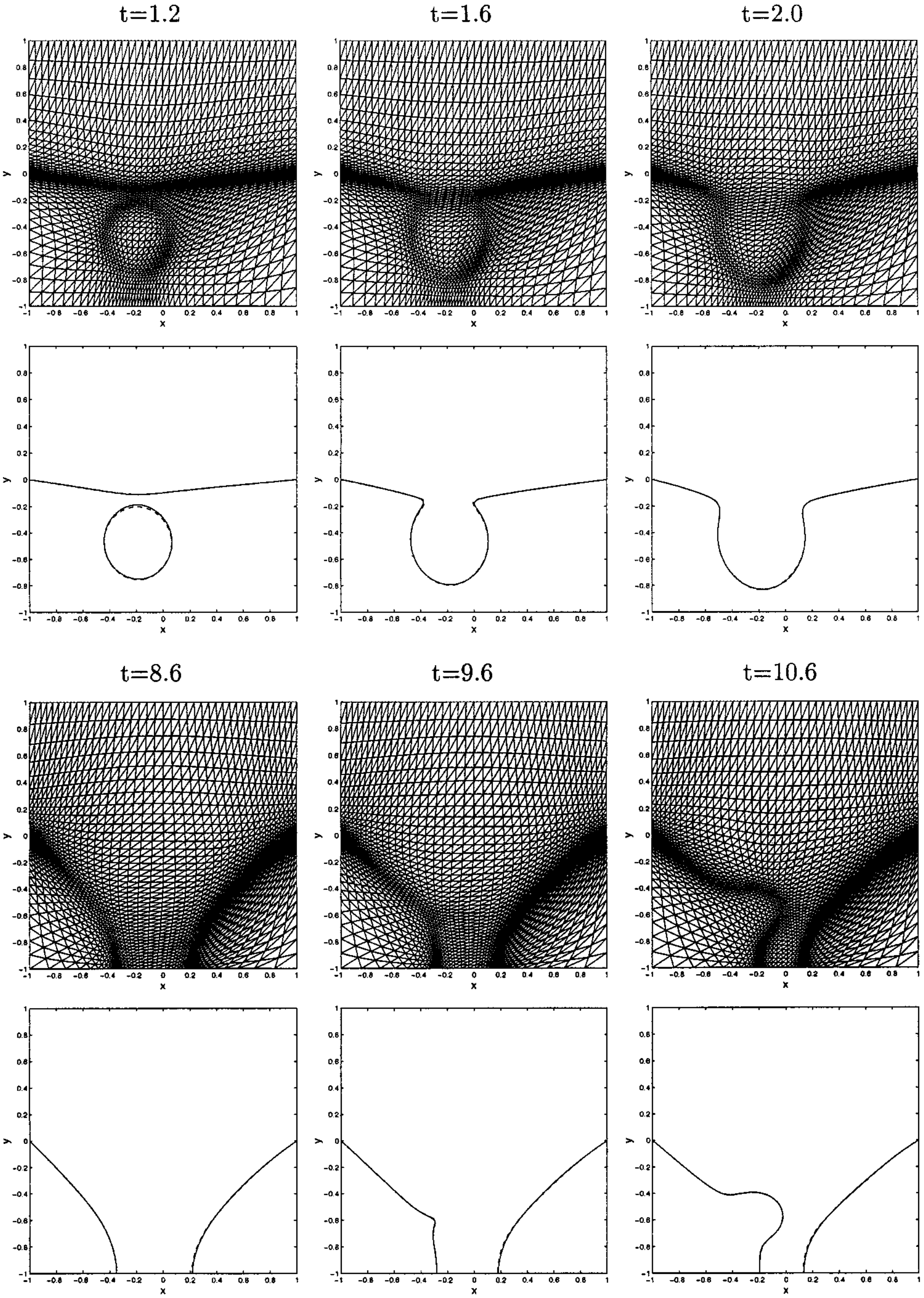

FIG. 8. Grids and interface predictions for oscillating source problem.

The evolution of the solution is driven by an oscillating heat source

$$
\begin{aligned}
f(\boldsymbol{x}, t) = {} & \cos(t/5)\max(0, 3.125 - 50|\boldsymbol{x} - (-1/5, -1/2)|^2) \\
& + \sin(t/5)\max(0, 3.125 - 50|\boldsymbol{x} - (-1/5, 1/2)|^2).
\end{aligned} \tag{5.3}
$$

The exact solution of this problem is unknown although it has been used as a test case by Nochetto *et al.* [22]. For this example we use the parameters $N = 40$, $Nt = 480$, $\tau = 0.1$, $\varepsilon = 0.005$, and the results are compared with a fine grid solution with $N = 120$, $Nt = 960$, and $\varepsilon = 0.005$. Figure 8 shows the development of two liquid phases that eventually merge. We note that at $t = 1.2$ the moving mesh has no difficulty adapting towards the two phase change interfaces. Furthermore, the mesh movement algorithm effectively deals with the change in topology as the two interfaces merge. The computed interface positions are very similar to those predicted using the far more complex h-adaptive method of [22].

6. CONCLUSIONS

In this paper we have presented a moving mesh finite element method for the enthalpy formulation of phase change problems. The algorithm is able to efficiently and accurately predict the evolution of the temperature field and the position of the phase front, even when it develops cusps or undergoes topological changes. The method is relatively simple and delivers comparable accuracy to more complicated h-adaptive schemes. Ideally, it makes sense to combine the two approaches and this is an area for future development. In addition, we believe the method is well suited to be applied to more sophisticated models of solidification, such as phase-field models, which account for important physical effects such as supercooling and surface tension.

ACKNOWLEDGMENTS

The authors thank D. Sloan (Strathclyde), W. Cao, R. D. Russell (Simon Fraser), and W. Huang (Kansas) for a number of useful discussions and suggestions. Support for G. Beckett was provided under EPSRC Grant GR/M26459. Partial support for J. A. Mackenzie while visiting R. D. Russell was provided under NSERC (Canada) Grant OGP-0008781.

REFERENCES

1. J. U. Brackbill, An adaptive grid with direction control, *J. Comput. Phys.* **108**, 38 (1993).
2. J. U. Brackbill and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions, *J. Comput. Phys.* **46**, 342 (1982).
3. H. Budhia and F. Kreith, Heat transfer with melting or freezing in a wedge, *Int. J. Heat Mass Transfer* **16**, 195 (1973).
4. W. Cao, W. Huang, and R. D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, *J. Comput. Phys.* **149**, 221 (1999).
5. W. Cao, W. Huang, and R. D. Russell, A study of monitor functions for two dimensional adaptive mesh generation, *SIAM J. Sci. Comput.* to appear.
6. N. Carlson and K. Miller, Design and application of a gradient-weighted moving finite element code. Part II. In two dimensions, *SIAM J. Sci. Comput.* **19**, 798 (1998).
7. S. Chen, B. Merriman, S. Osher, and P. Smereka, A simple level set method for solving Stefan problems, *J. Comput. Phys.* **135**(1), 8 (1997).
8. P. G. Ciarlet and P. A. Raviart, Maximum principle and uniform convergence for the finite element method, *Comput. Methods Appl. Mech. Eng.* **2**, 17 (1973).
9. J. Crank, *Free and Moving Boundary Value Problems* (Oxford Univ. Press, London 1984).
10. L. A. Crivelli and S. R. Idelsohn, A temperature-based finite element solution for phase-change problems, *Int. J. Numer. Methods Eng.* **23**, 99 (1986).

11. A. S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds. *J. Comput. Phys.* **95**, 450 (1991).

12. P. W. Egolf and H. Manz, Theory and modeling of phase change materials with and without mushy regions, *Int. J. Heat Mass Transfer* **37**, 2917 (1994).

13. Z.-X. Gong, Y.-F. Zhang, and A. S. Mujumdar, An efficient finite element procedure for the solidification of enthalpy model of phase change heat conduction problems, in *Computational Modelling of Free and Moving Boundary Problems. Vol 2. Heat Transfer*, edited by L. C. Wrobel and C. A. Brebbia. (WIT, Southhampton, 1991), pp. 93–104.

14. W. Huang and R. D. Russell, A high-dimensional moving mesh strategy, *Appl. Numer. Math.* **26**, 63 (1997).

15. W. Huang and R. D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, *SIAM J. Sci. Comput.* **20**(3), 998 (1999).

16. X. Lang, *Moving Mesh Control Volume Solution for Two-Dimensional Phase Change Problems in Enthalpy Formulation*, M. Sc. thesis (Kansas University, 1997).

17. D. R. Lynch and K. O'Neill, Continuously deforming finite elements for the solution of parabolic problems with and without phase, *Int. J. Numer. Meth. Eng.* **17**, 81 (1981).

18. J. A. Mackenzie and M. L. Robertson, The numerical solution of one-dimensional phase change problems using an adaptive moving mesh method, *J. Comput. Phys.* **161**, 537 (2000).

18a. K. Miller and R. N. Miller, Moving finite elements. I, *SIAM J. Numer. Anal.* **18**, 1019–1032 (1981).

19. G. H. Meyer, Multidimensional Stefan problems, *SIAM J. Numer. Anal.* **10**, 522 (1973).

20. R. H. Nochetto, Finite elements for free and moving boundary problems, in *SERC Numerical Analysis Summer School, Lancaster University* (1990).

21. R. H. Nochetto, M. Paolini, and 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**(5), 1207 (1991).

22. R. H. Nochetto, A. Schmidt, and C. Verdi, Error control for phase change problems, in *IVTAM Symposium on Variations of Domain and Free Boundary Problems, Paris* (Kluwer Academic, Dordrecht/Norwell, MA, 1997).

23. M. E. Rose, An enthalpy scheme for Stefan problems in several dimensions, *Appl. Numer. Math.* **12**, 229 (1993).

24. R. Viswanath and Y. Jaluria, A comparison of different solution methodologies for melting and solidification problems in enclosures, *Numer. Heat Transfer* **24**, 77 (1993).

25. V. Voller and M. Cross, Accurate solutions of moving boundary problems using the enthalpy method, *Int. J. Heat Mass Transfer* **24**, 545 (1981).

26. A. Winslow, Numerical solution of the quasilinear Poisson equation in a nonuniform mesh, *J. Comput. Phys.* **2**, 149 (1967).

27. M. Zerroukat and C. R. Chatwin, A finite-difference algorithm for multiple moving boundary value problems using real and virtual grid networks, *J. Comput. Phys.* **112**, 298 (1994).