COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. 6, No. 3, pp. 595-624

Commun. Comput. Phys.
September 2009

A Moving-Mesh Finite Element Method and its Application to the Numerical Solution of Phase-Change Problems

M. J. Baines[1], M. E. Hubbard[2,*], P. K. Jimack[2] and R. Mahmood[3]

[1] *Department of Mathematics, The University of Reading, UK.*
[2] *School of Computing, University of Leeds, UK.*
[3] *Computer Division, Pakistan Institute of Nuclear Science and Technology (PINSTECH), Islamabad, Pakistan.*

Received 29 August 2008; Accepted (in revised version) 24 December 2008

Available online 6 February 2009

Abstract. A distributed Lagrangian moving-mesh finite element method is applied to problems involving changes of phase. The algorithm uses a distributed conservation principle to determine nodal mesh velocities, which are then used to move the nodes. The nodal values are obtained from an ALE (Arbitrary Lagrangian-Eulerian) equation, which represents a generalization of the original algorithm presented in *Applied Numerical Mathematics*, **54**:450–469 (2005). Having described the details of the generalized algorithm it is validated on two test cases from the original paper and is then applied to one-phase and, for the first time, two-phase Stefan problems in one and two space dimensions, paying particular attention to the implementation of the interface boundary conditions. Results are presented to demonstrate the accuracy and the effectiveness of the method, including comparisons against analytical solutions where available.

AMS subject classifications: 35R35, 65M50, 65M60, 76M10, 76T99

Key words: Moving mesh method, finite elements, multiphase flows, interface tracking.

1 Introduction

Moving-mesh methods have been used for the solution of partial differential equations (PDEs) in recent years in various ways. The motivation is usually to improve the resolution of solutions [4, 14, 17, 19, 20, 29, 38], to track special features of a solution (such as shocks, singularities and moving boundaries) [2, 5–7, 10, 18, 30–33, 42–44], and/or to

[*] Corresponding author. *Email addresses:* m.j.baines@reading.ac.uk (M. J. Baines), meh@comp.leeds.ac.uk (M. E. Hubbard), pkj@comp.leeds.ac.uk (P. K. Jimack), dr_rashid_mahmood@yahoo.com (R. Mahmood)

exploit geometric properties such as scale-invariance, orderings or asymptotics [8,12,15]. The moving-mesh methods can be divided into two broad categories: those based on mappings between a fixed computational mesh and physical space [10, 16, 17, 38] and those based on velocities expressed in terms of the mesh coordinates in physical space [6, 12, 18–20, 34]. In this paper we focus on one particular velocity-based moving-mesh finite element method [6–8], which is related to the Geometric Conservation Law [18,40]. This method has been successfully applied to a range of time-dependent nonlinear PDE problems involving singularities and implicit moving boundaries: however, since its original introduction in [6], a number of improvements have been made which are incorporated into the description below.

The algorithm has been designed to be very general in nature and applicable to a large family of problems. Having validated the proposed method for a range of situations, the main thrust of this paper is its application, for the first time, to phase-change problems involving an internal moving boundary. Many numerical schemes have been applied to such problems, including those involving moving-mesh techniques [10, 25, 26, 33, 39]. The application of the moving-mesh finite element method described in this paper to this problem is however new, and is shown to be both accurate and robust.

The layout of the paper is as follows. Section 2 gives a description of the moving-mesh finite element method of [6] incorporating a number of recent developments. Section 3.1 presents validating results from a mass-conserving problem using the porous medium equation [41], which is a second order nonlinear diffusion equation for which simple self-similar solutions exist with finite support that grows with time. Section 3.2 then describes validating results from a non-mass-conserving implicit moving boundary problem, a simple model of absorption and diffusion, referred to here as the Crank-Gupta problem [24], which contains a sink term which causes the finite support to shrink with time. Section 4 contains the main application of the moving-mesh finite element method to one-phase and two-phase Stefan problems and Section 5 contains details of the results of numerical experiments, validated against analytic solutions whenever possible. Finally, Section 6 includes a discussion of points raised in the paper.

2 A moving-mesh finite element method

In this section we present a derivation of the equations used by the moving-mesh finite element method to find mesh velocity potentials which are subsequently used to define the nodal velocities. This is followed by a summary of the complete algorithm, as used in this paper, and the enhancements which have been made since its original publication in [6].

2.1 A mesh movement velocity potential

Let $u(\mathbf{x},t)$ be the solution of a well-posed time-dependent nonlinear PDE problem of general form

$$\frac{\partial u}{\partial t}=\mathcal{L}u \tag{2.1}$$

on a finite time-dependent domain $\mathcal{R}(t)$, where $\mathcal{L}$ is a differential operator involving space derivatives only, with consistent boundary conditions on the (in general moving) boundary $\partial\mathcal{R}(t)$.

An important quantity in our approach is the total integral of the dependent variable

$$\theta(t)=\int_{\mathcal{R}(t)} u\,d\mathbf{x}. \tag{2.2}$$

For brevity and convenience we will refer to $\theta(t)$ as the "mass" in the domain at time t. Note however that, as in Section 4 for example, the dependent variable may be related to a quantity other than density.

A moving polygonal or polyhedral approximation $\overline{\mathcal{R}}(t)$ to $\mathcal{R}(t)$, with boundary $\partial\overline{\mathcal{R}}(t)$ approximating $\partial\mathcal{R}(t)$, is assumed, together with a moving tessellation of simplices of $\overline{\mathcal{R}}(t)$. A piecewise linear approximation U is also assumed, which can be expanded in terms of piecewise linear basis functions $W_i(t,\mathbf{x})$ forming a partition of unity and moving with the domain. The total discrete mass $\Theta(t)\approx\theta(t)$ is defined by (cf. (2.2))

$$\Theta(t)=\int_{\overline{\mathcal{R}}(t)} U\,d\mathbf{x}. \tag{2.3}$$

Now consider the distribution of the mass (2.3) across the domain $\overline{\mathcal{R}}(t)$. Each basis function $W_i(t,\mathbf{x})$ is associated with a partial mass and a proportion C_i, defined by

$$\int_{\overline{\mathcal{R}}(t)} W_i U\,d\mathbf{x}=C_i\Theta(t)=C_i\int_{\overline{\mathcal{R}}(t)} U\,d\mathbf{x}. \tag{2.4}$$

A moving-mesh numerical method can be created by keeping the proportions $C_i\in[0,1]$ constant as time progresses. The fact that the weight functions $W_i(t,\mathbf{x})$ are non-negative and form a partition of unity means that conservation of the partial masses (2.4) is consistent with total mass conservation. (Clearly $\sum C_i=1$ is required to maintain the correct total mass and this also follows from $\sum W_i\equiv 1$.)

The velocity $\mathbf{V}(t,\mathbf{x})$ of the moving frame can be derived from a weak form of the Reynolds Transport Theorem [6],

$$\begin{aligned}\frac{d}{dt}\left[\int_{\overline{\mathcal{R}}(t)} W_i U\,d\mathbf{x}\right]&=\int_{\overline{\mathcal{R}}(t)}\left\{\frac{\partial(W_iU)}{\partial t}+\nabla\cdot(W_iU\mathbf{V})\right\}d\mathbf{x}\\&=\int_{\overline{\mathcal{R}}(t)} W_i\left\{\frac{\partial U}{\partial t}+\nabla\cdot(U\mathbf{V})\right\}d\mathbf{x},\end{aligned} \tag{2.5}$$

since we have assumed that the basis functions $W_i(t,\mathbf{x})$ move with the domain $\overline{\mathcal{R}}(t)$ and are therefore advected with velocity $\mathbf{V}$, i.e.,

$$\frac{\partial W_i}{\partial t}+\mathbf{V}\cdot\nabla W_i=0. \tag{2.6}$$

From (2.5) and a weak form of (2.1) a moving weak form of the PDE can be written as

$$\frac{d}{dt}\left[\int_{\overline{\mathcal{R}}(t)} W_i U \, d\mathbf{x}\right] - \int_{\overline{\mathcal{R}}(t)} W_i \nabla\cdot(U\mathbf{V})\, d\mathbf{x} = \int_{\overline{\mathcal{R}}(t)} W_i \mathcal{L} U \, d\mathbf{x}, \tag{2.7}$$

where the U on the right hand side may need to be regularized to lie in the domain of $\mathcal{L}$.

Hence, applying (2.4) with constant C_i, Eq. (2.7) gives

$$C_i\dot{\Theta} - \int_{\overline{\mathcal{R}}(t)} W_i \nabla\cdot(U\mathbf{V})\, d\mathbf{x} = \int_{\overline{\mathcal{R}}(t)} W_i \mathcal{L} U \, d\mathbf{x}. \tag{2.8}$$

A unique solution of Eq. (2.8) for $\mathbf{V}$ in multiple space dimensions is obtained if the curl of $\mathbf{V}$ is prescribed in $\overline{\mathcal{R}}(t)$ and the normal component of $\mathbf{V}$ is prescribed on $\partial\overline{\mathcal{R}}(t)$, by Helmholtz' Theorem. For ease of exposition we shall assume that *curl* $\mathbf{V}=0$, so that $\mathbf{V}=\nabla\Phi$, where Φ is a scalar velocity potential.

Instead of (2.8), Φ and $\dot{\Theta}$ now satisfy the equation

$$C_i\dot{\Theta} - \int_{\overline{\mathcal{R}}(t)} W_i \nabla\cdot(U\nabla\Phi)\, d\mathbf{x} = \int_{\overline{\mathcal{R}}(t)} W_i \mathcal{L} U \, d\mathbf{x} \tag{2.9}$$

or, after integration by parts,

$$C_i\dot{\Theta} - \oint_{\partial\overline{\mathcal{R}}(t)} W_i U \nabla\Phi\cdot\mathbf{n}\, ds + \int_{\overline{\mathcal{R}}(t)} U\nabla\Phi\cdot\nabla W_i \, d\mathbf{x} = \int_{\overline{\mathcal{R}}(t)} W_i \mathcal{L} U \, d\mathbf{x}, \tag{2.10}$$

where $\mathbf{n}$ is the unit outward normal to $\partial\overline{\mathcal{R}}(t)$, the boundary of the domain.

Summing Eq. (2.10) over all of the mesh nodes gives

$$\dot{\Theta} - \oint_{\partial\overline{\mathcal{R}}(t)} U\nabla\Phi\cdot\mathbf{n}\, ds = \int_{\overline{\mathcal{R}}(t)} \mathcal{L} U \, d\mathbf{x}, \tag{2.11}$$

which, in situations where $U\mathbf{V}\cdot\mathbf{n}$ is known on the entire boundary, can be used to give an explicit expression for $\dot{\Theta}$. However, for greater generality, the rate of change of mass $\dot{\Theta}$ is treated here as an unknown.

Combining (2.10) with (2.11) yields a square system of equations of reduced rank, since Eq. (2.11) is the sum of Eq. (2.10). Although the combined system has a single infinity of solutions, a unique solution for Φ and $\dot{\Theta}$ may be obtained by specifying the value of Φ at one of the nodes and ignoring the corresponding equation. Since Φ is a velocity potential there is no resulting loss of generality. Although we do not do so in this paper, a further option is to impose a constant value of Φ on the whole of $\partial\overline{\mathcal{R}}(t)$, which has the effect of weakly specifying zero tangential velocity of the boundary nodes.

The nodal velocities $\mathbf{V}$ may be recovered from Φ by minimizing the L_2 norm of each component of $\mathbf{V}-\nabla\Phi$, leading to the vector equation

$$\int_{\overline{\mathcal{R}}(t)} W_i \mathbf{V}\, d\mathbf{x} = \int_{\overline{\mathcal{R}}(t)} W_i \nabla\Phi\, d\mathbf{x}, \tag{2.12}$$

for all nodes i. Having found $\mathbf{V}$ and $\dot{\Theta}$, the nodal positions $\mathbf{X}$ as well as Θ can be advanced in time using any convenient time-stepping scheme, for example Heun's scheme.

The values of the solution U can then be recovered either by direct inversion of Eq. (2.4) or, more generally, by the use of a conservative Arbitrary Lagrangian-Eulerian (ALE) equation of the form

$$\frac{d}{dt}\left[\int_{\overline{\mathcal{R}}(t)} W_i U \, d\mathbf{x}\right] = \int_{\overline{\mathcal{R}}(t)} W_i \left(\mathcal{L}U + \nabla\cdot(U\mathbf{V})\right) d\mathbf{x} \tag{2.13}$$

(cf. Eq. (2.7)), which gives the consistent form of the integral

$$\int_{\overline{\mathcal{R}}(t)} W_i U \, d\mathbf{x} \tag{2.14}$$

prior to its inversion for U (but see Eqs. (2.15) and (2.16) below).

So far, boundary conditions for the dependent variable U have not been discussed. In circumstances where the methodology is designed to conserve mass (with appropriate boundary conditions on $\mathbf{V}$), the numerical method should do the same with the discrete form of the mass. However, the standard way of imposing Dirichlet boundary conditions in the finite element inversion of (2.14) for U destroys mass conservation in general (since the test functions corresponding to nodes on the Dirichlet boundary are not used and so the remaining test functions no longer form a partition of unity). In [29] a modification is proposed which retains mass conservation in the presence of strongly imposed Dirichlet boundary conditions, and moreover is shown to improve the accuracy of the method. This is achieved by defining a set of modified test functions $\widetilde{W}_i$ used in the recovery of the nodal values of the dependent variable U.

We shall therefore recover U values from modified partial masses $\Psi_i(t)$, defined by (cf. Eq. (2.4))

$$\Psi_i(t) = \int_{\overline{\mathcal{R}}(t)} \widetilde{W}_i U \, d\mathbf{x}, \tag{2.15}$$

using the discrete form of the ALE equation (cf. Eq. (2.13))

$$\dot{\Psi}_i = \int_{\overline{\mathcal{R}}(t)} \widetilde{W}_i \left(\mathcal{L}U + \nabla\cdot(U\mathbf{V})\right) d\mathbf{x} \tag{2.16}$$

(for those mesh nodes at which Dirichlet boundary conditions are not applied). Although they both represent rates of change of partial masses associated with mesh nodes, $C_i\dot{\Theta}$ and $\dot{\Psi}_i$ are different quantities in the discrete algorithm. The former are calculated to be consistent with the velocity field represented by $\nabla\Phi$ while the latter are consistent with the actual nodal velocities $\mathbf{V}$, as derived using (2.12). The rate of change of total mass of the system is governed by the ALE equation (2.16) used to update the dependent variable so, from now on,

$$\Psi(t) = \int_{\overline{\mathcal{R}}(t)} U \, d\mathbf{x} \tag{2.17}$$

will be used to denote the total mass of the discretized system.

2.2 Algorithm

The algorithm used in this paper for a single time-step can now be summarized as follows:

1. Given a set of mesh nodes with positions $\mathbf{X}_i$ and a distribution of the total mass Ψ, defined by the masses Ψ_i of (2.15), to all those mesh nodes where Dirichlet boundary conditions are *not* applied, invert (2.15) to recover the solution values U at these nodes. (The Dirichlet conditions on U are imposed at the remaining nodes.)
2. Having found the nodal values U_i, solve the set of Eqs. (2.10) and (2.11) for the velocity potentials Φ_i and for $\dot{\Theta}$, imposing Φ at one mesh node. Note that the constants C_i must be recalculated from (2.4) before this step can be carried out.
3. Use the resulting values of Φ_i to obtain the mesh node velocities $\mathbf{V}_i$ from (2.12) for all mesh nodes, either with or without imposing boundary conditions on $\mathbf{V}$.
4. Knowing $\mathbf{V}$ $(=\dot{\mathbf{X}})$, obtain the corresponding adjustments to the local masses (2.15) using the ALE equation (2.16), thus computing the $\dot{\Psi}_i$ values.

These four steps can be thought of as evaluating a vector function of the form $\vec{F}(\vec{\mathbf{X}},\vec{\Psi})$ (the arrows indicating a vector which contains the values stored at all the nodes of the mesh) which satisfies

$$\frac{d(\vec{\mathbf{X}},\vec{\Psi})}{dt}=\vec{F}(\vec{\mathbf{X}},\vec{\Psi}). \tag{2.18}$$

This system of ordinary differential equations in time can then be integrated using any convenient time-stepping scheme. Here, following the investigation presented in [29], Heun's scheme is used. Experiments recorded there demonstrated that Heun's scheme ensures that the error in the temporal discretization is negligible compared to the error in the spatial discretization for almost any (stable) time-step which avoids mesh tangling.

2.3 Comparison

The moving-mesh finite element method presented above is essentially that presented in [6] but with the following modifications:

- Instead of recovering the values of U on the new mesh directly from the conservation principle (2.4), an ALE equation (2.16) is used prior to using (2.15) to update U. This allows the extension of the method to more general monitor functions.
- The partial masses Ψ_i used to drive the mesh movement are updated using the ALE equation (2.16), so the distribution of the mass defined by the coefficients C_i in (2.4) is only preserved approximately at each time-step. This is a consequence of the previous modification.
- The rate of change of the conserved quantity $\dot{\Theta}$ is treated in Eqs. (2.10) and (2.11) as an unknown and calculated at each time-step (although now it is only ever used as a means to find Φ). This provides additional generality since $\dot{\Theta}$ may not always be available explicitly.

• Dirichlet boundary conditions are imposed strongly in a way which conserves mass when the underlying system does, using the modified test functions $\widetilde{W}_i$ of [29].

• To ensure that the system has a unique solution when solving for the mesh velocity potentials, Φ is imposed at just one mesh node, whereas in the previous work Φ was always specified over the whole boundary. This removes a potentially unnecessary constraint on Φ.

• Heun's scheme, an explicit, two-stage, second order, TVD Runge-Kutta method is used to discretize the time derivative instead of the forward Euler scheme used in [6]. This offers improved temporal accuracy [29].

3 Method validation

Two sets of numerical tests have been carried out to validate the method, in one and two space dimensions. The first set of tests is carried out on the porous medium equation, a second order nonlinear diffusion problem for which mass is conserved globally. The second set is on a linear diffusion equation with a source/sink term representing absorption, for which mass is not conserved. The results are validated against exact solutions and can be compared with those presented in [6].

Both sets of calculations have been carried out on meshes which are initially uniform, and for which the representative initial mesh size is repeatedly halved to estimate orders of accuracy. Mesh tangling is avoided by dividing the time-step by four with each refinement. The initial conditions sample the exact nodal values of the dependent variable at the nodes.

In Sections 3 and 4 it will be necessary to distinguish between different types of boundary. Fixed parts of the boundary will be denoted by Γ_F, at which the condition $\mathbf{V}\cdot\mathbf{n}=0$ is imposed, where $\mathbf{n}$ is a normal to the boundary. Implicit moving parts of the boundary, where no such condition is imposed, will be denoted by $\Gamma_M(t)$. In both cases the boundary conditions imposed on U will depend on the problem being solved.

Furthermore, in all of the test cases presented, a condition for $U\nabla\Phi\cdot\mathbf{n}$ is given on every boundary. Hence, $\dot{\Theta}$ is actually known explicitly, and it is only necessary to specify Φ at one point. Nevertheless, for the sake of generality, we choose to treat $\dot{\Theta}$ as an unknown when calculating the Φ_i and we thus specify $\Phi=0$ at one point. This point can be chosen arbitrarily: it is taken here to be the first node listed in the array containing the mesh data.

3.1 A mass-conserving nonlinear diffusion problem

The porous medium equation, details of which can be found in [3,41], is given by

$$\frac{\partial u}{\partial t}=\nabla\cdot(u^n\nabla u)\qquad \text{in } \mathcal{R}(t), \tag{3.1}$$

where $n>0$ is an integer exponent. A Dirichlet boundary condition $u=0$ is imposed on the moving boundary $\Gamma_M(t)$. For this problem it can be shown that mass, as defined by (2.2), is conserved.

For this problem the operator $\mathcal{L}u\equiv\nabla\cdot(u^n\nabla u)$ and Eq. (2.10) gives

$$C_i\dot{\Theta}+\int_{\mathcal{R}(t)}U\nabla\Phi\cdot\nabla W_i\,d\mathbf{x}=\oint_{\Gamma_F}W_iU^n\nabla U\cdot\mathbf{n}\,ds-\int_{\mathcal{R}(t)}U^n\nabla W_i\cdot\nabla U\,d\mathbf{x} \tag{3.2}$$

since $U=0$ on the moving boundary $\Gamma_M(t)$ and $\mathbf{V}\cdot\mathbf{n}=0$ on Γ_F. Similarly, Eq. (2.11) becomes

$$\dot{\Theta}=\oint_{\Gamma_F}U^n\nabla U\cdot\mathbf{n}\,ds. \tag{3.3}$$

Equations (3.2) and (3.3) give $\dot{\Theta}$ and the velocity potential Φ, which is used in (2.12) to find the mesh velocity $\mathbf{V}=\dot{\mathbf{X}}$. The discrete ALE equation (2.16) becomes

$$\frac{d\Psi_i}{dt}=\int_{\mathcal{R}(t)}\widetilde{W}_i\{\nabla\cdot(U^n\nabla U)+\nabla\cdot(U\mathbf{V})\}\,d\mathbf{x} \tag{3.4}$$

which, after integration by parts and imposition of the boundary conditions, gives

$$\frac{d\Psi_i}{dt}=\oint_{\Gamma_F}\widetilde{W}_iU^n\nabla U\cdot\mathbf{n}\,ds-\int_{\mathcal{R}(t)}\nabla\widetilde{W}_i\cdot(U^n\nabla U+U\mathbf{V})\,d\mathbf{x}. \tag{3.5}$$

Heun's method is used to update $\vec{\mathbf{X}}$ and $\vec{\Psi}$, and inversion of Eq. (2.15) to recover the dependent variable U. In all of the following cases $\Gamma_F=\emptyset$, so the boundary integrals in Eqs. (3.2)-(3.5) disappear and $\dot{\Theta}=0$ is known explicitly.

Equation (3.1) admits a family of exact compactly supported similarity solutions with moving boundaries on which $u=0$ [9, 35]. In d space dimensions a radially-symmetric self-similar solution exists of the form

$$u(r,t)=\begin{cases}\frac{1}{\lambda^d(t)}\left(1-\left(\frac{r}{r_0\lambda(t)}\right)^2\right)^{\frac{1}{n}}, & r\le r_0\lambda(t),\\ 0, & r>r_0\lambda(t),\end{cases} \tag{3.6}$$

where $r\ge 0$ is the usual radial coordinate, and where

$$\lambda(t)=\left(\frac{t}{t_0}\right)^{\frac{1}{2+dn}}\qquad\text{and}\qquad t_0=\frac{{r_0}^2n}{2(2+dn)}. \tag{3.7}$$

The problem is parameterized by the initial front position r_0 (at time t_0) and the position of the moving front is given by $r_0\lambda(t)$.

Two instances of (3.1) are considered here, one with exponent $n=1$ (for which the slope of the self-similar solution normal to the moving boundary is finite) and another with $n=3$ (for which the slope normal to the boundary is infinite). The test cases are run until time $T=t-t_0$, as detailed below. Results have been computed in both one and two space dimensions for each test case as follows.

- One dimension, $n=1$, $r_0=0.5$, $t_0=0.04167$, run until $T=10$ (when $r\approx 3.11154$).
- One dimension, $n=3$, $r_0=0.5$, $t_0=0.075$, run until $T=10$ (when $r\approx 1.33231$).
- Two dimensions, $n=1$, $r_0=0.5$, $t_0=0.03125$, run until $T=0.1$ (when $r\approx 0.71578$).
- Two dimensions, $n=3$, $r_0=0.5$, $t_0=0.046875$, run until $T=0.1$ (when $r\approx 0.57673$).

The values of various discrete error measures are shown in Figs. 1 and 2. The results exhibit approximately second order accuracy when $n=1$, which reduces to first order when $n=3$, as in [6].

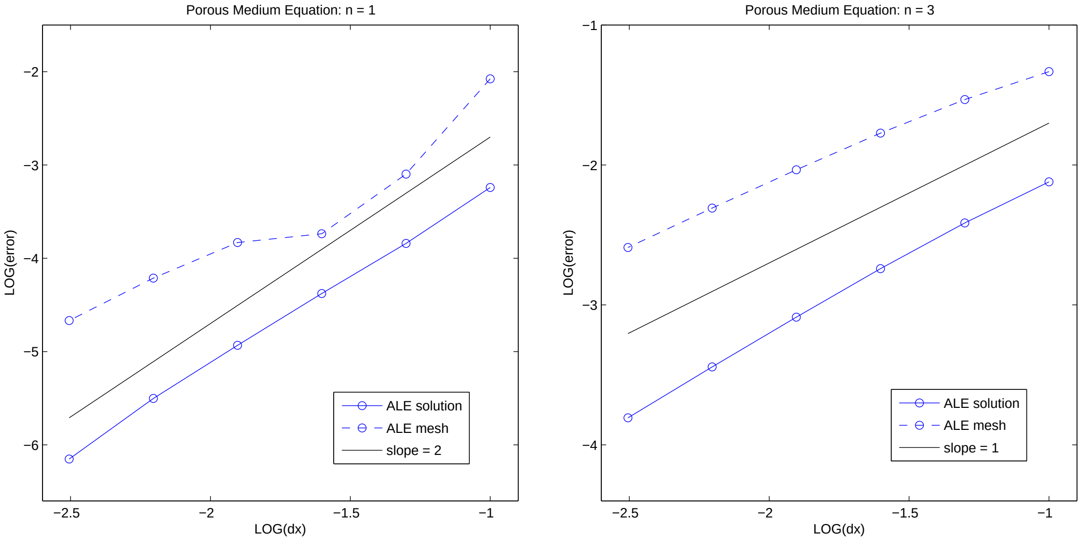

Figure 1: Comparison of L^2 errors in the solution and the magnitudes of the errors in the boundary node positions for the porous medium equation in one space dimension with $n=1$ (left) and $n=3$ (right).

3.2 A non-mass-conserving diffusion problem with a source

The Crank-Gupta problem [11,24,36] provides an example of a problem where the total mass of the system changes due to the presence of a source/sink term. The problem is defined by the differential equation

$$\frac{\partial u}{\partial t}=\nabla^2 u-1 \qquad \text{in } \mathcal{R}(t), \tag{3.8}$$

with boundary conditions $u=0$ and $\partial u/\partial n=0$ on the moving boundary $\Gamma_M(t)$.

For this problem $\mathcal{L}u\equiv\nabla^2 u-1$ and, after integration by parts, Eq. (2.10) becomes

$$\begin{aligned} &C_i\dot{\Theta}+\int_{\mathcal{R}(t)} U\nabla\Phi\cdot\nabla W_i\,d\mathbf{x} \\ =&\oint_{\Gamma_F} W_i\nabla U\cdot\mathbf{n}\,ds-\int_{\mathcal{R}(t)}\nabla W_i\cdot\nabla U\,d\mathbf{x}-\int_{\mathcal{R}(t)} W_i\,d\mathbf{x}, \end{aligned} \tag{3.9}$$

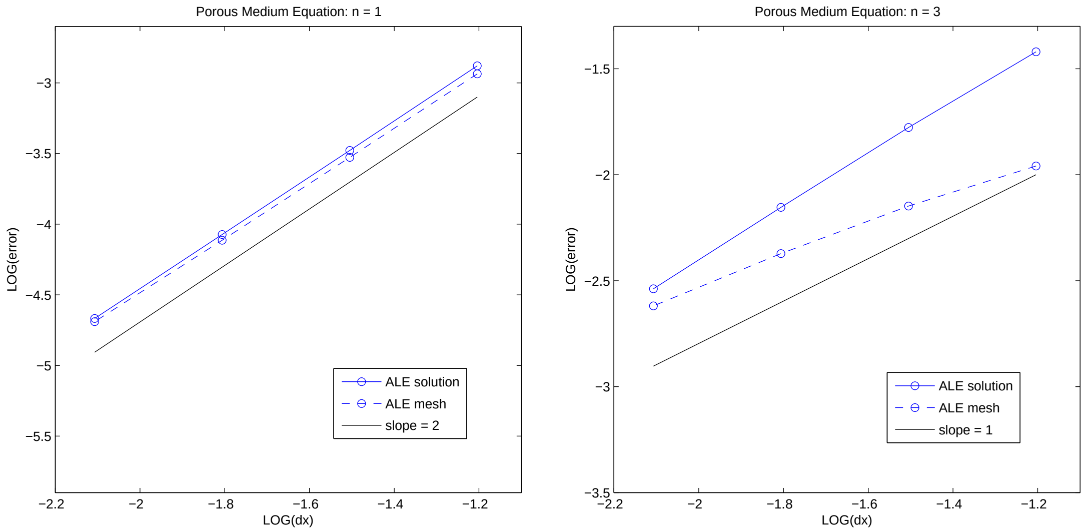

Figure 2: Comparison of L^2 errors in the solution and boundary node positions for the porous medium equation in two space dimensions with $n=1$ (left) and $n=3$ (right).

since $\mathbf{V}\cdot\mathbf{n}=0$ on the fixed boundary and $U=\nabla U\cdot\mathbf{n}=0$ at the moving boundary. Furthermore, summing (3.9) over the whole domain gives

$$\dot{\Theta}=\oint_{\Gamma_F} W_i \nabla U\cdot\mathbf{n}\,ds-\int_{\overline{\mathcal{R}}(t)} d\mathbf{x}. \tag{3.10}$$

The discrete ALE equation (2.16), after integration by parts and imposition of the boundary conditions, gives

$$\frac{d\Psi_i}{dt}=\oint_{\Gamma_F}\widetilde{W}_i\nabla U\cdot\mathbf{n}\,ds-\int_{\overline{\mathcal{R}}(t)}\nabla\widetilde{W}_i\cdot(\nabla U+U\mathbf{V})\,d\mathbf{x}-\int_{\overline{\mathcal{R}}(t)}\widetilde{W}_i\,d\mathbf{x}. \tag{3.11}$$

Heun's method is again used to update $\mathbf{X}$ and Ψ_i, after which Eq. (2.15) is used to recover U. The results summarized below are obtained with U_x specified at $x=0$.

In [11] an exact solution to the one-dimensional equation on the interval $[0,x(t)]$ with the same boundary conditions at the moving boundary, but with $u_x(0,t)=-1+e^{t-1}$ at $x=0$, is given as

$$u(x,t)=\begin{cases}-x-t+e^{x+t-1}, & x\le 1-t,\\ 0, & x>1-t,\end{cases} \tag{3.12}$$

with initial condition $u(x,0)=-x+e^{x-1}$ for $x\in[0,1]$ at $t=0$. Results for this test case have been compared with the exact solutions at $t=0.6$. The values of various error measures are shown in Fig. 3. The results match those of [6], giving second order accurate approximations to both solution and boundary position.

Two-dimensional calculations are not presented here due to the lack of an analytical solution to compare with. It is simply noted that the new scheme appears to be more

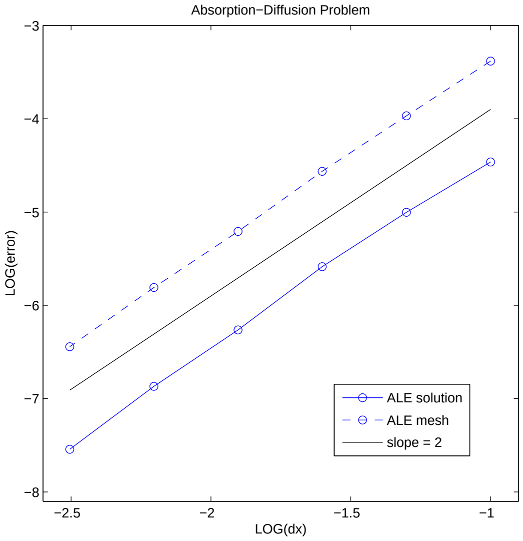

Figure 3: Comparison of L^2 errors in the solution and the magnitudes of the errors in the boundary node positions for the absorption-diffusion problem in one space dimension.

robust than the original one proposed in [6], since it is possible to run the numerical experiment described there for a significantly longer time with the improved method, taking the solution far closer to the point where the mass vanishes before mesh tangling occurs.

We now turn attention to the main purpose of this paper, which is to demonstrate that the moving-mesh finite element method described here may be applied to problems with change of phase at internal boundaries, in addition to the problems with exterior moving boundaries already considered.

4 Stefan problems and changes of phase

The two-phase problem will be presented first, for which the phase-change is modeled by applying the Stefan condition at a moving *internal* boundary. The single-phase problem will then be presented as a special, simpler, case, which can be used for validation purposes in both one and two space dimensions since exact solutions are available for comparison.

4.1 The two-phase problem

Consider a domain divided into regions of solid or liquid phase, denoted respectively by $\mathcal{R}_S(t)$ and $\mathcal{R}_L(t)$. This is illustrated for a domain split into two regions in Fig. 4. As indicated, the boundary of the solid region is divided into two components, the moving interface with the liquid phase $\Gamma_M(t)$ and a fixed part Γ_{F_S}, which in Fig. 4 coincides with

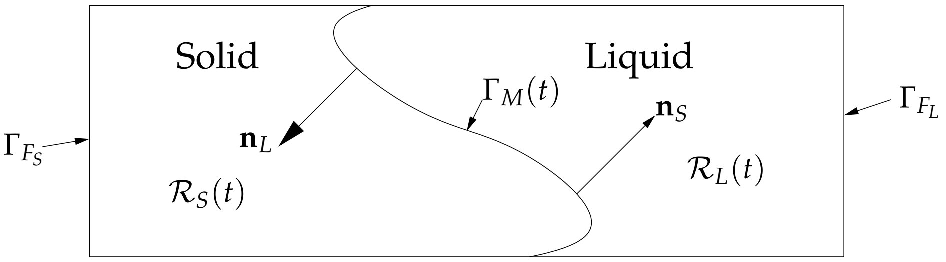

Figure 4: A representative domain, with our notation, for the two-phase Stefan problem.

the external boundary of the domain. The boundary of the liquid region is similarly divided between $\Gamma_M(t)$ and Γ_{F_L}. For simplicity, only two regions will be considered: all of the test cases investigated here (in two dimensions) have one region completely inside the other, although it should be noted that this is not a restriction of the method.

The two-phase Stefan problem (described in detail in [23]) models the transition of some substance between liquid and solid phases. This process takes place across an interface in the interior of the problem domain which moves as time progresses. It is modeled by the equations [13]

$$\begin{aligned} K_S u_t &= \nabla\cdot(k_S \nabla u) \qquad \text{in } \mathcal{R}_S(t), \\ K_L u_t &= \nabla\cdot(k_L \nabla u) \qquad \text{in } \mathcal{R}_L(t), \end{aligned} \tag{4.1}$$

in which K_S,K_L represent the volumetric heat capacities, k_S,k_L the thermal conductivities and u the temperature. The interface conditions are $u=u_M$ and, based upon an energy balance across the phase-change boundary $\Gamma_M(t)$,

$$k_S \nabla u_S\cdot\mathbf{n} - k_L \nabla u_L\cdot\mathbf{n} = \lambda \mathbf{v}\cdot\mathbf{n}, \tag{4.2}$$

where $\mathbf{n}=\mathbf{n}(t)$ is a normal to the moving interface, λ is the heat of phase-change per unit volume, and $\mathbf{v}\cdot\mathbf{n}$ is the normal velocity of the interface. All parameters are assumed here to be constant within their respective phases. In the (Stefan) condition (4.2), the derivative $\nabla u\cdot\mathbf{n}$ is discontinuous across the moving interface (although u is continuous) and subscripts are used to indicate the phase in which the gradient is evaluated. The fixed boundaries are treated appropriately, according to each individual test case.

Note that the entire problem (including the evolution of the interface) is unchanged by the addition of a constant value to u throughout the domain (as well as to u_M and the Dirichlet boundary conditions on Γ_{F_S} and Γ_{F_L}).

4.2 The numerical method

A moving polygonal approximation $\overline{\mathcal{R}}(t)$ to $\mathcal{R}(t)=\mathcal{R}_S(t)\cup\mathcal{R}_L(t)$ is set up, consisting of a moving tessellation of simplices with piecewise linear fixed and moving boundaries approximating $\Gamma_F=\Gamma_{F_S}\cup\Gamma_{F_L}$ and $\Gamma_M(t)$, respectively. The individual stages of the method can be described as follows.

4.2.1 Calculating the mesh velocity potential

Differentiating the weak form (2.4) with respect to time still results in (2.10), but substitution for $\mathcal{L}u$ from Eq. (4.1) leads to

$$\begin{aligned} &C_i\dot{\Theta} - \oint_{\Gamma_M(t)} W_i U \nabla\Phi\cdot\mathbf{n}\, ds + \int_{\overline{\mathcal{R}}(t)} U\nabla\Phi\cdot\nabla W_i\, d\mathbf{x} \\ &= \kappa \oint_{\Gamma_F} W_i \nabla U\cdot\mathbf{n}\, ds - \kappa \int_{\overline{\mathcal{R}}(t)} \nabla W_i\cdot\nabla U\, d\mathbf{x} \end{aligned} \tag{4.3}$$

after integration by parts, where $\kappa = k/K$ takes a different value in each of the two phases.

We consider the solid and liquid phases separately. Dividing the domain between its component phases leads to two discrete "masses" (cf. Eq. (2.3)),

$$\Theta_S(t) = \int_{\overline{\mathcal{R}}_S(t)} U\, d\mathbf{x} \qquad \text{and} \qquad \Theta_L(t) = \int_{\overline{\mathcal{R}}_L(t)} U\, d\mathbf{x} \tag{4.4}$$

and, since

$$\mathbf{V}\cdot\mathbf{n} = 0 \quad \text{on } \Gamma_F \qquad \text{and} \qquad U = U_M \quad \text{on } \Gamma_M(t),$$

the two corresponding equations (4.3) for the velocity potentials are, in the solid region,

$$\begin{aligned} &C_i^S\dot{\Theta}_S - U_M \oint_{\Gamma_M(t)} W_i \nabla\Phi\cdot\mathbf{n}_S\, ds + \int_{\overline{\mathcal{R}}_S(t)} U\nabla\Phi\cdot\nabla W_i\, d\mathbf{x} \\ &= \kappa_S \oint_{\Gamma_M(t)} W_i \nabla U_S\cdot\mathbf{n}_S\, ds + \kappa_S \oint_{\Gamma_{F_S}} W_i \nabla U\cdot\mathbf{n}\, ds - \kappa_S \int_{\overline{\mathcal{R}}_S(t)} \nabla W_i\cdot\nabla U\, d\mathbf{x} \end{aligned} \tag{4.5}$$

and in the liquid region,

$$\begin{aligned} &C_i^L\dot{\Theta}_L - U_M \oint_{\Gamma_M(t)} W_i \nabla\Phi\cdot\mathbf{n}_L\, ds + \int_{\overline{\mathcal{R}}_L(t)} U\nabla\Phi\cdot\nabla W_i\, d\mathbf{x} \\ &= \kappa_L \oint_{\Gamma_M(t)} W_i \nabla U_L\cdot\mathbf{n}_L\, ds + \kappa_L \oint_{\Gamma_{F_L}} W_i \nabla U\cdot\mathbf{n}\, ds - \kappa_L \int_{\overline{\mathcal{R}}_L(t)} \nabla W_i\cdot\nabla U\, d\mathbf{x}. \end{aligned} \tag{4.6}$$

In these equations $\mathbf{n}$ always represents the outward pointing normal to the relevant domain and, as illustrated in Fig. 4,

$$\mathbf{n}_L = -\mathbf{n}_S.$$

This implies that Φ is dual-valued on the moving interface $\Gamma_M(t)$, but this is not a problem because the recovery of $\dot{\mathbf{X}}$ from Φ in (2.12) in each phase requires only integrals over mesh cells.

The Stefan condition (4.2) has not yet been used. We shall write it in a distributed form, consistent with the velocity potential equations (4.5) and (4.6), given by

$$k_S \oint_{\Gamma_M(t)} W_i \nabla U_S\cdot\mathbf{n}\, ds - k_L \oint_{\Gamma_M(t)} W_i \nabla U_L\cdot\mathbf{n}\, ds = \lambda \oint_{\Gamma_M(t)} W_i \nabla\Phi\cdot\mathbf{n}\, ds, \tag{4.7}$$

for each node i on the interface Γ_M. Recall that ∇U is discontinuous across the interface, the subscripts S and L being used to denote the limiting values on either side in the solid and liquid phases respectively.

Provided that $U_M \neq 0$ and the dependent variable U does not change sign at any point in the whole domain (which may always be achieved by the addition of a suitable constant to the whole system), it is possible to apply the boundary condition (4.7) by replacing the terms involving $\nabla\Phi\cdot\mathbf{n}$ in the integrals along the moving interface on the left-hand sides of (4.5) and (4.6), giving

$$\begin{aligned}
&C_i^S\dot{\Theta}_S+\int_{\overline{\mathcal{R}}_S(t)} U\nabla\Phi\cdot\nabla W_i\,d\mathbf{x}\\
=&\kappa_S\oint_{\Gamma_M(t)} W_i\nabla U_S\cdot\mathbf{n}_S\,ds+\kappa_S\oint_{\Gamma_{F_S}} W_i\nabla U\cdot\mathbf{n}\,ds-\kappa_S\int_{\overline{\mathcal{R}}_S(t)}\nabla W_i\cdot\nabla U\,d\mathbf{x}\\
&+\frac{U_M}{\lambda}\oint_{\Gamma_M(t)} W_i(k_S\nabla U_S-k_L\nabla U_L)\cdot\mathbf{n}_S\,ds
\end{aligned}\tag{4.8}$$

for the solid region and

$$\begin{aligned}
&C_i^L\dot{\Theta}_L+\int_{\overline{\mathcal{R}}_L(t)} U\nabla\Phi\cdot\nabla W_i\,d\mathbf{x}\\
=&\kappa_L\oint_{\Gamma_M(t)} W_i\nabla U_L\cdot\mathbf{n}_L\,ds+\kappa_L\oint_{\Gamma_{F_L}} W_i\nabla U\cdot\mathbf{n}\,ds-\kappa_L\int_{\overline{\mathcal{R}}_L(t)}\nabla W_i\cdot\nabla U\,d\mathbf{x}\\
&+\frac{U_M}{\lambda}\oint_{\Gamma_M(t)} W_i(k_S\nabla U_S-k_L\nabla U_L)\cdot\mathbf{n}_L\,ds
\end{aligned}\tag{4.9}$$

for the liquid region.

Eqs. (4.8) and (4.9) can each be summed to obtain two additional equations, giving the rate of change of the total mass in each phase:

$$\begin{aligned}
\dot{\Theta}_S=&\kappa_S\oint_{\Gamma_M(t)}\nabla U_S\cdot\mathbf{n}_S\,ds+\kappa_S\oint_{\Gamma_{F_S}}\nabla U\cdot\mathbf{n}\,ds\\
&+\frac{U_M}{\lambda}\oint_{\Gamma_M(t)}(k_S\nabla U_S-k_L\nabla U_L)\cdot\mathbf{n}_S\,ds,
\end{aligned}\tag{4.10}$$

$$\begin{aligned}
\dot{\Theta}_L=&\kappa_L\oint_{\Gamma_M(t)}\nabla U_L\cdot\mathbf{n}_L\,ds+\kappa_L\oint_{\Gamma_{F_L}}\nabla U\cdot\mathbf{n}\,ds\\
&+\frac{U_M}{\lambda}\oint_{\Gamma_M(t)}(k_S\nabla U_S-k_L\nabla U_L)\cdot\mathbf{n}_L\,ds.
\end{aligned}\tag{4.11}$$

Following the procedure used for a single domain, the values of Φ_i and $\dot{\Theta}_S$ can be found by solving (4.8) and (4.10) with the assumption that $\Phi_i=0$ at an arbitrary node i in the solid region. Similarly, (4.9) and (4.11) can be solved for Φ_i and $\dot{\Theta}_L$ with the assumption that $\Phi_i=0$ at an arbitrary node i in the liquid phase.

4.2.2 Calculating the mesh velocity

The simplest way to recover the mesh velocities is to solve (2.12) for $\mathbf{V}$, exactly as for a single domain, except with $\mathbf{V}=\mathbf{0}$ imposed on Γ_F. While this procedure produces good approximations, it is significantly more accurate to impose the Stefan condition (4.2) explicitly in the recovery of $\mathbf{V}$ in (2.12). A single system is solved over the whole domain and, as before, an equation of the form (2.12) is constructed for each interior node, and $\mathbf{V}_i=\mathbf{0}$ imposed at all nodes on the fixed boundary, but for those nodes on the moving interface Eqs. (2.12) are replaced using (4.7), which gives the normal components of the velocity directly. It is assumed also that $U=U_M$ is constant along the moving interface, so ∇U is parallel to $\mathbf{n}$, and that the boundary nodes can only move in a direction perpendicular to the boundary. Under these circumstances, the resulting equations solved at the nodes on the moving interface take the form

$$\lambda \oint_{\Gamma_M(t)} W_i \mathbf{V}\, ds = k_S \oint_{\Gamma_M(t)} W_i \nabla U_S\, ds - k_L \oint_{\Gamma_M(t)} W_i \nabla U_L\, ds. \tag{4.12}$$

4.2.3 Recovering the dependent variable

The dependent variable U is recovered separately in each of the regions of the domain, $\overline{\mathcal{R}}_S$ and $\overline{\mathcal{R}}_L$, since they are decoupled by the Dirichlet boundary condition imposed on the interface. A robust algorithm is obtained by using the ALE equation given by (2.16) to recover the nodal values of the dependent variable. If Ψ_i^S and Ψ_i^L are the local masses in the solid and liquid phases (cf. Eq. (2.15)) we obtain

$$\begin{aligned}
\frac{d\Psi_i^S}{dt} &= \oint_{\Gamma_M(t)} \widetilde{W}_i (\kappa_S \nabla U_S + U_M \mathbf{V})\cdot \mathbf{n}_S\, ds + \kappa_S \oint_{\Gamma_{F_S}} \widetilde{W}_i \nabla U \cdot \mathbf{n}\, ds \\
&\quad - \int_{\overline{\mathcal{R}}_S(t)} \nabla \widetilde{W}_i \cdot (\kappa_S \nabla U + U\mathbf{V})\, d\mathbf{x}, \\
\frac{d\Psi_i^L}{dt} &= \oint_{\Gamma_M(t)} \widetilde{W}_i (\kappa_L \nabla U_L + U_M \mathbf{V})\cdot \mathbf{n}_L\, ds + \kappa_L \oint_{\Gamma_{F_L}} \widetilde{W}_i \nabla U \cdot \mathbf{n}\, ds \\
&\quad - \int_{\overline{\mathcal{R}}_L(t)} \nabla \widetilde{W}_i \cdot (\kappa_L \nabla U + U\mathbf{V})\, d\mathbf{x},
\end{aligned} \tag{4.13}$$

where $\widetilde{W}_i$ are the modified test functions of Section 2 which allow U to be specified strongly on Dirichlet boundaries. The values of $\Psi_i^S(t)$ and $\Psi_i^L(t)$ are then updated to the new time level by the time-stepping scheme, after which they can be substituted into

$$\int_{\overline{\mathcal{R}}_S(t)} \widetilde{W}_i U\, d\mathbf{x} = \Psi_i^S \qquad \text{and} \qquad \int_{\overline{\mathcal{R}}_L(t)} \widetilde{W}_i U\, d\mathbf{x} = \Psi_i^L \tag{4.14}$$

(cf. Eq. (2.15)) to find the new values for the dependent variable.

4.3 The one-phase problem

The single-phase Stefan problem is simply a special case of the two-phase problem described above, in which there is a single domain and it is assumed that $u=u_M$ everywhere in the region adjacent to the moving interface. This simplifies the equations considerably since only the liquid or the solid phase exists. For example, if the solid phase is being modeled then the equations to be solved for the mesh velocity potentials are given by (4.8) with $\nabla U_L=0$.

Regardless of whether a solid or a liquid phase is being modeled, in the single-phase case the interface condition (4.2) reduces to

$$k\nabla u\cdot\mathbf{n}=\lambda\mathbf{v}\cdot\mathbf{n}\qquad \text{on } \Gamma_M(t). \tag{4.15}$$

(Note however that the sign of k/λ will depend on the phase being modeled.) This leads to the following equations:

$$k\oint_{\Gamma_M(t)} W_i\nabla U\cdot\mathbf{n}\,ds=\lambda\oint_{\Gamma_M(t)} W_i\nabla\Phi\cdot\mathbf{n}\,ds \tag{4.16}$$

in place of (4.7),

$$\begin{aligned}&C_i\dot{\Theta}+\int_{\overline{\mathcal{R}}(t)} U\nabla\Phi\cdot\nabla W_i\,d\mathbf{x}\\ =\kappa&\oint_{\partial\overline{\mathcal{R}}(t)} W_i\nabla U\cdot\mathbf{n}\,ds-\kappa\int_{\overline{\mathcal{R}}(t)}\nabla W_i\cdot\nabla U\,d\mathbf{x}+\frac{kU_M}{\lambda}\oint_{\Gamma_M(t)} W_i\nabla U\cdot\mathbf{n}\,ds\end{aligned} \tag{4.17}$$

in place of (4.8)/(4.9),

$$\dot{\Theta}=\kappa\oint_{\partial\overline{\mathcal{R}}(t)}\nabla U\cdot\mathbf{n}\,ds+\frac{kU_M}{\lambda}\oint_{\Gamma_M(t)}\nabla U\cdot\mathbf{n}\,ds \tag{4.18}$$

in place of (4.10)/(4.11), and

$$\begin{aligned}\frac{d\Psi_i}{dt}=&\oint_{\Gamma_M(t)}\widetilde{W}_i(\kappa\nabla U+U_M\mathbf{V})\cdot\mathbf{n}\,ds+\kappa\oint_{\Gamma_F}\widetilde{W}_i\nabla U\cdot\mathbf{n}\,ds\\ &-\int_{\overline{\mathcal{R}}(t)}\nabla\widetilde{W}_i\cdot(\kappa\nabla U+U\mathbf{V})\,d\mathbf{x}\end{aligned} \tag{4.19}$$

in place of (4.13).

5 Numerical experiments

Throughout this section, unless stated otherwise, a constant value of $\delta=2$ is added to the initial and boundary conditions to allow the Stefan boundary condition to influence the evolution and ensure that the solution is positive throughout the whole domain.

5.1 Single-phase results

5.1.1 One-dimension

The first case considered is the one-dimensional single-phase example discussed in detail in [1]. As in all of the single-phase results presented here, it is assumed that

$$K=k=|\lambda|=1.$$

The single-phase problem permits each of the following analytic solutions on the domain indicated which satisfy the Stefan condition at $x=Vt$ (the point at which $u_i=u_M$) when $\lambda=-1$:

$$\begin{aligned} u^{(1)}(x,t)&=-1+e^{-V(x-Vt)} &&\text{for } x\le Vt,\\ u^{(2)}(x,t)&=-1+e^{-V(x-Vt)} &&\text{for } x\ge Vt. \end{aligned} \tag{5.1}$$

Clearly V is the (constant) interface velocity and, in this case, $U_M=0$. Two situations are examined here, both of which use a value of $V=-1$.

• Given the initial domain of $[-1,0]$ and initial conditions given by $u^{(1)}$ at $t_0=0$, a contracting domain is modeled with $u\le 0$ in the whole region. The choice $k/\lambda<0$ implies that this is modeling the liquid phase, so it actually represents a supercooled liquid, which is freezing at the moving interface, since $V<0$.

• Given the initial domain of $[0,1]$ and initial conditions given by $u^{(2)}$ at $t_0=0$, an expanding domain is modeled with $u\ge 0$ in the whole region. Again $k/\lambda<0$ implies that this is modeling the liquid phase, which is melting the adjacent solid at the moving interface, since $V<0$.

In each case $u_x=-Ve^{-V(x-Vt)}$ is imposed as a Neumann condition on the fixed boundary in the calculation of the mesh velocity potentials. Note that there are analogous exact solutions for the case where $k/\lambda>0$.

The results shown in Fig. 5 and Table 1 clearly show that the overall method is second order accurate for the one-dimensional problem. Note that, for this one-dimensional problem, the "mesh error" (i.e., the error in the position of the moving interface at the final time) is the same in each of the norms considered since the interface is just a single point.

Table 1: Numerical estimates of the orders of accuracy of the scheme applied to the single-phase Stefan problem.

	Solution errors			Mesh errors		
	L^1	L^2	L^∞	L^1	L^2	L^∞
1D Contracting	2.00	2.00	2.00		2.01	
1D Expanding	1.99	1.99	1.99		1.99	
2D	1.80	1.83	1.81	1.86	1.84	1.29

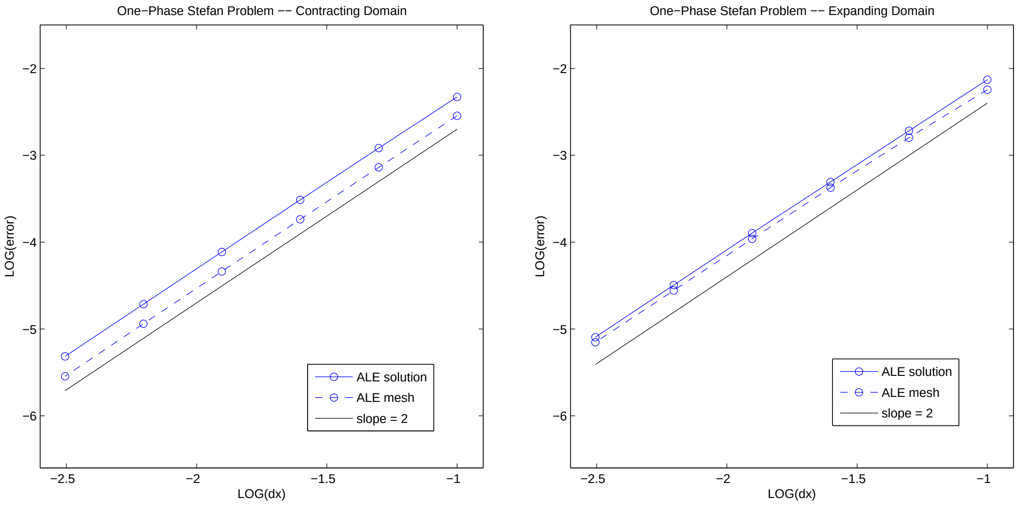

Figure 5: Accuracy of the approximate solutions on a sequence of meshes at $T=0.5$ with a contracting domain (left) and with an expanding domain (right). The solution and mesh errors shown are both L^2 approximations.

5.1.2 Two dimensions

The radially-symmetric two-dimensional test case used here is known as the Frank spheres problem [27] and has been studied in some detail in [1,22]. The exact solution is given by

$$u(r,t)=\begin{cases} u_\infty\left(1-\frac{E_1(s^2/4)}{E_1(S^2/4)}\right), & s\ge S,\\ 0, & s<S,\end{cases} \tag{5.2}$$

in which $s=rt^{-\frac{1}{2}}$ and $E_1(z)$ is the exponential integral

$$E_1(z)=\int_z^\infty \frac{e^{-\xi}}{\xi}\mathrm{d}\xi. \tag{5.3}$$

The problem is parameterized by the quantity S, where $R(t)=St^{\frac{1}{2}}$ is the radius of the expanding interface. Its value is calculated from u_∞ by solving

$$u_\infty=\frac{E_1(S^2/4)}{E_1{}'(S^2/4)}. \tag{5.4}$$

Here $u_\infty=-0.5$, giving $S\approx 1.5621239$, and the initial conditions were set at time $t_0=1$.

The initial domain for this test case was taken to be an annulus made up of a moving inner circle (initially at $r=S$) and a fixed outer circle at $r=2S$ (the boundary condition imposed on this fixed outer boundary is simply a Dirichlet condition given by (5.2)). Unlike the one-dimensional case, the value of $\nabla U\cdot\mathbf{n}$ was not specified on the fixed boundary during the calculation of the mesh node velocity potentials. Instead a one-sided approximation was used and the value of U was fixed at the boundary when the dependent variable was recovered. As in the one-dimensional test cases, $\lambda=-1$ so, since $U\le 0$ and the

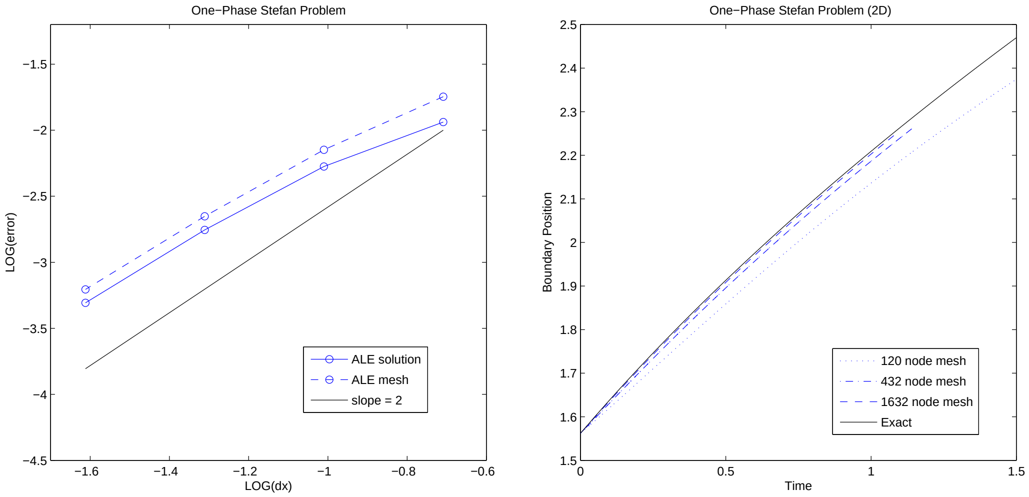

Figure 6: Accuracy of the approximate solutions on a sequence of meshes at $T=0.5$ for the two-dimensional one-phase Stefan problem (left) and the evolution of the average distance of the inner boundary nodes from the centre of the domain (right). The solution and mesh errors shown are both L^2 approximations.

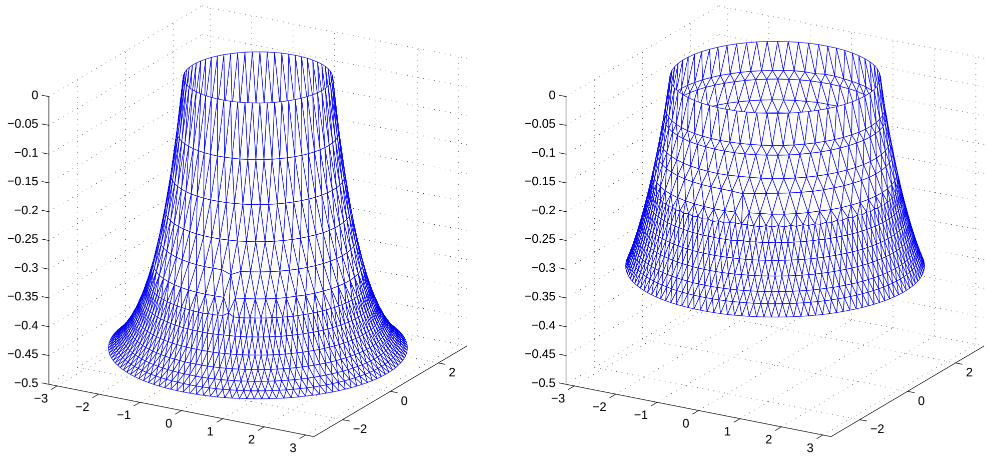

Figure 7: Snapshots of the solution with the mesh projected on to it for the two-dimensional one-phase Stefan problem at $T=0$ (left) and $T=1$ (right).

inner boundary is moving outwards, which represents the freezing of an under-cooled liquid.

The results shown in Fig. 6 and Table 1 suggest that the method gives an approximation which is second order accurate. The evolution of the boundary is shown on the right-hand side of Fig. 6, confirming the convergence of the approximation to the exact solution as the mesh is refined. Time $T=3$ is the point at which the moving inner boundary should coincide with the fixed outer boundary. A sample solution profile is shown in Fig. 7 which is at a much later experiment time than was possible with the original version of the method presented in [6].

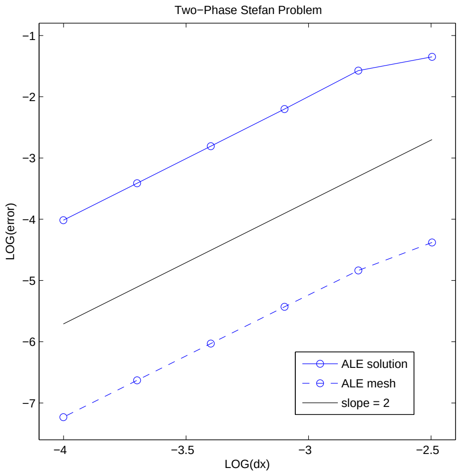

Figure 8: Comparison of L^2 errors in the solution and the magnitudes of the errors in the boundary node positions for the two-phase Stefan problem in one space dimension.

5.2 Two-phase results

5.2.1 One-dimension

An exact solution to the one-dimensional two-phase Stefan problem is given in [21] as

$$\begin{aligned} u_S &= u^*\left(1-\frac{erf\left(x/(2\sqrt{\kappa_S t})\right)}{erf\,\phi}\right), \\ u_L &= u_0\left(1-\frac{erfc\left(x/(2\sqrt{\kappa_L t})\right)}{erfc\left(\phi\sqrt{\kappa_S/\kappa_L}\right)}\right), \end{aligned} \tag{5.5}$$

where ϕ is the root of the transcendental equation

$$\frac{e^{-\phi^2}}{erf\,\phi}+\frac{k_L}{k_S}\sqrt{\frac{\kappa_S}{\kappa_L}}\frac{u_0 e^{-\kappa_S\phi^2/\kappa_L}}{u^*\,erfc\left(\phi\sqrt{\kappa_S/\kappa_L}\right)}+\frac{\phi\lambda\sqrt{\pi}}{K_S u^*}=0, \tag{5.6}$$

and erf and $erfc$ are the standard error and complementary error functions, respectively. The position of the moving interface is given by

$$s(t)=2\phi\sqrt{\kappa_S t}. \tag{5.7}$$

In order to compare with previous solutions [28, 33], the parameters are chosen to be

$$k_S=2.22, \quad k_L=0.556, \quad K_S=1.762, \quad K_L=4.226, \quad \lambda=338, \tag{5.8}$$

with $u^*=-20$ and $u_0=10$, which gives $\phi\approx 0.20542692937650$. In order to avoid the singularity at $t=0$, the initial time of the experiment is taken at $t_0=0.0012$. For this case a value of $\delta=25$ was added to the initial and boundary conditions to ensure that the solution remained positive throughout the domain.

Table 2: Numerical estimates of the orders of accuracy of the scheme applied to the two-phase Stefan problem.

	Solution errors			Mesh errors		
	L^1	L^2	L^∞	L^1	L^2	L^∞
1D	2.00	2.00	2.01	2.00		

The results shown in Fig. 8 and Table 2 (sampled at $T=0.001$) clearly show the method to be second order accurate. Note that, in order to represent the interface accurately, the initial meshes have been adjusted so that half of the total number of mesh cells are in the left-hand region where they are of uniform width. The widths of the remaining cells, in the right-hand region, are calculated using a geometric progression, setting the width of the cell adjacent to the moving interface to be equal to that of its neighbour, on the opposite side of the interface, and fixing the final node at $x=1$.

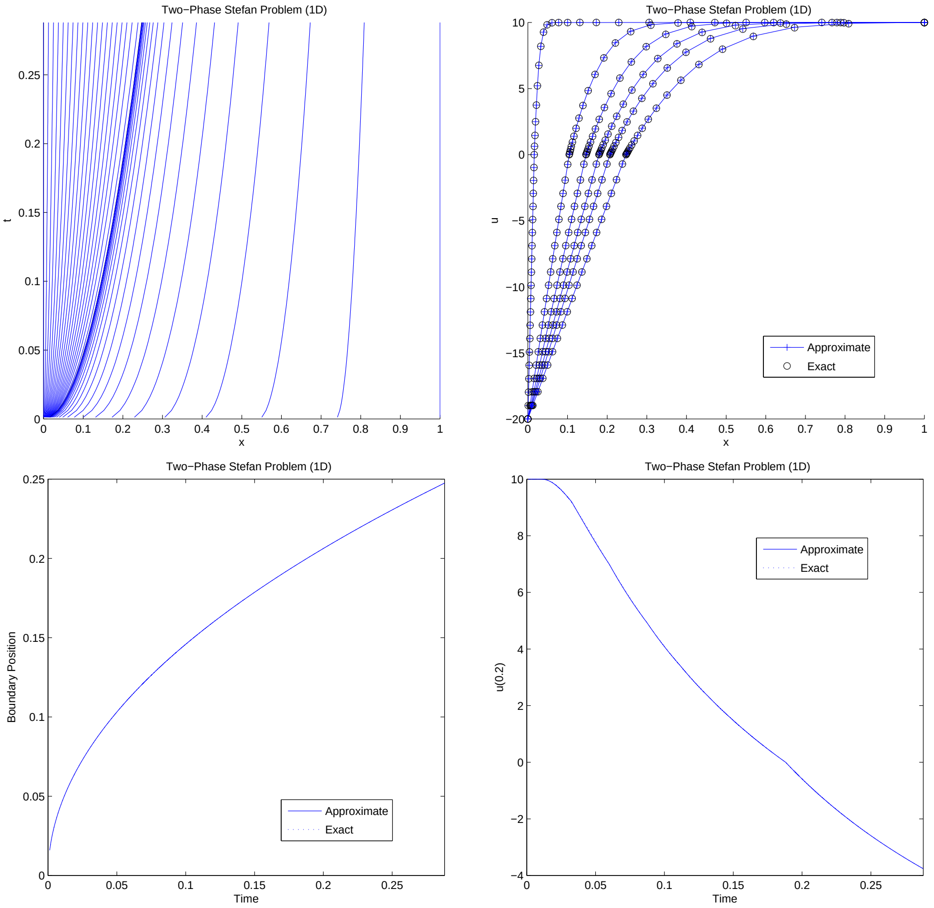

Figure 9: Comparison of exact and approximate solutions to the one-dimensional two-phase Stefan problem for a 41 node mesh: node trajectories (top left), snapshots of the solution (top right), interface position (bottom left) and solution value at $x=0.2$ (bottom right).

Fig. 9 shows a series of results at $T=0.288$, which can be compared with those presented elsewhere in the literature [28, 33]. The main points to note are that the results are considerably more accurate than those presented in [33] but that, due to the explicit time-stepping scheme that is used here, they require significantly smaller (and therefore many more) time-steps to be taken. Fig. 10 shows a similar series of results for the same experiment, run until the mesh tangles. The exact and approximate graphs of the boundary position and the solution at $x=0.2$ remain indistinguishable for the whole time (until the final few time-steps prior to mesh tangling, which are not shown).

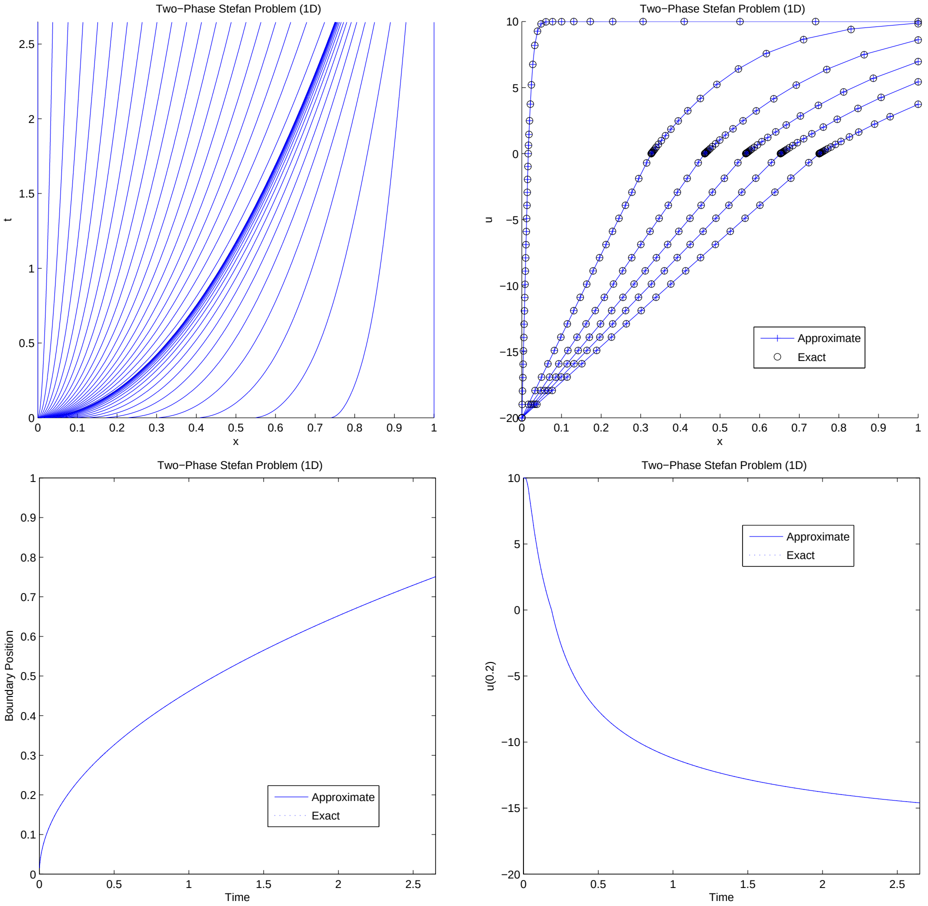

Figure 10: Comparison of exact and approximate solutions to the one-dimensional two-phase Stefan problem for a 41 node mesh: node trajectories (top left), snapshots of the solution (top right), interface position (bottom left) and solution value at $x=0.2$ (bottom right).

Before moving on to the two-dimensional, two-phase case, for which we have no analytic results against which quantitative comparisons may be made, we consider a final one-dimensional example based upon a radially-symmetric problem. This may then be used as the basis for a qualitative assessment of the two-dimensional scheme in the

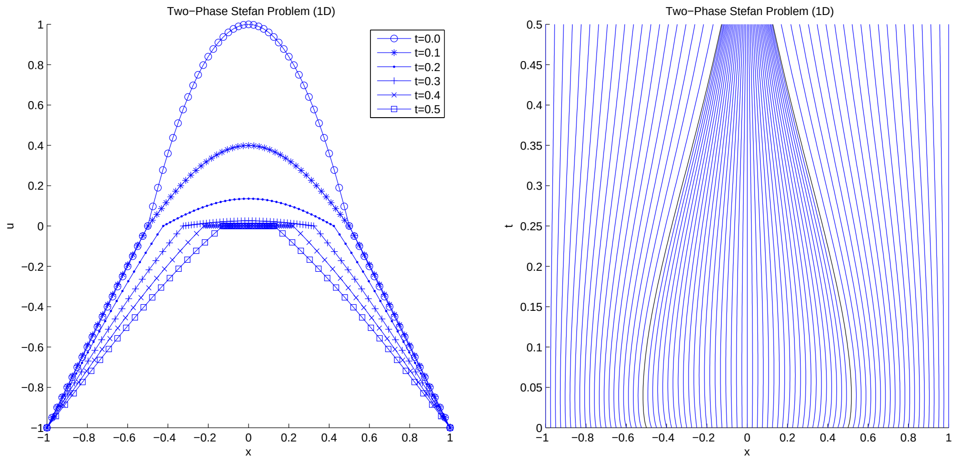

Figure 11: Evolution of the solutions to the one-dimensional two-phase Stefan problem for a symmetric initial profile and an 81 node mesh: snapshots of the solution profile (left) and node trajectories (right).

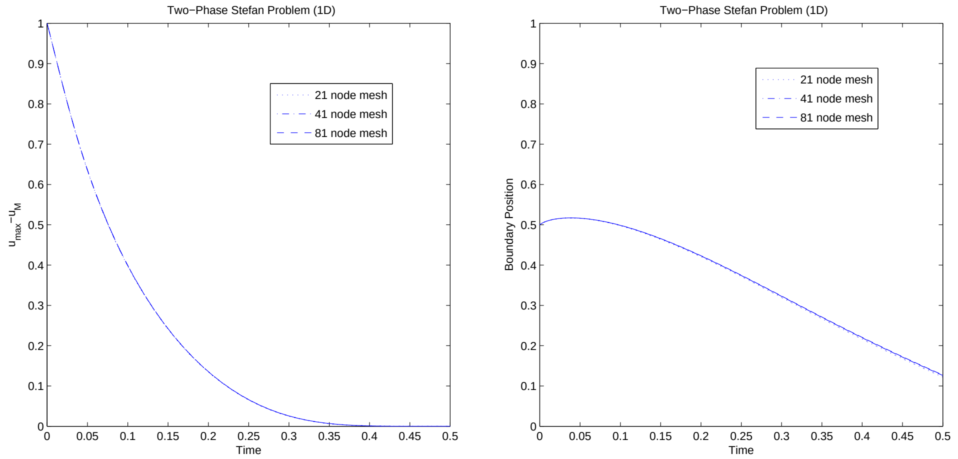

Figure 12: Evolution of the value of U at the centre of the domain (left) and the position of the moving interface (right) for the one-dimensional two-phase Stefan problem with symmetric initial profile.

following section. Problem (4.1) and (4.2) is solved with initial conditions given by

$$u(r,0)=\begin{cases} 1-4r^2, & r\leq\frac{1}{2}, \\ 1-2r, & r>\frac{1}{2} \end{cases} \tag{5.9}$$

in which $r\in[0,1]$ is the usual radial coordinate. This represents liquid in the inner region, surrounded by solid which is held at a fixed temperature (below freezing point, $u_M=0$) at the outer boundary: $K_L=K_S=k_L=k_S=\lambda=1$ is assumed. The outer boundaries are treated as Dirichlet boundaries with $u=-1$ fixed throughout the experiment.

The results obtained for the one-dimensional problem with initial conditions given by (5.9) are illustrated by Figs. 11 and 12. Varying the mesh resolution produces results which are almost indistinguishable: in all cases the peak value in the interior decreases steadily, while the interface initially moves outwards until the point where the gradient of the dependent variable is the same on both sides, after which it starts to move inwards. Note that, as the solution flattens in the inner region, the mesh cells get smaller and eventually (shortly after the final snapshot shown in Fig. 11) the mesh tangles and cells with negative lengths appear.

A non-symmetric solution is illustrated in Fig. 13. The initial conditions were obtained by shifting the liquid region by 0.25 to the left and assuming linear variation of u in the remaining solid regions, with the same outer boundary value as before. These results exhibit features that are very similar to those of the symmetric test case.

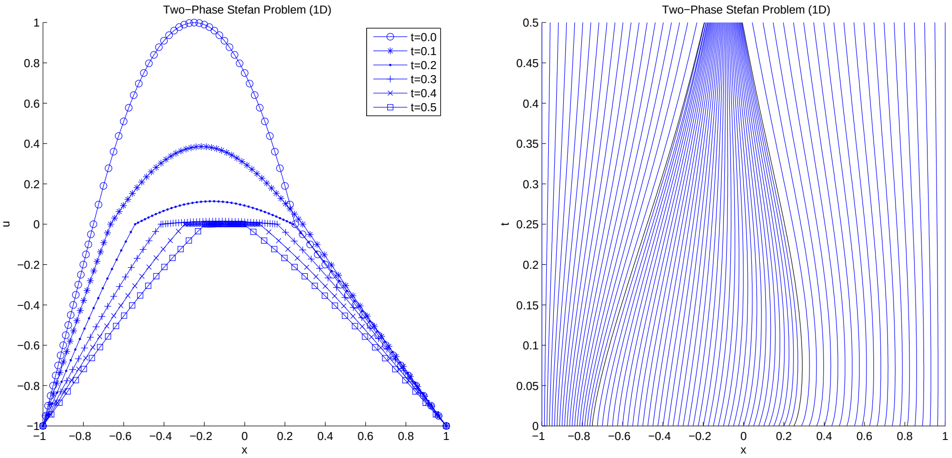

Figure 13: Evolution of the solutions to the one-dimensional two-phase Stefan problem for a non-symmetric initial profile and an 81 node mesh: snapshots of the solution profile (left) and node trajectories (right).

5.2.2 Two dimensions

The first two-dimensional test case presented here also uses (5.9) to define the initial conditions, giving a radially-symmetric situation. Snapshots of the solution at four different times are given in Fig. 14 for an unstructured mesh of 1589 nodes and 3050 cells. The evolution of the solution is very similar to the one-dimensional case already considered, in that the interface moves outwards until the point where the component of the gradient of the dependent variable normal to the interface is the same on both sides, after which it starts to move inwards (see the right-hand side of Fig. 15). Fig. 15 also illustrates the convergence of the scheme on a succession of meshes for the evolution of the problem up to $t=0.2$. Mesh tangling starts to occur beyond $t=0.21$, when the values of U in the interior region have all dropped well below 10^{-4}.

Finally, the method is used to illustrate the behaviour of the moving-mesh finite element method for a non-radially-symmetric problem. The initial conditions used are simple perturbations of the radially-symmetric case and are shown in Fig. 16 along with the solution profiles and meshes shortly before the mesh tangles.

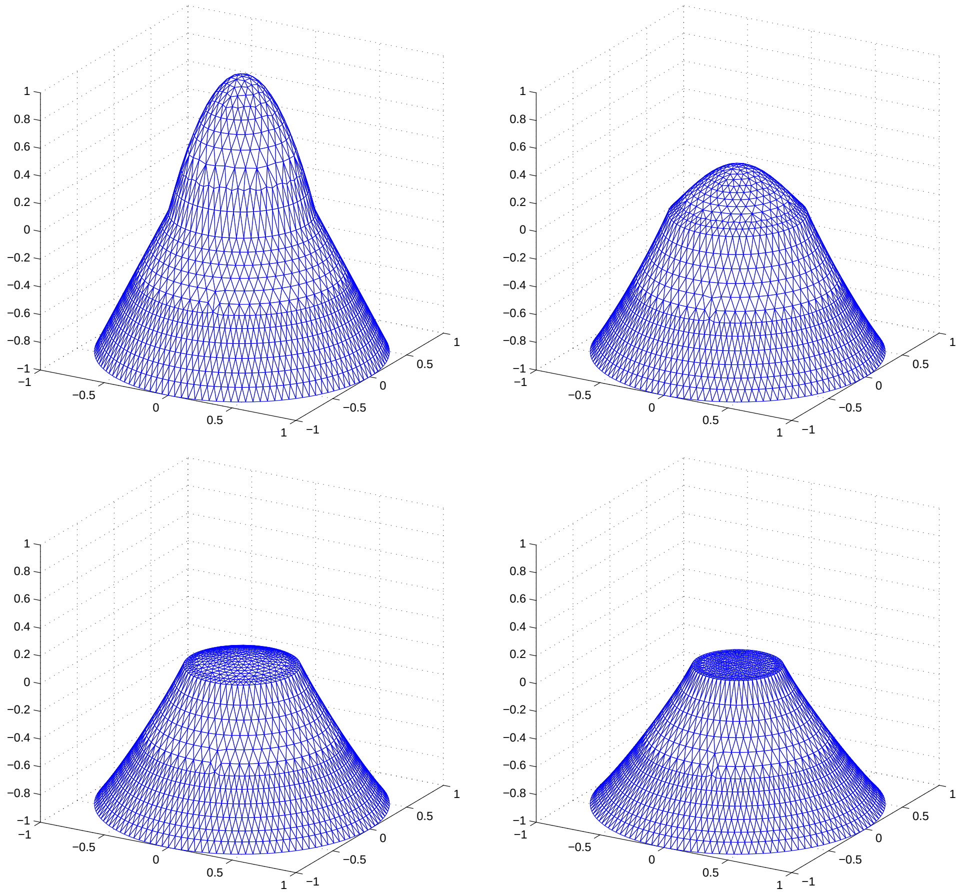

Figure 14: Solution profiles for the two-phase Stefan results approximating a two-dimensional radially-symmetric problem at times $t = 0.0, 0.05, 0.1, 0.15$.

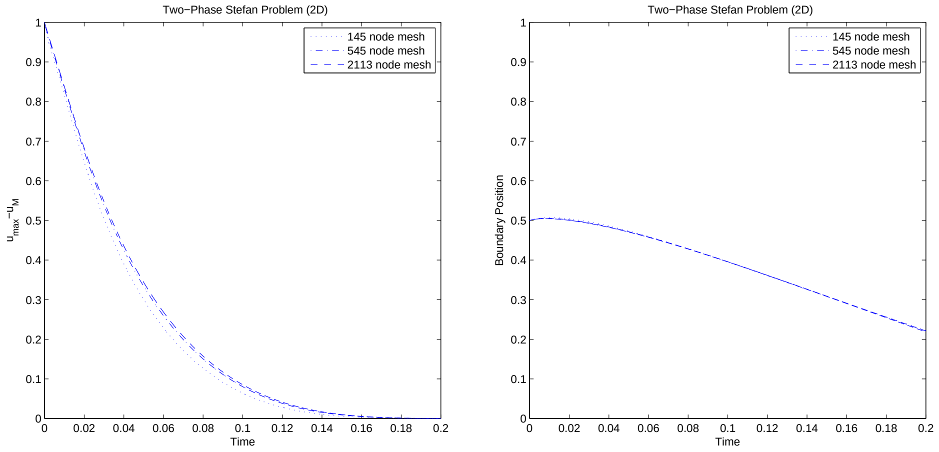

Figure 15: Evolution of the value of U at the centre of the domain (left) and the average distance of the moving interface from the centre of the domain (right) for the two-dimensional two-phase Stefan problem.

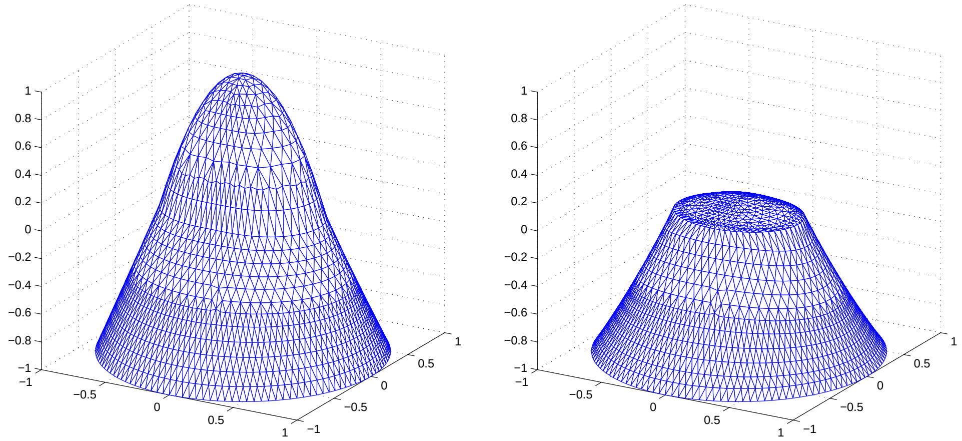

Figure 16: Solution profiles for the two-phase Stefan results approximating a two-dimensional non-radially-symmetric problem at times $t=0.0,0.09$.

6 Discussion

In this paper we have described a generalized form of the multidimensional moving-mesh finite element algorithm introduced in [6] and demonstrated its effectiveness in simulating the solutions of moving boundary problems. The improved robustness and flexibility of the method has been achieved through a series of modifications, which have then been validated on a number of test problems with moving boundaries. These modifications include the ability to impose Dirichlet boundary conditions strongly, the use of an ALE approach to update the solution values (rather than recovering them directly from (2.4)) and a generalization of the way in which the velocity potential (Φ) and rate-of-change of mass ($\dot{\Theta}$) are calculated. However, the method still requires no mesh smoothing.

The generalized method has been described in detail in Sections 2.1 and 2.2, with the algorithm given and compared with its predecessor in Section 2.3. Its accuracy and robustness have been illustrated through comparison with analytic solutions of a number of moving boundary problems, in particular a two-phase problem with an internal moving boundary, treated for the first time by this method. The other moving boundary problems studied here include a mass-conserving nonlinear diffusion problem and a non-mass-conserving diffusion problem with a sink term in Section 3, and a single-phase Stefan problem included in Section 4, both of which have a moving boundary as an implicit part of the solution (both expanding and contracting domains are tested). In all these problems the generalized method successfully reproduces analytic solutions to second order accuracy.

However, the main advance presented in this paper has been the application of the algorithm to multiphase Stefan problems in Section 4, involving a weak implementation of the Stefan condition which models the change of phase across a moving interface in the interior of the domain. Comparison with analytic solutions to one-phase and two-phase Stefan problems have demonstrated that the scheme is second order accurate in its approximation of both the solution and the interface position in one and two space dimensions. Further results show that the moving-mesh finite element algorithm can be reliably applied to phase-change problems approximated on genuinely unstructured multidimensional meshes.

The benefits of the strong imposition of the Dirichlet boundary conditions are illustrated in detail in [29]. However, the solution of (2.10) and (2.11) for the velocity potentials Φ_i and for $\dot{\Theta}$, imposing $\Phi=0$ at one point, and the reliance on the ALE equation do perhaps require further comment. Although the ALE approach appears to yield about the same accuracy as recovering the solution values directly from (2.4) and both approaches allow conservation of mass to be guaranteed, for some of the test problems considered, such as the two-dimensional two-phase examples, the ALE approach shows significant benefits: without it the solutions obtained can show small non-physical oscillations and the mesh tangles significantly earlier in the simulation. Other approaches to recovering the mesh velocity potentials have been investigated. In particular, it is possible to construct methods which allow $U_M=0$ on the moving interface in the Stefan problem and which solve for Φ over the whole computational domain, instead of splitting it into its component phases. However, the approach presented here has proved to be the most robust.

As with all moving-mesh methods, there are a number of limitations to the scheme presented here that it is important to discuss. As described, in this work the number of degrees of freedom and the connectivity of the finite element mesh remain fixed throughout each simulation. Clearly this is a restriction that can, and does, lead to mesh tangling when the solution evolves very significantly. Consequently, it makes sense to combine this approach with some form of discrete remeshing that would permit the topology of the mesh to be adapted on occasions. It would of course be important in this approach to address the issue of mass conservation (both local and global) when interpolating between meshes following such an adaptive step.

Another significant constraint on the method, as implemented here, is the explicit time integration scheme that is currently used. This leads to a non-trivial stability restriction on the maximum time-step that may be taken, and thus reduces the overall efficiency of the implementation. This may be overcome by the use of an implicit time-stepping strategy, based upon a multi-step scheme such as BDF2 for example [5], possibly combined with adaptive time-step selection as in [37]. Other generalizations (that are the subject of current work) include the use of different monitor functions to control the mesh evolution and the use of different ALE schemes to update the solution values on the moving grid (see, for example, [8]).

The technique generalizes naturally to three-dimensional problems using tetrahedral meshes and to problems exhibiting features such as blow-up in finite time (see, *e.g.*, [16]). Its capability of simulating multidimensional moving boundary problems and problems with evolving discontinuities suggests that it can be a powerful tool in predicting multidimensional behaviour of nonlinear problems and providing conjectures for further analysis.

Acknowledgments

This work was undertaken with the support of EPSRC Grant EP/D058791/1. R. Mahmood wishes to thank his employer PINSTECH for granting him study leave to carry out research work at Leeds.

References

[1] R. Almgren, Variational algorithms and pattern formation in dendritic solidification, J. Comput. Phys., 106:337–354, 1993.

[2] B. N. Azarenok and T. Tang, Second-order Godunov-type scheme for reactive flow calculations on moving meshes. J. Comput. Phys., 206:48–80, 2005.

[3] D. G. Aronson, The Porous Medium Equation, in Lecture Notes in Mathematics, 1224:1–46, 1986.

[4] M. J. Baines, Moving finite elements, Oxford University Press, 1994.

[5] K. W. Blake and M. J. Baines, Moving-mesh methods for nonlinear partial differential equations, Numerical Analysis Report 7/01, Department of Mathematics, University of Reading, UK, 2001.

[6] M. J. Baines, M. E. Hubbard and P. K. Jimack, A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries, Appl. Numer. Math., 54:450–469, 2005.

[7] M. J. Baines, M. E. Hubbard and P. K. Jimack, A moving mesh finite element algorithm for fluid flow problems with moving boundaries, Int. J. Numer. Methods Fluids, 47(10/11):1077–1083, 2005.

[8] M. J. Baines, M. E. Hubbard, P. K. Jimack and A. C. Jones, Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions, Appl. Numer. Math., 56:230–252, 2006.

[9] G. I. Barenblatt, Scaling, Cambridge University Press, 2003.

[10] G. Beckett, J. A. Mackenzie and M. L. Robertson, A moving-mesh finite element method for the solution of two-dimensional Stefan problems, J. Comput. Phys., 168:500–518, 2001.

[11] A. E. Berger, M. Ciment and J. C. W. Rogers, Numerical solution of a diffusion consumption problem with a free boundary, SIAM J. Numer. Anal., 12:646–672, 1975.

[12] P. Bochev, G. Liao and G. dela Pena, Analysis and computation of adaptive moving grids by deformation, Numer. Meth. Part. D. E., 12:489–506, 1998.

[13] C. Bonacina, G. Comini, A. Fasano and M. Primicerio, Numerical solution of phase-change problems, Int. J. Heat Mass Tran., 16:1825–1832, 1973.

[14] C. J. Budd, G. J. Collins, W. Z. Huang and R. D. Russell, Self-similar numerical solutions of the porous-medium equation using moving mesh methods, Philos. T. Roy. Soc. A, 357:1047–1078, 1999.

[15] C. J. Budd and M. D. Piggott, Geometric integration and its applications, Foundations of Computational Mathematics, Handbook of Numerical Analysis XI, ed. P.G.Ciarlet and F.Cucker, Elsevier, pp. 35–139, 2003.

[16] C. J. Budd and J. F. Williams, Parabolic Monge-Ampère methods for blow-up problems in several spatial dimensions, J. Phys. A-Math. Gen., 39:5425–5444, 2006.

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

[18] W. M. Cao, W. Z. Huang and R. D. Russell, A moving-mesh method based on the geometric conservation law, SIAM J. Sci. Comput., 24:118–142, 2002.

[19] N. N. Carlson and K. Miller, Design and application of a gradient-weighted moving finite element code I: In one dimension, SIAM J. Sci. Comput., 19:728–765, 1998.

[20] N. N. Carlson and K. Miller, Design and application of a gradient-weighted moving finite element code II: In two dimensions, SIAM J. Sci. Comput., 19:766–798, 1998.

[21] H. S. Carslaw and J. C. Jaeger, Conduction of Heat in Solids, Oxford University Press, 1959.

[22] S. Chen, B. Merriman, S. Osher and P. Smereka, A simple level set method for solving Stefan problems, J. Comput. Phys., 135:8–29, 1997.

[23] J. Crank, Free and Moving Boundary Problems, Oxford University Press, 1984.

[24] J. Crank and R. S. Gupta, A moving boundary problem arising from the diffusion of oxygen in absorbing tissue, J. Inst. Math. Appl., 10:19–33, 1972.

[25] Y. Di, R. Li and T. Tang, A general moving mesh framework in 3D and its application for simulating the mixture of multi-phase flows, Commun. Comput. Phys., 3:582–602, 2008.

[26] M. J. Djomehri and J. H. George, Application of the moving finite element method to moving boundary Stefan problems, Comput. Method. Appl. M., 71:125–136, 1988.

[27] F. C. Frank, Radially symmetric phase growth controlled by diffusion, Proc. R. Soc. A, 201:586–599, 1950.

[28] R. M. Furzeland, A comparative study of numerical methods for moving boundary value problems, J. Inst. Math. Appl., 26: 411–429, 1980.

[29] M. E. Hubbard, M. J. Baines and P. K. Jimack, Mass-conserving Dirichlet boundary conditions for a moving mesh finite element algorithm, Appl. Numer. Math., to appear. doi:10.1016/j.apnum.2008.08.002, 2008.

[30] W. Z. Huang and R. D. Russell, Adaptive mesh movement – the MMPDE approach and its applications, J. Comput. Appl. Math., 128:383–398, 2001.

[31] W. Z. Huang and X. Y. Zhan, Adaptive moving mesh modeling for two dimensional groundwater flow and transport, In AMS Contemporary Mathematics, 383:283–296, 2005.

[32] R. Li, T. Tang and P. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys., 177:365–393, 2002.

[33] 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., 161:537–557, 2000.

[34] K. Miller and R. N. Miller, Moving finite elements I, SIAM J. Numer. Anal., 18:1019–1032, 1981.

[35] J. D. Murray, Mathematical Biology: An Introduction (3rd edition), Springer, 2002.

[36] J. Ockendon, The role of the Crank-Gupta model in the theory of free and moving boundary problems, Adv. Comp. Math., 6:281–293, 1996.

[37] J. Rosam, P. K. Jimack and A. M. Mullis, A fully implicit, fully adaptive time and space discretization method for phase-field simulation of binary alloy solidification, J. Comput. Phys., 225:1271–1287, 2007.

[38] T. Tang, Moving mesh methods for computational fluid dynamics, In AMS Contemporary Mathematics, 383:141–173, 2005.

[39] Z.-J. Tang, T. Tang and Z. R. Zhang, A simple moving mesh method for one- and two-

dimensional phase-field equations, J. Comput. Appl. Math., 190:252–269, 2006.

[40] P. D. Thomas and C. K. Lombard, The geometric conservation law and its application to flow computations on moving grids, AIAA J., 17:1030–1037, 1979.

[41] J. L. Vazquez, The Porous Medium Equation: Mathematical Theory, Oxford University Press, 2006.

[42] B. V. Wells, A moving mesh finite element method for the numerical solution of partial differential equations and systems, PhD thesis, Department of Mathematics, University of Reading, UK, 2005.

[43] B. V. Wells, M. J. Baines and P. Glaister, Generation of Arbitrary Lagrangian-Eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations, Int. J. Numer. Methods Fluids, 47(10/11):1375–1381, 2005.

[44] P. A. Zegeling, W. de Boer and H. Tang, Robust and efficient adaptive moving mesh solution of the 2-D Euler equations, In AMS Contemporary Mathematics, 383:419–430, 2005.