A moving-mesh finite element method and its application to the numerical solution of phase-change problems

Moving Mesh Finite Elements for Moving Internal and External Dirichlet Boundaries

M.J. Baines
M.E. Hubbard
P.K. Jimack
R. Mahmood

November 25, 2005

Abstract

The application of a recently developed moving mesh finite element algorithm to problems with moving internal and external boundaries is discussed. Particular attention is paid to the problem of combining Dirichlet boundary conditions on the dependent variable with mass conservation, and approaches which allow both to be satisfied simultaneously are proposed. Their merits relative to approaches which impose only one of these conditions exactly are also commented on. A modification to the original scheme is suggested which allows its use in the approximation of problems where mass is not conserved, even when its variation in time cannot be written out explicitly. The discussion is illustrated through one- and two-dimensional numerical approximations of (i) a mass-conserving problem, the porous medium equation, for which simple self-similar solutions exist with finite but time-dependent support, (ii) a problem for which mass isn't conserved, the one-phase Stefan problem, which also has simple exact solutions available for validation, and (iii) a problem with a moving internal Dirichlet boundary, a two-phase Stefan problem.

1 Introduction

REDO THIS SECTION!

The moving mesh finite element method of Baines, Hubbard and Jimack [1, 2, 3] has been successfully applied to a range of time-dependent approximations to PDEs where a moving boundary is involved. In those papers, the results shown are obtained using a simple implementation of the necessary conditions at the moving boundaries. In particular, when the nodal values of the dependent variable were recovered the Dirichlet boundary conditions were only imposed approximately, but in a manner which ensured that mass was conserved exactly in the interior of the domain. This work describes how it is possible to impose boundary conditions of this type exactly while still maintaining conservation of mass, and discusses the relative merits of the different approaches.

Section 2 provides a brief summary of the moving mesh finite element method, and is followed in Section 3 by a description of the available options for applying the boundary conditions. These are then compared in Section 3.2 using solutions of the one- and two-dimensional porous medium equation, to illustrate the differences, and the paper concludes with a summary of the observations.

2 The Moving Mesh Finite Element Method

The proposed algorithms may be applied in any number of space dimensions but, for the purposes of clarity the much of the description of the method will be illustrated in two space dimensions.

Consider initially an abstract time-dependent partial differential equation (PDE) of the general form

$$\frac{\partial u}{\partial t} = Lu\,, \tag{1}$$

on a finite, time-dependent domain $\Omega(t)$. L is a differential operator involving space derivatives only and the dependent variable $u = u(\mathbf{x}, t)$ is defined in a fixed frame of reference with coordinate $\mathbf{x}$ at time t.

In the context of PDE (1), the total mass of the system can be expressed as

$$\theta = \int_{\Omega(t)} u\, d\Omega\,, \tag{2}$$

and its variation in time may be written as

$$\frac{d}{dt}\int_{\Omega(t)} u\, d\Omega = \dot{\theta}\,. \tag{3}$$

The original moving mesh finite element method was derived for the situation where $\dot{\theta} = 0$, *i.e.* mass conservation. The algorithm was driven by *local* mass conservation, whereby a proportion of the total mass was assigned to each mesh node and the dependent variable was recovered after the mesh was moved in a manner which retained this distribution of the total mass across the nodes.

Mass is no longer assumed to be conserved, so the mesh movement must be driven in a slightly more general manner. It is still possible to "distribute" the mass at each time-step across a series of test functions $w_i = w_i(\mathbf{x}, t)$, leading to a slightly different set of local weak forms (which reduce to local weak conservation principles when $\dot{\theta} = 0$) given by

$$\frac{d}{dt}\int_{\Omega(t)} w_i\, u\, d\Omega = \dot{\theta}_i = c_i\,\dot{\theta}\,. \tag{4}$$

The c_i represent the proportion of the total mass associated with the i^{th} test function, and in this more general case it is this that will be kept constant as time progresses, to drive an appropriate mesh movement. Importantly, these equations sum to the global equation (3) when $\sum w_i \equiv 1$ over the whole domain. Careful choice of the w_i, *e.g.* to be standard finite element test functions, then ensures that the equations (4) are localised in some sense, which simplifies the discrete system which will result. Similarly, these test functions also lead to a corresponding distributed form for Equation (2), given by

$$\int_{\Omega(t)} w_i\, u\, d\Omega \;=\; c_i\,\theta\,, \tag{5}$$

where $c_i \in [0,1]$ represents the fraction of the total mass associated with the test function w_i. Clearly $\sum c_i = 1$ is required to maintain the correct total mass.

The moving mesh finite element method is derived by differentiating the integrals in (4) with respect to time, assuming that the test functions are advected with velocity $\dot{\mathbf{x}}$, applying Leibnitz' rule and integrating by parts. This gives [1]

$$c_i\,\dot\theta \;-\; \oint_{\partial\Omega(t)} w_i\, u\,\dot{\mathbf{x}}\cdot\mathbf{n}\; d\Gamma \;+\; \int_{\Omega(t)} u\,\dot{\mathbf{x}}\cdot\nabla w_i\; d\Omega \;=\; \int_{\Omega(t)} w_i\, Lu\; d\Omega\,, \tag{6}$$

where $\mathbf{n}$ is the unit outward normal to $\partial\Omega(t)$, the boundary of the domain. These equations could be discretised directly to return velocities but, instead, $\dot{\mathbf{x}}$ is replaced by the gradient of a velocity potential, $\nabla\phi$. This imposes an irrotationality condition on the velocity field (this is not necessary, the curl can be prescribed to be non-zero [1], but will be assumed here) which allows two-dimensional velocities to be specified uniquely. Thus

$$c_i\,\dot\theta \;-\; \oint_{\partial\Omega(t)} w_i\, u\,\nabla\phi\cdot\mathbf{n}\; d\Gamma \;+\; \int_{\Omega(t)} u\,\nabla\phi\cdot\nabla w_i\; d\Omega \;=\; \int_{\Omega(t)} w_i\, Lu\; d\Omega\,, \tag{7}$$

is solved to obtain velocity potentials. The velocities themselves are then recovered from the vector equations

$$\int_{\Omega(t)} w_i\,\dot{\mathbf{x}}\; d\Omega \;=\; \int_{\Omega(t)} w_i\,\nabla\phi\; d\Omega\,. \tag{8}$$

Even in one dimension, where specifying the curl of the velocity field is not required for uniqueness, the introduction of a velocity potential still has the value of replacing one non-symmetric linear system with two symmetric ones.

Application of the finite element method leads to discrete forms of (7) and (8) given by

$$C_i\,\dot\theta \;-\; \oint_{\partial\Omega(t)} W_i\, U\,\nabla\Phi\cdot\mathbf{n}\; ds \;+\; \int_{\Omega(t)} U\,\nabla\Phi\cdot\nabla W_i\; d\Omega \;=\; \int_{\Omega(t)} W_i\, LU\; d\Omega \tag{9}$$

and

$$\int_{\Omega(t)} W_i\,\dot{\mathbf{X}}\; d\Omega \;=\; \int_{\Omega(t)} W_i\,\nabla\Phi\; d\Omega\,. \tag{10}$$

Note that from now on, $\mathbf{X}\approx\mathbf{x}$, $\dot{\mathbf{X}}\approx\dot{\mathbf{x}}$, $\Phi\approx\phi$ and $U\approx u$ are assumed to be local (piecewise linear in this case) finite element functions, and $W_i \approx w_i$ are the usual piecewise linear basis functions moving with the velocity field $\dot{\mathbf{X}}$. The variable i can now be thought of as a mesh node index. A standard Galerkin finite element approach is used to carry out the approximation, on a mesh of $N + B$ nodes, the first N of which are the interior nodes, the remainder being on the boundary. It is possible to use other (upwind, for example) finite element methods, and future research may show this to be preferable.

A single time-step of the moving mesh finite element method consists of the following stages [1].

a Given a set of mesh nodes with positions $\mathbf{X}_i$ and a total mass θ, solve the discrete form of (5),

$$\int_{\Omega(t)} W_i U \, d\Omega \;=\; C_i \theta \tag{11}$$

to recover the solution values U_i at all of the mesh nodes, $i = 1, \ldots, N + B$. The constants c_i and θ are calculated from the initial conditions (solution and mesh). Application of boundary conditions to this equation will be discussed in more detail Section 3.

b Having found appropriate values for the U_i, solve (9) for $i = 1, \ldots, N+B$, to obtain the velocity potentials Φ at the mesh nodes and, if it is not known explicitly, $\dot{\theta}$. Invertibility of the resulting system is ensured by imposing $\Phi = 0$ at one or more nodes, typically chosen to be on the boundary.

For the purposes of this work, when $\dot{\theta}$ is not known explicitly, in order to ensure that the number of equations matches the number of unknowns $\Phi = 0$ will be specified at one node but the equation associated with that point will be retained.

c Use the resulting values of Φ_i to obtain the mesh node velocities via (10) for $i = 1, \ldots, N + B$, with no boundary conditions imposed.

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

$$\frac{d(\vec{\mathbf{X}}, \theta)}{dt} \;=\; \vec{F}(\vec{\mathbf{X}}, \theta)\,. \tag{12}$$

This system of ordinary differential equations in time can then be integrated using an appropriate method (here it is simply forward Euler).

3 Boundary Conditions, Conservation and the PME

The issues presented here will be illustrated using the Porous Medium Equation (PME)

$$\frac{\partial u}{\partial t} \;=\; \nabla \cdot (u^n \nabla u) \tag{13}$$

where $n > 0$ is an integer exponent, with the Dirichlet boundary condition $u = 0$. This permits solutions with finite, but time-dependent, support in which mass (the integral of the dependent variable u over the whole domain) is conserved, *i.e.* $\dot{\theta} = 0$.

This gives explicitly an expression for LU, the discrete counterpart of the right hand side of (1), which leads to

$$- \oint_{\partial\Omega(t)} W_i \, U \, \nabla\Phi \cdot \mathbf{n} \, d\Gamma \; + \int_{\Omega(t)} U \, \nabla\Phi \cdot \nabla W_i \, d\Omega \; = \; - \int_{\Omega(t)} U^n \, \nabla W_i \cdot \nabla U \, d\Omega \tag{14}$$

in place of (9), which must be solved alongside (10) and (11). The additional condition of $U = 0$ on the boundary, then gives

$$\int_{\Omega(t)} U \, \nabla\Phi \cdot \nabla W_i \, d\Omega \; = \; - \int_{\Omega(t)} U^n \, \nabla W_i \cdot \nabla U \, d\Omega \, . \tag{15}$$

The boundary conditions for (15) are taken to here be $\Phi = 0$ on the whole boundary [1], so the equations themselves are only solved for internal nodes. One implication of this is that

$$\dot{\mathbf{X}} \cdot \mathbf{s} \; = \; \nabla\Phi \cdot \mathbf{s} \; = \; 0 \, , \tag{16}$$

for any tangent $\mathbf{s}$ to the boundary. In other words, boundary nodes can only move perpendicular to the boundary. This is compatible with imposing zero curl on the mesh velocity field and appropriate to the radially symmetric two-dimensional test cases used below. In more complex situations, the condition can easily be relaxed because $\Phi = 0$ needs only to be imposed at one boundary node to obtain $\vec{\Phi}$ uniquely. This does not change the approach described below. In fact the radial symmetry of the problem means that solving the system with $\Phi = 0$ at one node on the boundary actually leads to $\Phi = 0$ on the whole boundary, and the solutions are virtually indistinguishable.

When the mesh velocities are recovered from (10) the boundary values of $\dot{\mathbf{X}}$ are left free, though clearly other conditions may be applied if more information is known about the boundary's behaviour.

Of prime interest here is the application of the boundary condition $U = 0$ to (11). It has already been applied in the derivation of (15) and so is implicitly included in the resulting mesh velocity field. However, (15) and (10) are only approximations, and (12) is not integrated exactly, so these velocities will not necessarily provide new nodal positions which are exactly compatible with both mass conservation and the Dirichlet boundary condition. Therefore $U = 0$ must be imposed directly on the recovery of U in (11) if it is needed precisely. It is not immediately obvious how to impose this condition while simultaneously maintaining exact mass conservation.

Separately, the two conditions are easy to satisfy:

- exact boundary conditions with approximate mass conservation can be achieved by overwriting the boundary equations with $U = 0$, but solving (11) for internal nodes only. It guarantees that the mass associated with the internal nodes is conserved, not the total mass.

- approximate boundary conditions with exact mass conservation can be achieved by solving (11) for all nodes without any overwriting.

Both approaches rely on the mesh velocities being approximated accurately (the approach is designed to do this) for the other condition to be recovered adequately. However, for both $U = 0$ and conservation of mass to be enforced *exactly*, a slight adjustment in viewpoint is required.

3.1 A Mass-Conserving Dirichlet Boundary Condition

The moving mesh finite element method is driven by the local conservation principles of (11) for each of the mesh nodes i. These are used both to recover the dependent variable U and (after differentiating with respect to time) to calculate the mesh velocity $\dot{\mathbf{X}}$ via the velocity potential Φ. Since imposing $U = 0$ on the boundary involves overwriting equations, in order to conserve mass globally as well, information carried by the boundary equations must somehow be associated with internal nodes. Furthermore, this must be done in a manner which retains invertible systems.

Two approaches are proposed.

1 It follows from (11) that the proportion of the overall mass associated with each internal node remains constant, leading to

$$\int_{\Omega(t)} W_i\, U \; d\Omega \;=\; C_i'\,\theta' \tag{17}$$

for the internal nodes, $i = 1, \ldots, N$, where the coefficients

$$C_i' \;=\; \frac{\int_{\Omega(t)} W_i\, U \; d\Omega}{\sum_{j=1}^{N} \int_{\Omega(t)} W_j\, U \; d\Omega} \tag{18}$$

and the "internal" mass

$$\theta' \;=\; \sum_{j=1}^{N} \int_{\Omega(t)} W_j\, U \; d\Omega \tag{19}$$

are calculated from the initial conditions and fixed throughout the computation. Simple algebraic manipulation of (17) leads to

$$\int_{\Omega(t)} W_i\, U \; d\Omega \,+\, C_i' \sum_{j=N+1}^{N+B} \int_{\Omega(t)} W_j\, U \; d\Omega \;=\; C_i'\,\theta\,. \tag{20}$$

Equation (20) can then be solved for U_i, $i = 1, \ldots, N$ with $U = 0$ substituted for the boundary nodes $i = N+1, \ldots, N+B$. It can be confirmed that this guarantees that mass is conserved exactly by summing (20) over the internal nodes, noting that $\sum_{j=1}^{N} c_i' = 1$.

Note that differentiation of (17) with respect to time leads directly to (9), as before, but only for the internal nodes. No constraint is explicitly applied at the boundary nodes, but at a later stage the boundary node equations for Φ are overwritten by $\Phi_i = 0$ for $i = N+1, \ldots, N+B$. Unfortunately, direct differentiation of (20) leads to a different, singular, set of equations for the velocity potentials: constraining the nodal masses *and* the total mass to be constant overspecifies the system, so a constraint must be eliminated prior to solving the equations for the velocity potential.

Differentiating (17) instead of (20) effectively ignores the boundary equations associated with

$$\sum_{j=N+1}^{N+B} \frac{d}{dt} \int_{\Omega(t)} W_j \, U \, d\Omega \;=\; 0\,, \tag{21}$$

leaving a non-singular system for the velocity potentials (the same one as (11) leads to when only internal nodes are considered). The removal of the singularity in this manner is related to applying $\Phi = 0$ at the boundary nodes, in which case the derived boundary equations are overwritten, even though the boundary nodes are included in the initial conservation principle. This is because a conservation constraint is ignored whenever an equation for U or Φ is overwritten by strongly imposing Dirichlet boundary conditions at a node. The penalty for doing this is that, although it removes the singularity it leads instead to values for Φ which imply velocities in (10) which may not be compatible with exact conservation, even if calculated precisely.

Although it achieves the primary objective ($U = 0$ on the boundary and constant θ), this approach still has two significant shortcomings: both the symmetry and (in more than one space dimension) the sparsity of the system generated for recovering U are lost. Even in simple cases this leads to crippling computational expense.

2 Approach **1** can be thought of as distributing the total mass associated with the boundary nodes to the internal nodes, in proportion to the amount of mass originally associated with each internal node. This boundary mass has to be redistributed for the scheme to be conservative, but it could be sent to any nodes. From an efficiency point of view it is most convenient to associate it with adjacent internal nodes. For example, in one dimension (with nodes numbered $0, 1, \ldots, N, N+1$) this leads to the equations

$$\begin{aligned} \int_{X_0}^{X_{N+1}} (W_0 + W_1)\, U \, dx &= (c_0 + c_1)\,\theta \\ \int_{X_0}^{X_{N+1}} W_i U \, dx &= c_i\,\theta \quad \text{for } i = 2, \ldots, N-1 \\ \int_{X_0}^{X_{N+1}} (W_N + W_{N+1})\, U \, dx &= (c_N + c_{N+1})\,\theta \end{aligned} \tag{22}$$

It is then implicitly assumed that the mass associated with each of the two boundary nodes (0 and $N+1$) remains constant, which allows the associated equations to be eliminated from the derivation of the velocity potential equations and creates a non-singular system to be solved for Φ.

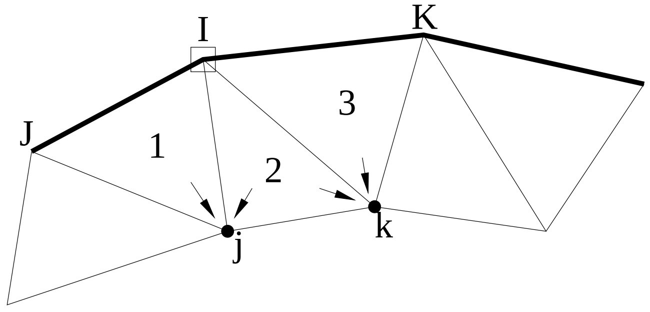

Figure 1: Boundary nodes for two-dimensional boundary conditions.

The two-dimensional situation is a little more complicated. Consider a node, I say, on the boundary: the mass associated with it, according to the connectivity shown in Figure 1, is given by

$$\begin{aligned}
&\int_{\Omega(t)} W_I\, U\, d\Omega \\
&= \int_{\triangle_1} W_I\, U\, d\Omega + \int_{\triangle_2} W_I\, U\, d\Omega + \int_{\triangle_3} W_I\, U\, d\Omega \\
&= \int_{\triangle_1} W_I\,(W_I\, U_I + W_J\, U_J + W_j\, U_j)\, d\Omega + \int_{\triangle_2} W_I\,(W_I\, U_I + W_j\, U_j + W_k\, U_k)\, d\Omega \\
&\qquad + \int_{\triangle_3} W_I\,(W_I\, U_I + W_k\, U_k + W_K\, U_K)\, d\Omega \qquad (23)
\end{aligned}$$

Now, since U_I, U_J and U_K are constrained by the Dirichlet boundary conditions to retain the same value for all time, the mass associated with these terms in the above integral will also remain constant (whether or not $\dot{\theta} = 0$). This means that they can safely be ignored in the recovery of U, as long as they are eliminated from both the left and right hand sides of the equation used to recover the dependent variable (11). The remaining contribution of the integral is then

$$\begin{aligned}
&\int_{\triangle_1} W_I\,(W_j\, U_j)\, d\Omega + \int_{\triangle_2} W_I\,(W_j\, U_j + W_k\, U_k)\, d\Omega + \int_{\triangle_3} W_I\,(W_k\, U_k)\, d\Omega \\
&= \underbrace{\int_{\Omega(t)} W_I\,(W_j\, U_j)\, d\Omega}_{\text{add to equation } j} + \underbrace{\int_{\Omega(t)} W_I\,(W_k\, U_k)\, d\Omega}_{\text{add to equation } k} \qquad (24)
\end{aligned}$$

The contributions are combined with the equations of nodes adjacent to the boundary in the manner illustrated by (22) and the resulting equations solved for the internal values of U. A a similar approach can be followed, however many internal nodes are adjacent to I, though it can be noted that in the special case where node I has no adjacent internal nodes, the use of linear finite elements means that the Dirichlet boundary condition implies that the mass for this cell does not change with time and can therefore be ignored. In fact $U = 0$ has been assumed at the boundary nodes throughout the calculations carried out in (24).

The additional terms in (24) are, as in approach **1**, implicitly assumed to be a constant proportion of the overall mass, which means that they may be ignored in the derivation of the velocity potentials and these equations also remain as (15). The advantage of this alternative approach is that it simply adds terms to the diagonal of the mass matrix to be inverted to find U, thus maintaining symmetry and sparsity. In fact, the additional terms make the matrix more diagonally dominant and hence easier to invert.

An alternative view of both of the above approaches is that they modify the test functions W_i. In the first approach the test functions are no longer local, which explains the loss of sparsity, but in the second their support is the same as it would be for the standard linear hat functions. Typical two-dimensional test functions are shown in Figure 2. An example is shown for an internal node and a boundary node for each of the two approaches.

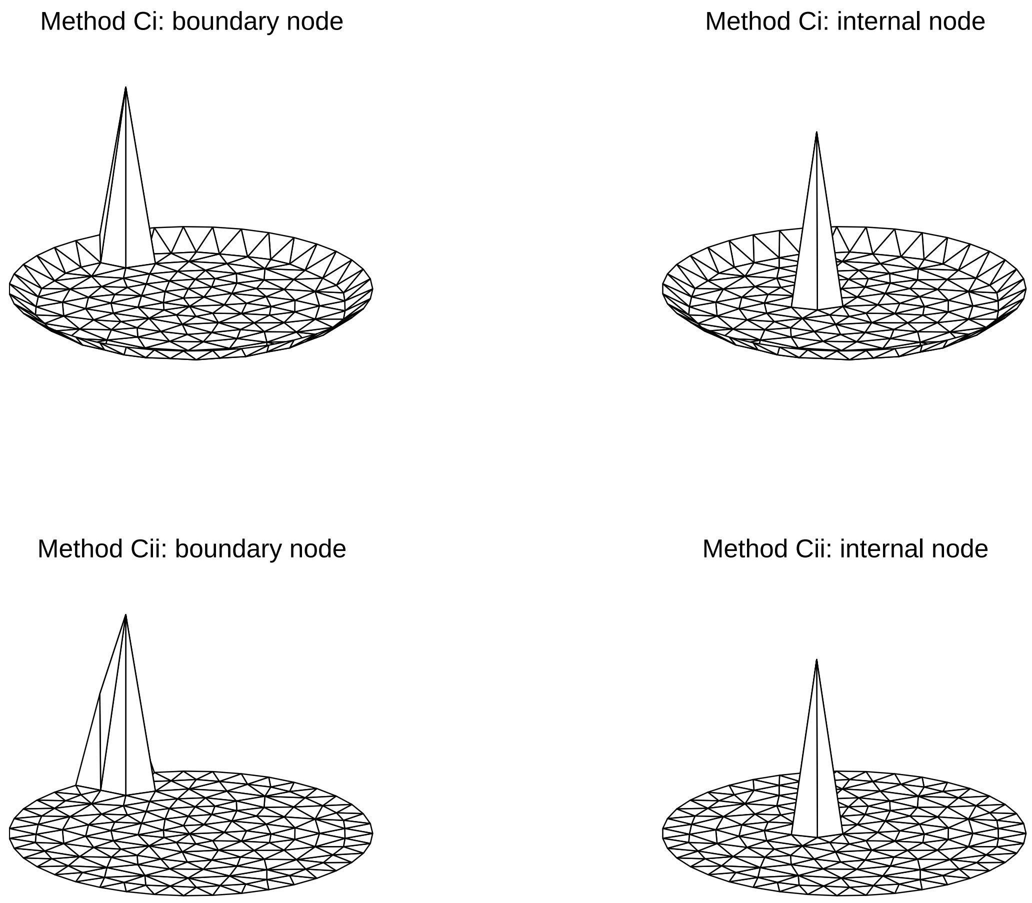

Figure 2: Test functions for the recovery of U via mass conservation in two dimensions which are compatible with exact imposition of Dirichlet boundary conditions.

This gives the four options which will be compared in Section 3.2:

A $U = 0$ imposed exactly, mass conservation approximated.

B $U = 0$ recovered approximately, mass conserved exactly.

C $U = 0$ imposed exactly and mass conserved exactly,

  i with boundary mass distributed across the domain.

  ii with boundary mass distributed to adjacent nodes.

Note that other monitors can be used to drive the mesh movement [3], but although analogous approaches to applying the boundary conditions exist, they have not yet been investigated. It also remains to investigate the behaviour of the method proposed here, which solves for $\dot{\theta}$ rather than trying to impose it, when it is combined with alternative monitor functions. This is partly because in these cases the moving mesh method generally results in a nonlinear conservation principle which cannot yet be inverted in a robust enough manner to warrant direct recovery of U. A conservative Arbitrary Lagrangian-Eulerian (ALE) approach is preferred for the recovery of U in this situation [3], though this reduces to (11) when the mass monitor is used.

3.2 Numerical Experiments

Equation (13) admits a family of compact support similarity solutions with moving boundaries at which $u = 0$ [4]. Two cases are considered, one with exponent $n = 1$ (for which the slope of the self-similar solution at the moving boundary is finite) and another with $n = 3$ (for which the slope at the boundary is infinite). It can be shown that in d space dimensions (here $d = 1$ or 2) a radially symmetric self-similar solution exists of the form

$$u(r,t) = \begin{cases} \frac{1}{\lambda^d(t)}\left(1 - \left(\frac{r}{r_0\lambda(t)}\right)^2\right)^{\frac{1}{n}} & |r| \le r_0\lambda(t) \\ 0 & |r| > r_0\lambda(t) \end{cases} \tag{25}$$

in which r is the usual radial coordinate, and where

$$\lambda(t) = \left(\frac{t}{t_0}\right)^{\frac{1}{2+dn}} \qquad \text{and} \qquad t_0 = \frac{{r_0}^2 n}{2(2+dn)} \, . \tag{26}$$

The problem is parameterised by the initial front position r_0 (at time t_0) and the position of the moving front is given by $r_0\lambda(t)$. The test cases are run until time $T = t - t_0$.

Results are shown for two test cases each in one and two space dimensions, as follows.

- One dimension, $n = 1$, $r_0 = 0.5$ (giving $t_0 = 0.04167$) and run until $T = 10$ (when $r \approx 3.11154$).

- One dimension, $n = 3$, $r_0 = 0.5$ (giving $t_0 = 0.075$) and run until $T = 10$ (when $r \approx 1.33231$).

- Two dimensions, $n = 1$, $r_0 = 0.5$ (giving $t_0 = 0.03125$) and run until $T = 0.1$ (when $r \approx 0.71578$).

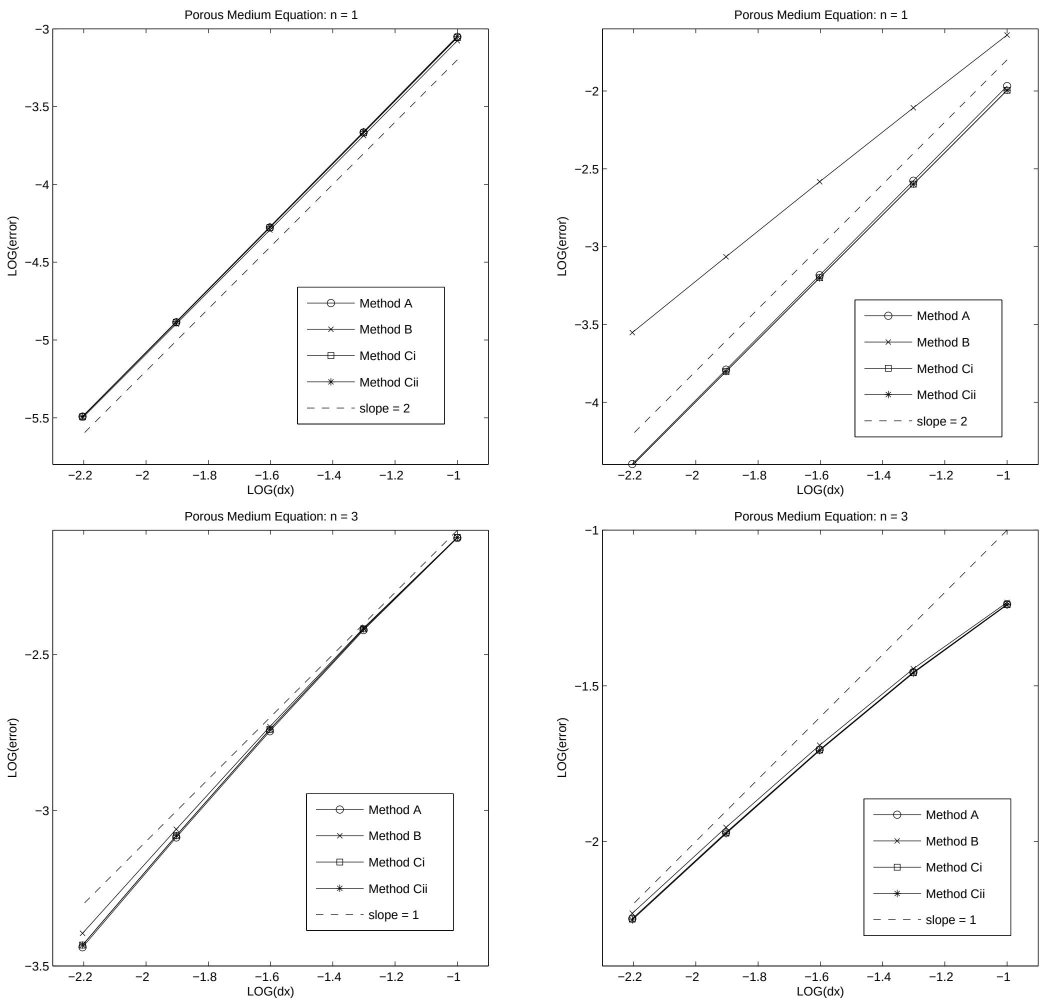

Figure 3: Comparison of L^2 errors in the solution (left) and L^∞ errors in the boundary node positions (right) for the different boundary approximations in one space dimension.

Table 1: Numerical estimates of the orders of accuracy of the scheme for the dependent variable U and the boundary position $\mathbf{X}$ using each set of boundary conditions, obtained by comparing errors on the finest pair of meshes used in the experiments.

	One dimension				Two dimensions			
	$n = 1$		$n = 3$		$n = 1$		$n = 3$	
Method	U	$\mathbf{X}$	U	$\mathbf{X}$	U	$\mathbf{X}$	U	$\mathbf{X}$
A	2.01	2.01	1.17	0.92	2.10	2.46	1.02	1.31
B	2.00	1.62	1.11	0.92	1.91	2.02	1.00	1.30
Ci	2.02	2.00	1.17	0.92	–	–	–	–
Cii	2.02	2.00	1.17	0.92	2.00	2.50	1.01	1.31

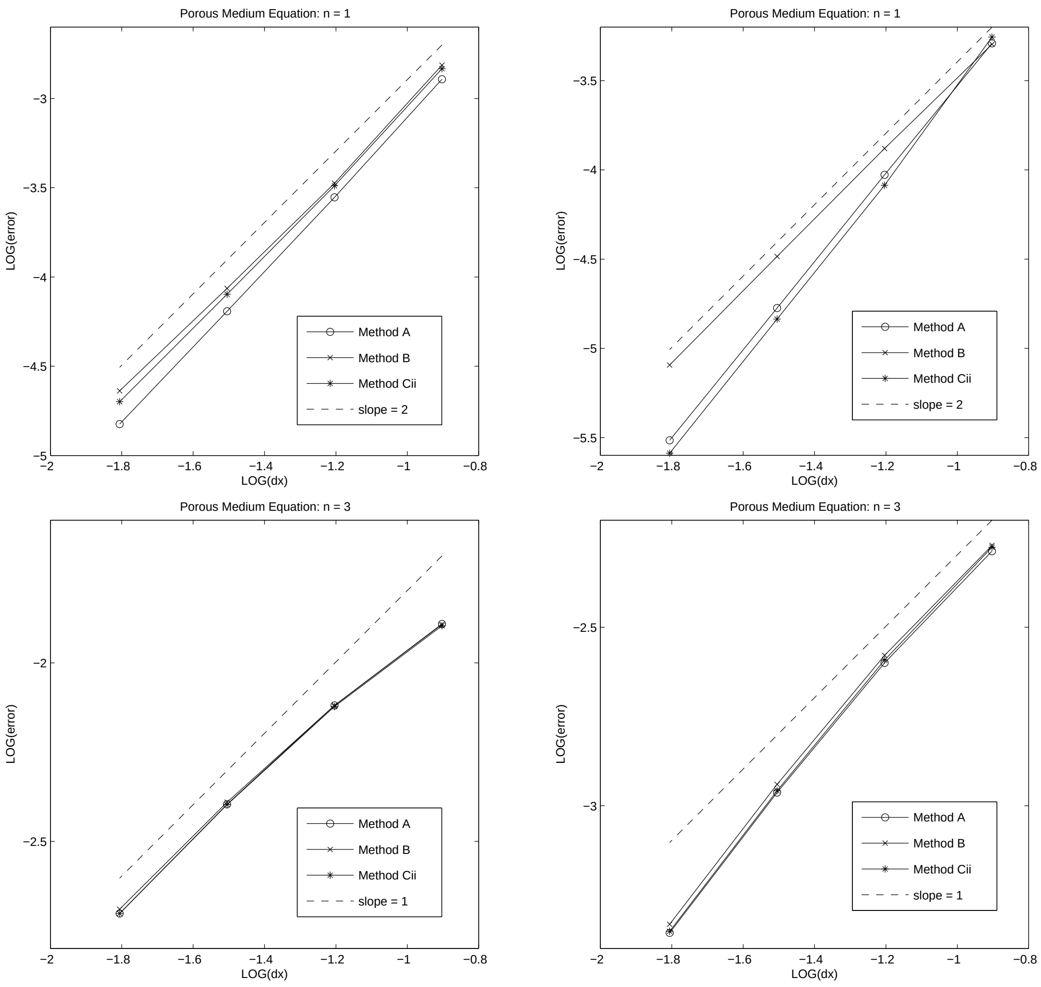

Figure 4: Comparison of L^2 errors in the solution (left) and boundary node positions (right) for the different boundary approximations in two space dimensions.

- Two dimensions, $n = 3$, $r_0 = 0.5$ (giving $t_0 = 0.046875$) and run until $T = 0.1$ (when $r \approx 0.57673$).

All results have been obtained on uniform meshes, for which the representative *initial* mesh size dx is repeatedly halved to estimate orders of accuracy. The time-step is divided by four with each refinement. The results shown have been calculated with $\dot{\theta} = 0$ imposed explicitly with $\phi = 0$ on the whole boundary, but experiments with the new approach which solves for $\dot{\theta}$ along with ϕ give results which are indistinguishable to about 8 decimal places. The errors are shown in Figures 3 and 4, while numerically estimated orders of accuracy are given in Table 1 These lead to a number of observations.

- There can be significant differences between the errors in approximating the boundary position even when the errors in the solution approximation are very similar.

- Typically, the method is second order accurate when $n = 1$ and first order accurate when $n = 3$, whatever approach is used model the boundary conditions. The exception is the approximation to the boundary position in two dimensions, which appears to be of higher order in each case.

- Overall, the least accurate approach seems to be to conserve mass exactly and impose $U = 0$ weakly (method **B**). This is most noticeable when considering the error in the boundary position. The other approaches all give similar results in each case.

- In terms of computational speed, methods **A**, **B** and **Cii** are very similar. In fact method **Cii** is typically the fastest, seemingly because the adjustments made to the matrix improve its conditioning and so accelerate the convergence of the conjugate gradient algorithm used to solve for the dependent variable. Method **Ci** is significantly slower (by factors of more than 100 for even moderately fine meshes such as the 2113 node mesh used as part of the two-dimensional experiments), leading to its omission from the two-dimensional results.

- There is almost no difference in quality between the results obtained using methods **Ci** and **Cii**.

- Method **A** seems less robust than the others. In this case all results were necessarily obtained using half the time-step of the other approaches due to stability problems and mesh tangling.

- All of the methods allow the dependent variable to become negative, but the nature of the solutions tested has meant that this has only happened when the Dirichlet boundary condition is not enforced exactly. The negative values (when they appear) are only small, but there may be practical circumstances in which even this would be unacceptable.

Of the four proposed, method **Cii**, which allows mass to be conserved exactly simultaneously with exact imposition of the Dirichlet boundary conditions, appears to have the best combination of accuracy, robustness, speed and physical realism and so is proposed for future use. It has already been shown to have robust counterparts which can be implemented in other situations (where mass is not conserved and/or $u \neq 0$ on the boundary – which can be due to modelling PDEs other than the PME) but these have not yet been thoroughly tested in terms of their accuracy compared to the simpler approaches.

4 Internal Boundaries and Stefan Problems

Consider a domain divided in to inner and outer regions, Ω_I and Ω_O respectively, as illustrated in Figure 5. The two-phase Stefan problem, which represents the transition between, for example, liquid and solid phases, is modelled by the equation

$$u_t \;=\; \nabla^2 u \qquad \text{in } \Omega_I \text{ and } \Omega_O \tag{27}$$

with Dirichlet boundary conditions $u = u_I = 0$ on Γ_I, the interface between Ω_I and Ω_O, and $u = u_O$ on Γ_O the outer boundary. In addition, conditions are also placed on the velocities of the boundaries:

$$[\nabla u \cdot \mathbf{n}] \;=\; C_L\, \dot{\mathbf{x}} \cdot \mathbf{n} \qquad \text{on } \Gamma_I \tag{28}$$

where $\mathbf{n}$ is the outward pointing unit normal to the given domain, and for the purposes of this problem, the outer boundary is assumed to be fixed, *i.e.* $\dot{x} = \mathbf{0}$ on Γ_O. `Must check which way the jump is taken! One-phase Stefan suggests it's liquid-solid, though it is clearly linked with the meaning of the sign of CL.` We will assume here that the inner region is solid and the outer region is liquid. `It may be wise to change the subscripts from I(nner) and O(uter) to S(olid) and L(iquid).`

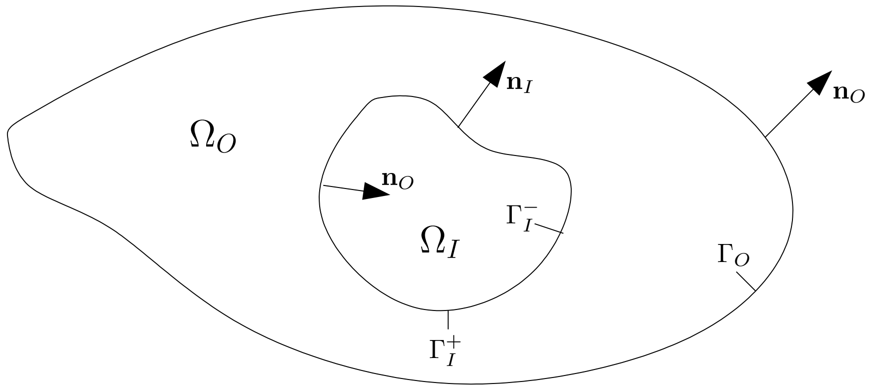

Figure 5: A generic domain for the two-phase Stefan problem.

For the purposes of clarity, the mesh nodes are assumed to be numbered as follows:

- Nodes $1, \ldots, N_I$ are those in the interior of Ω_I.

- Nodes $N_I+1,\ldots,N_I+B_I$ are those on the interface Γ_I. In Figure **??** this is indicated by both Γ_I^-, the boundary of Ω_I, and Γ_I^+, the coincident boundary of Ω_O.
- Nodes $N_I+B_I+1,\ldots,N_I+B_I+N_O$ are those in the interior of Ω_O.
- Nodes $N_I+B_I+N_O+1,\ldots,N_I+B_I+N_O+B_O$ are those on the interface of Γ_O.

4.1 Calculating the Mesh Velocity Potential

Following the approach presented earlier, differentiating the weak form

$$\int_{\Omega(t)} W_i\,U\,d\Omega \;=\; C_i\,\theta(t)\,, \tag{29}$$

with respect to time leads, ultimately, to

$$-\int_{\Omega(t)} \nabla W_i\cdot(\nabla U+U\dot{\mathbf{X}})\,d\Omega \;+\; \oint_{\Gamma(t)} W_i(\nabla U+U\dot{\mathbf{X}})\cdot\mathbf{n}\,d\Gamma \;=\; C_i\,\dot\theta \tag{30}$$

where $\Omega(t) = \Omega_O(t)\cup\Omega_I(t)$ and $\Gamma(t) = \Gamma_O\cup\Gamma_I^-(t)\cup\Gamma_I^+(t)$. With a small amount of manipulation to move the unknowns (largely) to the left hand side, and substituting in the velocity potential, this then becomes

$$\begin{aligned}\int_{\Omega(t)} \nabla W_i\cdot U\nabla\Phi\,d\Omega \;+\; C_i\,\dot\theta \qquad&\\ =\;-\int_{\Omega(t)} \nabla W_i\cdot\nabla U\,d\Omega \;+\; \oint_{\Gamma(t)} W_i\,\nabla U\cdot\mathbf{n}\,d\Gamma \;+\; \oint_{\Gamma(t)} W_i\,U\,\dot{\mathbf{X}}\cdot\mathbf{n}\,d\Gamma&\end{aligned} \tag{31}$$

It would be nice to be able to solve this in the same way across the whole domain but, for reasons which will be elaborated on later, this does not work. Hence, at this stage, the integrals are separated into components over each of the domains which leads to

$$\begin{aligned}\int_{\Omega_I(t)} \nabla W_i\cdot U\nabla\Phi\,d\Omega \;+\; \int_{\Omega_O(t)} \nabla W_i\cdot U\nabla\Phi\,d\Omega \;+\; C_i^I\,\dot\theta_I \;+\; C_i^O\,\dot\theta_O&\\ =\;-\int_{\Omega_I(t)} \nabla W_i\cdot\nabla U\,d\Omega \;-\; \int_{\Omega_O(t)} \nabla W_i\cdot\nabla U\,d\Omega&\\ +\oint_{\Gamma_I^-(t)} W_i\,\nabla U\cdot\mathbf{n}_I\,d\Gamma \;+\; \oint_{\Gamma_I^+(t)} W_i\,\nabla U\cdot\mathbf{n}_O\,d\Gamma \;+\; \oint_{\Gamma_O} W_i\,\nabla U\cdot\mathbf{n}_O\,d\Gamma&\\ +\oint_{\Gamma_I^-(t)} W_i\,U\,\nabla\Phi\cdot\mathbf{n}_I\,d\Gamma \;+\; \oint_{\Gamma_I^+(t)} W_i\,U\,\nabla\Phi\cdot\mathbf{n}_O\,d\Gamma \;+\; \oint_{\Gamma_O} W_i\,U\,\nabla\Phi\cdot\mathbf{n}_O\,d\Gamma&\end{aligned} \tag{32}$$

All of the final three terms disappear, either because $U=0$ on Γ_I or $\dot{\mathbf{X}}=\mathbf{0}$ on Γ_O. The remaining boundary integrals along do not cancel out because ∇U is discontinuous across this boundary.

For what follows it is assumed that no cell spans either Ω_I or Ω_O, *i.e.* no cell has all of its vertices on Γ_I and no cell has vertices on both Γ_I and Γ_O. Given this, the following five situations arise.

1. For $i = 1, \ldots, N_I$, $W_i = 0$ on Γ_I, Γ_O and Ω_O so
$$\int_{\Omega_I(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; + \; C_i^I \, \dot{\theta}_I \; = \; - \int_{\Omega_I(t)} \nabla W_i \cdot \nabla U \, d\Omega \tag{33}$$

2. For $i = N_I + 1, \ldots, N_I + B_I$, but considering only integration over the region Ω_I, since $U = 0$ on Γ_I (the process would be slightly more complicated if $U = U_I \neq 0$),
$$\int_{\Omega_I(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; + \; C_i^I \, \dot{\theta}_I$$
$$= \; - \int_{\Omega_I(t)} \nabla W_i \cdot \nabla U \, d\Omega \; + \; \oint_{\Gamma_I^-(t)} W_i \, \nabla U \cdot \mathbf{n}_I \, d\Gamma \tag{34}$$
which, having applied the boundary condition at the interface in the form
$$\oint_{\Gamma_I^-} W_i \, \nabla U \cdot \mathbf{n}_I \, d\Gamma \; + \; \oint_{\Gamma_I^+} W_i \, \nabla U \cdot \mathbf{n}_O \, d\Gamma \; = \; \oint_{\Gamma_I} W_i \, C_L \, \nabla \phi \cdot \mathbf{n}_O \, d\Gamma \tag{35}$$
becomes
$$\int_{\Omega_I(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; - \; \oint_{\Gamma_I(t)} W_i \, C_L \, \nabla \Phi \cdot \mathbf{n}_O \, d\Gamma \; + \; C_i^I \, \dot{\theta}_I$$
$$= \; - \int_{\Omega_I(t)} \nabla W_i \cdot \nabla U \, d\Omega \; - \; \oint_{\Gamma_I^+(t)} W_i \, \nabla U \cdot \mathbf{n}_O \, d\Gamma \tag{36}$$

3. For $i = N_I + 1, \ldots, N_I + B_I$, but considering only integration over the region Ω_O, the same procedures carried out in stage 2 lead to
$$\int_{\Omega_O(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; + \; C_i^O \, \dot{\theta}_O$$
$$= \; - \int_{\Omega_O(t)} \nabla W_i \cdot \nabla U \, d\Omega \; + \; \oint_{\Gamma_I^+(t)} W_i \, \nabla U \cdot \mathbf{n}_O \, d\Gamma \tag{37}$$
and then
$$\int_{\Omega_O(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; - \; \oint_{\Gamma_I(t)} W_i \, C_L \, \nabla \Phi \cdot \mathbf{n}_O \, d\Gamma \; + \; C_i^O \, \dot{\theta}_O$$
$$= \; - \int_{\Omega_O(t)} \nabla W_i \cdot \nabla U \, d\Omega \; - \; \oint_{\Gamma_I^-(t)} W_i \, \nabla U \cdot \mathbf{n}_I \, d\Gamma \tag{38}$$

4. For $i = N_I + B_I + 1, \ldots, N_I + B_I + N_O$, $W_i = 0$ on Γ_I, Γ_O and Ω_I so the equations used are the same as those used in stage 1 for the interior of Ω_I.
$$\int_{\Omega_O(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; + \; C_i^O \, \dot{\theta}_I \; = \; - \int_{\Omega_O(t)} \nabla W_i \cdot \nabla U \, d\Omega \tag{39}$$

5. For $i = N_I + B_I + N_O + 1, \ldots, N_I + B_I + N_O + B_O$, $W_i = 0$ on Γ_I and Ω_I, and $\dot{\mathbf{X}} = \mathbf{0}$ on Γ_O so
$$\int_{\Omega_O(t)} \nabla W_i \cdot U \nabla \Phi \, d\Omega \; + \; C_i^O \, \dot{\theta}_O$$
$$= \; - \int_{\Omega_O(t)} \nabla W_i \cdot \nabla U \, d\Omega \; + \; \oint_{\Gamma_O} W_i \, \nabla U \cdot \mathbf{n}_O \, d\Gamma \tag{40}$$

It is typical to apply homogeneous Neumann boundary conditions at this boundary, so in many cases the final term, given by the boundary integral, disappears.

Clearly this implies that Φ is dual valued on Γ_I. This is not a problem because the recovery of $\dot{\mathbf{X}}$ from Φ only requires integrals over mesh cells, even when the boundary condition for $\dot{\mathbf{X}}$ is not explicitly enforced at the interface.

$\dot{\theta}_I$ is found by assuming that $\Phi_i = 0$ at an arbitrary node i for which $1 \le i \le N_I + B_I$, but keeping all of the equations. Similarly $\dot{\theta}_O$ is obtained in the same manner, assuming $\Phi_i = 0$ for any i where $N_I + 1 \le i \le N_I + B_I + N_O + B_O$.

Remark: when $i = N_I + 1, \ldots, N_I + B_I$, integrating over the whole domain leads to

$$\int_{\Omega(t)} \nabla W_i \cdot U \nabla\Phi \, d\Omega \;+\; C_i^I \,\dot{\theta}_I \;+\; C_i^O \,\dot{\theta}_O \tag{41}$$
$$= \;-\int_{\Omega(t)} \nabla W_i \cdot \nabla U \, d\Omega \;+\; \oint_{\Gamma_I^-(t)} W_i \,\nabla U \cdot \mathbf{n}_I \, d\Gamma \;+\; \oint_{\Gamma_I^+(t)} W_i \,\nabla U \cdot \mathbf{n}_O \, d\Gamma$$

which, having applied the boundary condition at the interface, becomes

$$\int_{\Omega_I(t)} \nabla W_i \cdot U \nabla\Phi \, d\Omega \;-\; \oint_{\Gamma_I(t)} W_i \, C_L \,\nabla\Phi \cdot \mathbf{n}_O \, d\Gamma \;+\; C_i^I \,\dot{\theta}_I \;+\; C_i^O \,\dot{\theta}_O$$
$$= \;-\int_{\Omega_I(t)} \nabla W_i \cdot \nabla U \, d\Omega \tag{42}$$

All of this relies on the boundary condition being applied as

$$\oint_{\Gamma_I^-(t)} W_i \,\nabla U \cdot \mathbf{n}_I \, d\Gamma \;+\; \oint_{\Gamma_I^+(t)} W_i \,\nabla U \cdot \mathbf{n}_O \, d\Gamma \;=\; \oint_{\Gamma_I(t)} W_i \, C_L \,\nabla\Phi \cdot \mathbf{n}_) \, d\Gamma \tag{43}$$

where $C_L > 0$ represents freezing and $C_L < 0$ represents thawing. `This needs the sign checking, along with its relationship with CL.` This leaves us with a counting problem, since Φ would need to be specified at at least 2 points in order to free up enough equations to solve for both $\dot{\theta}_I$ and $\dot{\theta}_O$.

It is possible to solve for the total mass variation $\dot{\theta}$ but this requires extra work with including the Dirichlet boundary condition at the interface. The equations may need to be derived in the knowledge that ∇U is discontinuous here, and hence that $\nabla^2 u$ is infinite. `I say this as though it's more complicated, but the more I look at it the more it looks like it might actually be slightly simpler, not least because it would lead to a single-valued phi on the interface.`

4.2 Recovering the Mesh Velocity

The following can either be solved all at once or on Ω_I and Ω_O separately.

1. For $i = 1, \ldots, N_I$, solve

$$\int_{\Omega_I(t)} W_i \,\dot{\mathbf{X}} \, d\Omega \;=\; \int_{\Omega_I(t)} W_i \,\nabla\Phi \, d\Omega \tag{44}$$

2. For $i = N_I + 1, \ldots, N_I + B_I$, on either Ω_I or Ω_O, either solve

$$C_L \oint_{\Gamma_I(t)} W_i \, \dot{\mathbf{X}} \cdot \mathbf{n}_I \, d\Gamma \;=\; \oint_{\Gamma_I^-} W_i \, \nabla u \cdot \mathbf{n}_I \, d\Gamma \;+\; \oint_{\Gamma_I^+} W_i \, \nabla u \cdot \mathbf{n}_O \, d\Gamma \tag{45}$$

or impose

$$C_L \, \dot{\mathbf{X}}_i \cdot \mathbf{n} \;=\; [\nabla u \cdot \mathbf{n}]_i \tag{46}$$

along with $\dot{\mathbf{X}}_i \cdot \mathbf{s} = 0$. In the radially symmetric situations considered here it should also be valid to solve

$$C_L \oint_{\Gamma_I(t)} W_i \, \dot{\mathbf{X}} \, d\Gamma \;=\; \oint_{\Gamma_I^-} W_i \, \nabla u \, d\Gamma \;+\; \oint_{\Gamma_I^+} W_i \, \nabla u \, d\Gamma \tag{47}$$

directly to obtain both components of the velocity without explicitly imposing zero tangential velocity. `Is this generally safe? It clearly satisfies the desired boundary condition but do we want to impose a stronger condition?`

3. For $i = N_I + B_I + 1, \ldots, N_I + B_I + N_O$, solve

$$\int_{\Omega_O(t)} W_i \, \dot{\mathbf{X}} \, d\Omega \;=\; \int_{\Omega_O(t)} W_i \, \nabla \Phi \, d\Omega \tag{48}$$

4. For $i = N_I + B_I + N_O + 1, \ldots, N_I + B_I + N_O + N_I$, simply impose

$$\dot{\mathbf{X}}_i = \mathbf{0} \tag{49}$$

Initial experiments in one dimension suggest that the inclusion of the interface boundary condition in the calculation of the velocity potential makes it unnecessary to do so again here so it may be possible (and would definitely be simpler) to instead solve the following

1. For $i = 1, \ldots, N_I + B_I + N_O$, solve

$$\int_{\Omega_I(t)} W_i \, \dot{\mathbf{X}} \, d\Omega \;=\; \int_{\Omega_I(t)} W_i \, \nabla \Phi \, d\Omega \tag{50}$$

2. For $i = N_I + B_I + N_O + 1, \ldots, N_I + B_I + N_O + N_I$, simply impose

$$\dot{\mathbf{X}}_i = \mathbf{0} \tag{51}$$

4.3 Recovering the Dependent Variable

The following can be solved on Ω_I and Ω_O independently.

1. For $i = 1, \ldots, N_I$, solve

$$\int_{\Omega_I(t)} \tilde{W}_i \, U \, d\Omega \;=\; \tilde{C}_i^I \, \theta_I \tag{52}$$

2. For $i = N_I + 1, \ldots, N_I + B_I$, impose $U_i = U_I = 0$.

3. For $i = N_I + B_I + 1, \ldots, N_I + B_I + N_I$, solve

$$\int_{\Omega_O(t)} \tilde{W}_i \, U \, d\Omega \;=\; \tilde{C}_i^O \, \theta_O \tag{53}$$

4. For $i = N_I + B_I + N_O + 1, \ldots, N_I + B_I + N_O + N_I$, impose $U_i = U_O$.

Note that $\tilde{W}_i$ are the modified test functions of Section 3 which allow U to be specified strongly on Dirichlet boundaries, with $\tilde{C}_i^I$ and $\tilde{C}_i^O$ being the corresponding proportions of the total masses. Also, $\tilde{C}_i^I = 0$ for $i = N_I + 1, \ldots, N_I + B_I + N_O + B_O$ and $\tilde{C}_i^O = 0$ for $i = 1, \ldots, N_I + B_I$ and $i = N_I + B_I + N_O + 1, \ldots, N_I + B_I + N_O + B_O$.

Note also that, although it might well be possible to solve a single system over the whole domain instead of the two separate systems split by th emoving interface, U changes sign between the two domains, which means that it is no longer entirely appropriate as a monitor function unless the two domains are considered separately.

4.4 Single Phase Stefan Problems

The single phase Stefan problem can simply be considered to be a special case of the two-phase problem described above in which there is a single domain and it is assumed that $u = u_I = 0$ in the neighbouring region, adjacent to the moving interface. The only modification to be made is that solution in the second domain must be avoided, since $\nabla u = 0$ here and this would lead to a singularity in the calculation of the velocity potential.

5 Results

With new $\dot{\theta}$...

For PME and 1/2-phase Stefan...

One and two dimensions...

Similarity solutions, Frank spheres, other?

6 Conclusions

EXPAND THIS SECTION.

Four approaches to the imposition of Dirichlet boundary conditions on the dependent variable of a time-dependent, mass-conserving PDE on a time-dependent domain have been compared for a newly developed moving mesh finite element method. The porous medium equation has been used in one and two space dimensions to test the different approaches. The results suggest that a method which imposes both mass conservation and boundary conditions exactly is the best. Of the two methods proposed which satisfy both of these properties, the one which maintained the sparsity (and appears to improve the conditioning) of the mass matrix which is inverted to find the dependent variable, was considerably more efficient. This approach (and its variants derived for other situations) is proposed for use with the moving mesh finite element method in the future.

References

[1] Baines MJ, Hubbard ME, Jimack PK. A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries. *Appl Numer Math*, to appear.

[2] Baines MJ, Hubbard ME, Jimack PK. A moving mesh finite element algorithm for fluid flow problems with moving boundaries. *Int J Numer Meth Fl*, to appear.

[3] Baines MJ, Hubbard ME, Jimack PK, Jones AC. Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions. *Appl Numer Math*, submitted.

[4] Murray JD, Mathematical Biology: An Introduction (3rd edition), Springer (2002).