

A Moving Mesh Finite Element Algorithm for the Adaptive Solution of Time-Dependent Partial Differential Equations with Moving Boundaries

M.J. Baines^a M.E. Hubbard^b P.K. Jimack^b

^a*Department of Mathematics, The University of Reading, UK*

^b*School of Computing, University of Leeds, UK*

Abstract

A moving mesh finite element algorithm is proposed for the adaptive solution of nonlinear diffusion equations with moving boundaries in one and two dimensions. The moving mesh equations are based upon conserving a local proportion, within each patch of finite elements, of the total “mass” that is present in the projected initial data. The accuracy of the algorithm is carefully assessed through quantitative comparison with known similarity solutions, and its robustness is tested on more general problems.

Applications are shown to a variety of problems involving time-dependent partial differential equations with moving boundaries. Problems which conserve mass, such as the porous medium equation and a fourth order nonlinear diffusion problem, can be treated by a simplified form of the method, while problems which do not conserve mass require the full theory.

Key words: time-dependent nonlinear diffusion, moving boundaries, finite element method, Lagrangian meshes, conservation of mass

1 Introduction

In this paper an adaptive finite element method is proposed for the solution of partial differential equations (PDEs) with moving boundaries, using a moving mesh. The approach is prompted by recent interest in geometric integration and scale invariance (see for example [10] and references therein) which has rekindled interest in the use of adaptive moving meshes. Scale invariance treats independent and dependent variables alike, suggesting that both solution and

mesh should be varied when designing numerical schemes to inherit this property.

The use of moving meshes has been proposed in many different contexts over the past three decades, ranging from phase change problems [4,8,21], blow-up problems [9] or hyperbolic conservation laws [16], to general classes of time-dependent problem [2,15,22]. It is apparent from this significant body of research that moving grids have much to offer in terms of improved efficiency although, in order to provide robust and reliable software, they must often be applied in conjunction with other adaptive techniques such as local remeshing [24] or order enrichment [14]. In addition, when there is sufficient *a priori* knowledge to ensure that the initial mesh provides an adequate resolution to avoid the need for later refinement, moving grids can provide an extremely powerful computational tool in their own right.

In the approach taken in this work the moving mesh equations are based upon conserving the local proportion, within each patch of finite elements, of the total integral (mass) of the dependent variable across the domain. Although not considered here, the integral may be generalised to conserve other quantities [6,7], yielding an approach similar to that of using a monitor function to control the movement of the mesh, as in the Moving Mesh Partial Differential Equation (MMPDE) method [4,17] for example. It is also strongly related to the Deformation method of Liao and co-workers [19,20] and to the Geometric Conservation Law (GCL) method of Cao, Huang and Russell [11] (see also [26]). In particular the idea of a mesh velocity potential proposed in [11] is exploited in order to obtain uniqueness of the mesh velocity in greater than one space dimension.

The organisation of the paper is as follows. In the next section strong and weak formulations of the PDE in a moving reference frame are discussed, together with the local conservation principle upon which the proposed method is based. Section 3 contains the details of the method, which is a generalisation of the one-dimensional finite volume approach described in [6,7]. Applications are presented in Sections 4 and 5. The mass conserving applications in Section 4, for which the theory simplifies, consist of moving boundary problems governed by the porous medium equation (PME) and a fourth order nonlinear diffusion problem. Numerical results are presented in this section for comparison against known similarity solutions (in order to assess the accuracy of the technique), as well as problems for which there is no known analytic solution (to assess its robustness). Section 5 is devoted to the application of the technique to two non-mass-conserving problems, consisting of a one-phase Stefan problem and a diffusion problem with a negative source term. The paper concludes with a discussion of the work and a number of suggestions as to how it may be extended.

2 Background

Throughout this and the following section an abstract time-dependent partial differential equation (PDE) of the general form

$$\frac{\partial u}{\partial t} = Lu, \quad (1)$$

on a time-dependent domain $\Omega(t)$ will be considered. Specific equations will be treated in the subsequent sections. In the PDE (1) $u = u(\mathbf{x}, t)$ is defined in a fixed frame of reference with coordinate \mathbf{x} at time t and L is a differential operator involving space derivatives only. For clarity all descriptions of the method will be given in the context of two space dimensions with the simplification to a single dimension following easily from this. In Section 2.1 the ways in which (1) may be expressed in a moving frame of reference are described and then in Section 2.2 the underlying conservation principle behind the proposed numerical method is introduced.

2.1 Fixed and moving frames

Instead of working in the fixed (Eulerian) frame it is possible to take a Lagrangian viewpoint in which \mathbf{x} is taken to be a moving coordinate $\mathbf{x}(t)$, the result of a time-dependent mapping from a fixed set of reference coordinates, $\mathbf{a} = \mathbf{x}(0)$ say. Define an invertible mapping between the fixed coordinates \mathbf{a} and the moving coordinates \mathbf{x} , at time t , of the form

$$\mathbf{x} = \hat{\mathbf{x}}(\mathbf{a}, t), \quad (2)$$

so that

$$u(\mathbf{x}, t) = u(\hat{\mathbf{x}}(\mathbf{a}, t), t) = \hat{u}(\mathbf{a}, t) \quad (3)$$

say, where $\hat{u}, \hat{\mathbf{x}}$ are Eulerian. Then by the chain rule,

$$\frac{\partial \hat{u}}{\partial t} = \frac{\partial \hat{\mathbf{x}}}{\partial t} \cdot \nabla u + \frac{\partial u}{\partial t}. \quad (4)$$

Hence, writing

$$\dot{u} = \frac{\partial \hat{u}}{\partial t}, \quad \dot{\mathbf{x}} = \frac{\partial \hat{\mathbf{x}}}{\partial t}, \quad (5)$$

the PDE (1) in the moving frame becomes

$$\dot{u} - \dot{\mathbf{x}} \cdot \nabla u = Lu. \quad (6)$$

Clearly, in order to determine both of the unknown Eulerian velocities, \dot{u} and $\dot{\mathbf{x}}$, an additional equation is required. This will be discussed in the next subsection. Prior to this, though, two integral forms of (6) are described.

Let $\Omega(0)$ be a reference test volume at $t = 0$ and $\Omega(t)$ be the corresponding test volume in the moving frame \mathbf{x} , given by (2), at time t . Hence $\Omega(t)$ may be thought of as a moving test volume. Application of Leibnitz' rule (or Reynolds' Transport Theorem) yields the equation

$$\frac{d}{dt} \int_{\Omega(t)} u \, d\Omega = \int_{\Omega(t)} \frac{\partial u}{\partial t} \, d\Omega + \oint_{\partial\Omega(t)} u \dot{\mathbf{x}} \cdot \mathbf{n} \, ds = \int_{\Omega(t)} \left(\frac{\partial u}{\partial t} + \nabla \cdot (u \dot{\mathbf{x}}) \right) \, d\Omega. \quad (7)$$

The PDE (1) (in the moving frame) can therefore be written in the integral form (cf. (6))

$$\frac{d}{dt} \int_{\Omega(t)} u \, d\Omega - \int_{\Omega(t)} \nabla \cdot (u \dot{\mathbf{x}}) \, d\Omega = \int_{\Omega(t)} Lu \, d\Omega. \quad (8)$$

In order to be able to apply a finite element method the integral expressions (7) and (8) must be generalised to suitable weak forms. This requires the introduction of a test function, w say, which moves with the velocity $\dot{\mathbf{x}}$, and therefore satisfies the advection equation

$$\frac{\partial w}{\partial t} + \dot{\mathbf{x}} \cdot \nabla w = 0. \quad (9)$$

Application of Leibnitz' rule gives the following generalisation of (7):

$$\begin{aligned} \frac{d}{dt} \int_{\Omega(t)} wu \, d\Omega &= \int_{\Omega(t)} \frac{\partial}{\partial t}(wu) \, d\Omega + \oint_{\partial\Omega(t)} wu \dot{\mathbf{x}} \cdot \mathbf{n} \, ds \\ &= \int_{\Omega(t)} \left(w \frac{\partial u}{\partial t} + u \frac{\partial w}{\partial t} + \nabla \cdot (wu \dot{\mathbf{x}}) \right) \, d\Omega. \end{aligned} \quad (10)$$

Making use of the property (9) the weak form of the PDE in the moving frame is (cf. (8))

$$\frac{d}{dt} \int_{\Omega(t)} wu \, d\Omega - \int_{\Omega(t)} w \nabla \cdot (u \dot{\mathbf{x}}) \, d\Omega = \int_{\Omega(t)} wLu \, d\Omega. \quad (11)$$

Equation (11) is the form of the PDE that will be used in the derivation of the algorithm that follows.

2.2 A distributed conservation principle

Equation (11) has been derived directly from the governing PDE (1) using the moving coordinates (2). Following [10], a principle is now proposed upon which the derivation of this moving coordinate system can be based. First, define the total mass $\theta(t)$ by

$$\theta(t) = \int_{\Omega(t)} u \, d\Omega, \quad (12)$$

where the test volume $\Omega(t)$ is chosen to be the entire spatial domain of the problem at time t , moving with velocity $\dot{\mathbf{x}}$. Motivated by the scale invariance arguments in [10] and references therein, let the velocity $\dot{\mathbf{x}}$ of the moving frame be determined by the weak conservation principle

$$\int_{\Omega(t)} w u \, d\Omega = c \theta(t), \quad (13)$$

where w is a test function advected with velocity $\dot{\mathbf{x}}$, satisfying (9), and c is a constant determined by w and the initial data. Equation (13) is an integral equation for u depending on $\theta(t)$ and the moving coordinate system.

Differentiation of (13) with respect to time yields

$$\frac{d}{dt} \int_{\Omega(t)} w u \, d\Omega = c \frac{d\theta}{dt} = c \dot{\theta}(t). \quad (14)$$

Hence the weak form (11) of the PDE becomes

$$c \dot{\theta}(t) - \int_{\Omega(t)} w \nabla \cdot (u \dot{\mathbf{x}}) \, d\Omega = \int_{\Omega(t)} w L u \, d\Omega \quad (15)$$

or, after integration by parts,

$$c \dot{\theta}(t) - \oint_{\partial\Omega(t)} w u \dot{\mathbf{x}} \cdot \mathbf{n} \, ds + \int_{\Omega(t)} u \dot{\mathbf{x}} \cdot \nabla w \, d\Omega = \int_{\Omega(t)} w L u \, d\Omega, \quad (16)$$

where \mathbf{n} is the unit outward normal. If $\dot{\theta}$ and u are given, the weak form (15) is in effect an equation for the divergence of $u\dot{\mathbf{x}}$. Hence, in order to determine $\dot{\mathbf{x}}$ uniquely additional constraints are required on the mesh velocity. In particular, following [11], if the vorticity $\nabla \times \dot{\mathbf{x}}$ is specified (together with suitable boundary conditions) then, given $\dot{\theta}$ and u , Equation (15) or (16) determines the velocity $\dot{\mathbf{x}}$ uniquely.

For example, suppose that \mathbf{q} is specified and that $\nabla \times \dot{\mathbf{x}} = \nabla \times \mathbf{q}$. Then there exists a velocity potential, ϕ say, such that

$$\dot{\mathbf{x}} = \mathbf{q} + \nabla\phi, \quad (17)$$

in which case (16) can be written

$$\begin{aligned} & c\dot{\theta}(t) - \oint_{\partial\Omega(t)} wu\nabla\phi \cdot \mathbf{n} \, ds + \int_{\Omega(t)} u\nabla\phi \cdot \nabla w \, d\Omega \\ &= \int_{\Omega(t)} wLu \, d\Omega + \oint_{\partial\Omega(t)} wu\mathbf{q} \cdot \mathbf{n} \, ds - \int_{\Omega(t)} u\mathbf{q} \cdot \nabla w \, d\Omega. \end{aligned} \quad (18)$$

If $\dot{\theta}(t)$ and u are known, Equation (18) can be used to determine ϕ , after which $\dot{\mathbf{x}}$ may be found from the weak form of (17),

$$\int_{\Omega(t)} w\dot{\mathbf{x}} \, d\Omega = \int_{\Omega(t)} w\nabla\phi \, d\Omega + \int_{\Omega(t)} w\mathbf{q} \, d\Omega, \quad (19)$$

which is the L^2 projection of $\mathbf{q} + \nabla\phi$ on to the space of \mathbf{x} .

All that remains, in principle at least, is to determine $\dot{\theta}(t)$ in (18). Differentiating (12) with respect to time and making use of the identity (8) gives

$$\dot{\theta}(t) = \int_{\Omega(t)} (Lu + \nabla \cdot (u\dot{\mathbf{x}})) \, d\Omega, \quad (20)$$

which may be evaluated (in terms of $\dot{\mathbf{x}}$) when u is known. In practice, Equations (18) and (20) will be solved simultaneously for ϕ and $\dot{\theta}$.

3 A Moving Finite Element Method

It is now possible to construct a moving grid finite element algorithm based upon the weak forms derived in the previous section, specifically equations

(13), (18), (19) and (20) for the unknowns $\dot{\mathbf{x}}$, ϕ , u and $\dot{\theta}$. In order to do this it is first necessary to produce an initial, reference, grid that covers the initial spatial domain $\Omega(0)$. Let this reference grid have $N + B$ vertices (the first N being in the interior of the domain and the remainder on the boundary, which is assumed to be a Dirichlet boundary for simplicity) at positions $\mathbf{A}_1, \dots, \mathbf{A}_{N+B}$. Also let $\hat{W}_i(\mathbf{a})$ ($i=1, \dots, N+B$) be the usual piecewise linear basis functions on this mesh (i.e. $\hat{W}_i(\mathbf{A}_j) = \delta_{ij}$). It is now possible to define a discrete form of (2) by

$$\mathbf{X}(t, \mathbf{a}) = \sum_{i=1}^{N+B} \mathbf{X}_i(t) \hat{W}_i(\mathbf{a}), \quad (21)$$

where $\mathbf{X}_i(0) = \mathbf{A}_i$ for $i = 1, \dots, N + B$. So long as no tangling occurs in the mesh $\mathbf{X}(t, \mathbf{a})$, then (21) is an invertible mapping and so it is possible to define

$$W_i(\mathbf{X}) = \hat{W}_i(\mathbf{a}) \quad (i = 1, \dots, N + B), \quad (22)$$

where \mathbf{a} is the point that maps to \mathbf{X} in (21) (i.e. $\mathbf{X} = \mathbf{X}(t, \mathbf{a})$). Note that it is easily verified that each of these functions W_i satisfies the constraint (9):

$$\begin{aligned} \frac{\partial W_i}{\partial t} &= \sum_{j=1}^{N+B} \frac{\partial W_i}{\partial \mathbf{X}_j} \cdot \dot{\mathbf{X}}_j \\ &= \sum_{j=1}^{N+B} (-W_j \nabla W_i) \cdot \dot{\mathbf{X}}_j \\ &= - \left(\sum_{j=1}^{N+B} W_j \dot{\mathbf{X}}_j \right) \cdot \nabla W_i \\ &= -\dot{\mathbf{X}} \cdot \nabla W_i. \end{aligned} \quad (23)$$

A proof that the second line follows from the first can be found in [18].

We are now able to define discrete equivalents to u and ϕ :

$$U(\mathbf{X}, t) = \sum_{i=1}^{N+B} U_i(t) W_i(\mathbf{X}), \quad (24)$$

$$\Phi(\mathbf{X}, t) = \sum_{i=1}^{N+B} \Phi_i(t) W_i(\mathbf{X}). \quad (25)$$

Having prescribed the finite element spaces, the method itself (consisting of discrete forms of (13), (18), (19) and (20)) may be expressed as the solution

of a system of ordinary differential equations (ODEs) in time of the form

$$\frac{d}{dt} \begin{pmatrix} \vec{\mathbf{X}} \\ \theta \end{pmatrix} = \vec{\mathbf{F}}(\vec{\mathbf{X}}, \theta), \quad (26)$$

where $\vec{\mathbf{X}} = (\mathbf{X}_1, \dots, \mathbf{X}_{N+B})^T$. The following sequence may be used to evaluate $\vec{\mathbf{F}}(\vec{\mathbf{X}}, \theta)$:

- (1) Given $\vec{\mathbf{X}}$ and θ , solve the discrete form of (13),

$$\int_{\Omega(t)} W_i U \, d\Omega = c_i \theta \quad (27)$$

for $i = 1, \dots, N + B$ to obtain U_1, \dots, U_{N+B} (and therefore U from (24)) in terms of θ and $\mathbf{X}_1, \dots, \mathbf{X}_{N+B}$. Note that since $\sum_{i=1}^{N+B} W_i = 1$ the total mass is conserved by (27) so long as the sum of the c_i is equal to 1. This is ensured when the c_i are calculated using each test function W_i with the initial data.

- (2) Given U , solve the discrete form of (18),

$$\begin{aligned} c_i \dot{\theta}(t) - \oint_{\partial\Omega(t)} W_i U \nabla \Phi \cdot \mathbf{n} \, ds + \int_{\Omega(t)} U \nabla \Phi \cdot \nabla W_i \, d\Omega \\ = \int_{\Omega(t)} W_i L U \, d\Omega + \oint_{\partial\Omega(t)} W_i U \mathbf{q} \cdot \mathbf{n} \, ds - \int_{\Omega(t)} U \mathbf{q} \cdot \nabla W_i \, d\Omega \end{aligned} \quad (28)$$

for $i = 1, \dots, N + B$, together with the discrete form of (20),

$$\dot{\theta}(t) = \int_{\Omega(t)} (L U + \nabla \cdot (U \dot{\mathbf{X}})) \, d\Omega, \quad (29)$$

to obtain $\Phi_1, \dots, \Phi_{N+B}$ (and therefore Φ from (25)) and $\dot{\theta}$. Note that, since only the gradient of Φ appears in (28), it is necessary to specify a value of Φ at (at least) one point, such as $\Phi_{N+B} = 0$, in order to ensure that Φ has a unique solution.

- (3) Given Φ , solve the discrete form of (19),

$$\begin{aligned} \int_{\Omega(t)} \begin{pmatrix} W_i \\ 0 \end{pmatrix} \cdot \dot{\mathbf{X}} \, d\Omega &= \int_{\Omega(t)} \begin{pmatrix} W_i \\ 0 \end{pmatrix} \cdot \nabla \Phi \, d\Omega + \int_{\Omega(t)} \begin{pmatrix} W_i \\ 0 \end{pmatrix} \cdot \mathbf{q} \, d\Omega, \\ \int_{\Omega(t)} \begin{pmatrix} 0 \\ W_i \end{pmatrix} \cdot \dot{\mathbf{X}} \, d\Omega &= \int_{\Omega(t)} \begin{pmatrix} 0 \\ W_i \end{pmatrix} \cdot \nabla \Phi \, d\Omega + \int_{\Omega(t)} \begin{pmatrix} 0 \\ W_i \end{pmatrix} \cdot \mathbf{q} \, d\Omega, \end{aligned} \quad (30)$$

for $i = 1, \dots, N + B$ to obtain $\dot{\mathbf{X}}_1, \dots, \dot{\mathbf{X}}_{N+B}$ (and therefore $\dot{\mathbf{X}}$ from (21)).

(4) Return $\vec{\mathbf{F}}$ (i.e. $\vec{\mathbf{X}}$ augmented with $\dot{\theta}$).

Any standard scheme for integrating the resulting initial value ODE system (26) may be used to complete the solution algorithm.

In this paper the system (26) has been integrated using an explicit time-stepping algorithm for simplicity (forward Euler, unless stated otherwise). The explicit nature of such schemes imposes a stability restriction on the maximum allowable time-step, and it is this restriction that has guided the time-step selection mechanism used in this work. Furthermore, unless stated otherwise, the initial data used in the examples that follow has been generated on a uniform initial mesh. This is done using a continuous piecewise linear least squares fit to the exact initial data at the mesh nodes. All results presented in this paper are computed using $\mathbf{q} \equiv 0$.

4 Mass-conserving Applications

We now consider applications of the method to problems which conserve mass. In these problems the total mass θ in (12) is constant and so $\dot{\theta}$ is zero. The algorithm is therefore simplified in that θ is no longer a dependent variable and there is no need to compute $\dot{\theta}$ in the algorithm in the previous section. Two moving boundary problems are presented, one involving the porous medium equation and the other being a fourth order nonlinear diffusion equation.

4.1 The Porous Medium Equation

The Porous Medium Equation (PME),

$$\frac{\partial u}{\partial t} = \nabla \cdot (u^m \nabla u) \quad (31)$$

(in a fixed reference frame), where $m > 0$ is an integer exponent, is a well-known model equation for gas flows in porous media, spreading liquids, etc. With the boundary condition $u = 0$, Equation (20) gives, on substitution of Lu ,

$$\dot{\theta}(t) = \oint_{\partial\Omega(t)} (u^m \nabla u + u \dot{\mathbf{x}}) \cdot \mathbf{n} \, ds = 0, \quad (32)$$

which implies that the total mass θ is conserved.

Equation (31) admits a family of compact support similarity solutions with moving boundaries on which $u = 0$, for which comparison results are also known, making it an ideal test problem. Further details may be found in [10] and the references therein.

When applying (27)-(30) to solve the PME in two dimensions a number of simplifications can be made. Since θ is constant it may be removed from the ODE system (26). The right-hand side of (27) is then simply a constant (for each i) that is determined from the initial data. In addition, θ disappears from Equation (28). This last equation may be further simplified when the Dirichlet boundary condition $u|_{\partial\Omega(t)} = 0$ is imposed and $\mathbf{q} \equiv 0$ is set:

$$\int_{\Omega(t)} U \nabla \Phi \cdot \nabla W_i \, d\Omega = - \int_{\Omega(t)} U^m \nabla U \cdot \nabla W_i \, d\Omega. \quad (33)$$

Specifying a value of Φ strongly at one node on the boundary ensures that this system has a unique solution however in this work the condition $\Phi = 0$ is applied on the whole moving boundary. This is equivalent to imposing the constraint that the tangential component of $\dot{\mathbf{X}}$ is zero on the boundary and means that (33) need only be solved for $i = 1, \dots, N$.

The choice $\mathbf{q} \equiv \mathbf{0}$ also simplifies Equations (30).

Throughout this paper it is assumed that any Dirichlet boundary conditions on u are applied weakly. By this it is meant that the Dirichlet condition on u is used to simplify (18) (and hence (28)) but that this condition is not imposed strongly on the solution U (i.e. U_{N+1}, \dots, U_{N+B} are still unknowns in (27)).

Results are presented here which demonstrate the accuracy of the method in one and two dimensions against known similarity solutions, [23]. Two cases are considered, with exponents $m = 1$ (for which the slope of the self-similar solution at the moving boundary is finite) and $m = 3$ (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

$$u(r, t) = \begin{cases} \frac{1}{\lambda^{d(t)}} \left(1 - \left(\frac{r}{r_0 \lambda(t)} \right)^2 \right)^{\frac{1}{m}} & |r| \leq r_0 \lambda(t) \\ 0 & |r| > r_0 \lambda(t) \end{cases} \quad (34)$$

in which r is the usual radial coordinate, and where

$$\lambda(t) = \left(\frac{t}{t_0} \right)^{\frac{1}{2+dm}} \quad \text{and} \quad t_0 = \frac{r_0^2 m}{2(2+dm)}. \quad (35)$$

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 $r_0\lambda(t)$. It should be emphasised that although in two dimensions the exact solution is radially symmetric the finite element meshes used in the following examples are not.

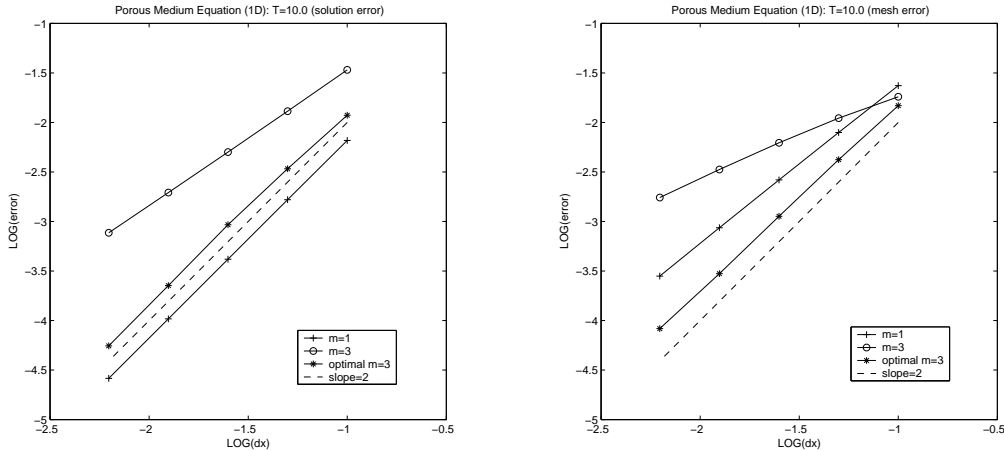


Fig. 1. Accuracy of the approximate solutions ($m = 1$ and 3) on a sequence of meshes at $T = 10$: solution error in the L^1 norm (left) and mesh error, *i.e.* the error in the position of the moving boundary, in the maximum norm (right).

Figure 1 illustrates the accuracy of the method in the one-dimensional case. The initial domain was taken to be $[-0.5, 0.5]$ for both $m = 1$ and $m = 3$ and, on the coarsest uniform mesh ($dx = 0.1$) a constant time-step of 0.0016 was used for $m = 1$ and 0.0064 for $m = 3$. In both cases the time-step was reduced by a factor of 4 each time the mesh was refined. On the optimal mesh for $m = 3$ the time-step taken on the coarsest mesh was 0.02, but each time the number of cells was doubled it was reduced by a factor of approximately 10. Second order accuracy is clearly demonstrated for $m = 1$ using a uniform initial mesh, although not for $m = 3$ in this case, presumably due to the infinite slope of the exact solution at the moving boundary. However, when the initial mesh is obtained by taking a least squares best fit to the initial conditions using *adjustable* nodes (see [3,25] for a description of the algorithm used), second order accuracy is also obtained for $m = 3$. This optimal initial grid has a much denser distribution of points near to the free boundary than the uniform mesh. The comparisons between the computational and the exact solutions are made at time $T = t - t_0 = 10$ (see (34) and (35)).

Figure 2 shows that second order accuracy is also obtained for $m = 1$ in the corresponding two-dimensional case but, once again, is not achieved when using a uniform initial mesh when $m = 3$. In both cases, the initial domain was a circle of radius 0.5 while the coarse mesh time-step was 0.0016 (which was reduced by a factor of 4 each time dx was halved). Obtaining an optimal mesh

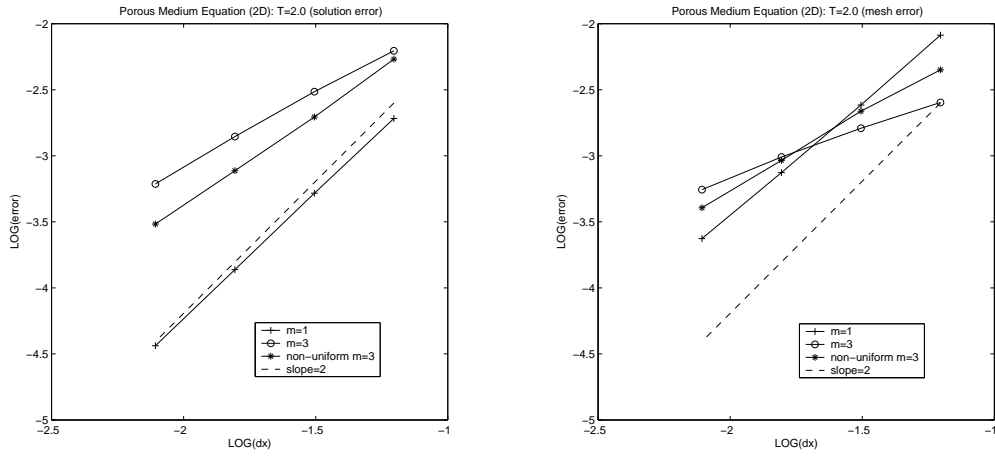


Fig. 2. Accuracy of the approximate solutions ($m = 1$ and 3) on a sequence of meshes at $T = 2$: solution error in the L^1 norm (left) and mesh error in the maximum norm (right).

in two dimensions is a significantly more computationally intensive task than in one dimension [3,25] and calculations using an optimal initial mesh have not been attempted here for $m = 3$. Instead, Figure 2 includes results from non-uniform initial meshes in which the nodes have been clustered towards the moving boundary. The order of accuracy is increased by this strategy but is still less than second order. The initial domain and time-step used were exactly those of the uniform meshes. The evolution of the two-dimensional test cases are illustrated in Figure 3. Note how the infinite slope of the solution at the boundary when $m = 3$ manifests itself through a non-zero solution value: an advantage of imposing the Dirichlet condition weakly is that it allows a more accurate representation of the initial conditions, and hence of the evolving solution. Note also that in all of these calculations the mass, θ in (12), is conserved perfectly, as expected.

4.2 A Fourth Order Problem

The fourth order diffusion equation,

$$\frac{\partial u}{\partial t} = \nabla \cdot (u^m \nabla p) \quad (36)$$

where $p = \nabla^2 u$ is a pressure, models the capillary effects in the coating of a solid surface by a thin liquid film [13]. It also admits compact support solutions

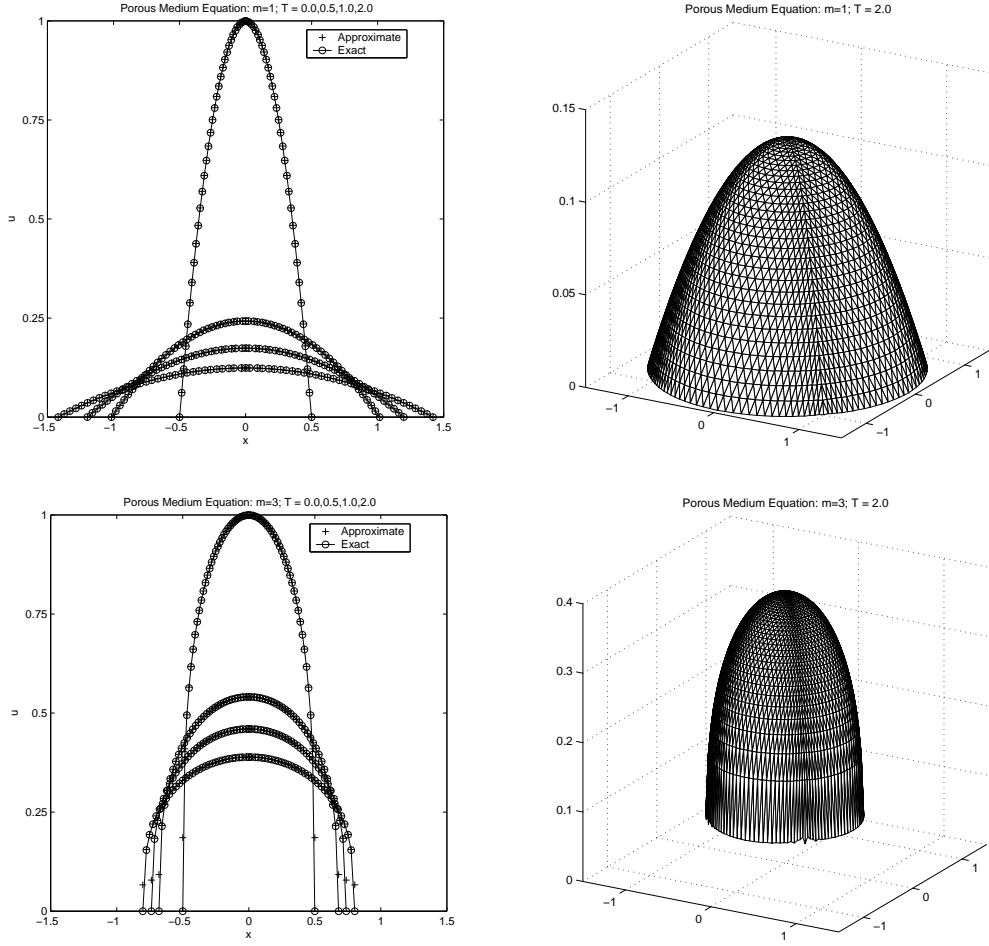


Fig. 3. Slices, along $y = 0$, of exact and approximate solutions for $m = 1$ and 3 at four different times (left) and the approximate solution surfaces at $T = 2$ (right). The results were obtained using a uniform 2113 node, 4096 cell mesh.

with a free boundary (for $0 < m < 3$) when

$$u|_{\partial\Omega(t)} = \frac{\partial u}{\partial n}\Big|_{\partial\Omega(t)} = 0. \quad (37)$$

The boundary conditions (37) ensure that the total mass of the system is conserved with time since

$$\begin{aligned} \frac{d}{dt} \int_{\Omega(t)} u \, d\Omega &= \int_{\Omega(t)} \frac{\partial u}{\partial t} \, d\Omega + \oint_{\partial\Omega(t)} u \mathbf{\dot{x}} \cdot \mathbf{n} \, ds \\ &= \int_{\Omega(t)} \nabla \cdot (u^m \nabla p) \, d\Omega = \oint_{\partial\Omega(t)} u^m \nabla p \cdot \mathbf{n} \, ds = 0. \end{aligned} \quad (38)$$

The algorithm used to approximate (36) is precisely that used for the PME, except for an additional step required to recover the approximation

$$P(\mathbf{X}, t) = \sum_{i=1}^{N+B} P_i(t)W_i(\mathbf{X}) \quad (39)$$

to the pressure $p(\mathbf{x}, t)$. This can be done using the weak form of $p = \nabla^2 u$ integrated by parts,

$$\int_{\Omega(t)} W_i P \, d\Omega = - \int_{\Omega} \nabla W_i \cdot \nabla U \, d\Omega \quad (40)$$

for $i = 1, \dots, N + B$, before solving for Φ in Step 2 of the algorithm in Section 3. Thus, the full method consists of solving, successively, (27), (40), (33) (with the final occurrence of U in this equation replaced by P) and (30).

Results are presented here for one- and two-dimensional problems, with the exponent $m = 1$. In this case, an exact radially symmetric solution to (36) in d space dimensions (see [13]) is given by

$$u(r, t) = \begin{cases} At^\beta U_0(1 - \eta^2)^2 & |r| \leq A^{\frac{m}{4}} t^\delta \\ 0 & |r| > A^{\frac{m}{4}} t^\delta \end{cases} \quad (41)$$

in which

$$\eta = \frac{r}{A^{\frac{m}{4}} t^\delta} \quad , \quad \delta = \frac{1}{4 + dm} \quad \text{and} \quad \beta = \frac{4\delta - 1}{m} . \quad (42)$$

This solution assumes an initial position for the moving front of $r = 1$, while A and U_0 have been chosen here to give $u(0, t_0) = 1$ (for which t_0 can be found).

Figure 4 illustrates the accuracy of the method in both the one- and two-dimensional cases. In one dimension fourth order accuracy is apparent (though this is reduced to second order if a *linear* least squares best fit is used to approximate the initial conditions instead of using the exact initial solution at the nodes). Uniform initial meshes were used throughout. A time-step of 0.0002 was used on the coarsest mesh ($dx = 0.2$) and this was reduced by a factor of 20 each time the mesh was refined. The accuracy reduces to second order in two dimensions. The coarse time-step here was 0.00001, with the same reduction factor with each refinement. The evolution of the two-dimensional test case is illustrated in Figure 5. Mass is again conserved exactly.

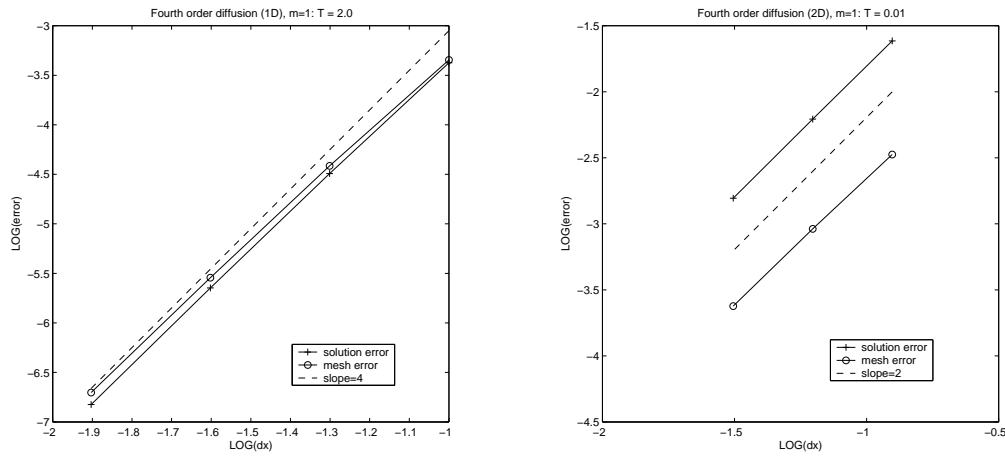


Fig. 4. Accuracy of the approximate solutions ($m = 1$) on a sequence of meshes at $T = 2$ in one dimension (left) and $T = 0.01$ in two dimensions (right). The solution error shown is in the L^1 norm while the mesh error in the maximum norm.

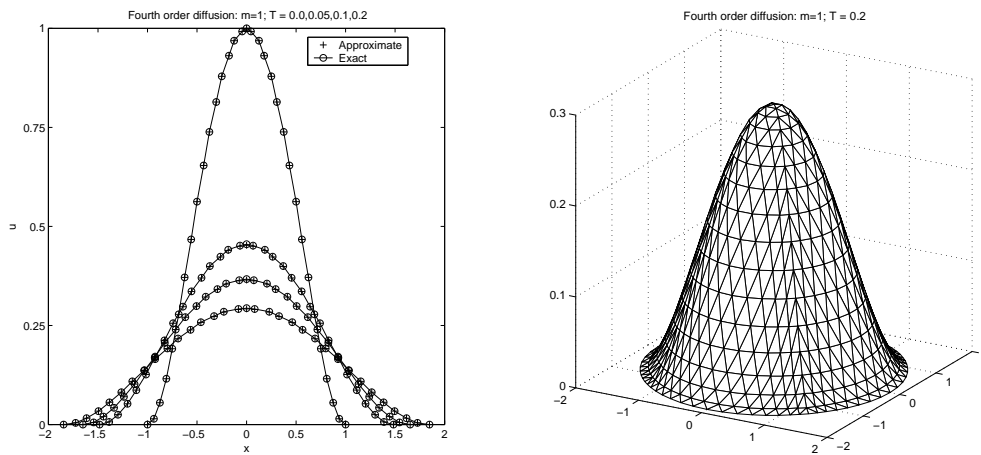


Fig. 5. Slices, along $y = 0$, of exact and approximate solutions for $m = 1$ at four different times (left) and the approximate solution surface at $T = 0.2$ (right). The results were obtained using a uniform 545 node, 1024 cell mesh.

4.3 Further Numerical Results

The method in two dimensions is not restricted to uniform meshes or to radially symmetric problems (which have simply been used to illustrate the accuracy of the technique via comparison with analytic solutions). Figure 6 shows the evolution, in the case of the PME with $m = 1$ and $m = 3$, of non-convex initial domains as they approach the similarity solutions associated with their initial masses.

Figure 7 shows the solution of an interesting test problem that has been se-

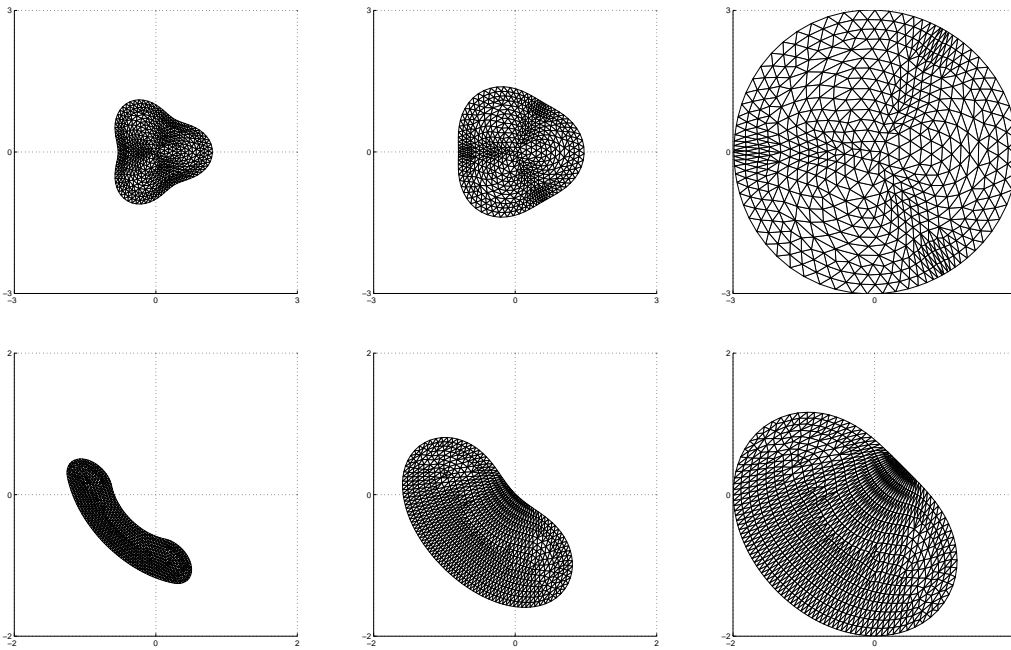


Fig. 6. Approximate solution surfaces at three different times for non-convex initial domains. The results were obtained for $m = 1$ (top) using a uniform but unstructured 615 node, 1149 cell mesh and for $m = 3$ (bottom) using a uniform, unstructured 931 node, 1727 cell mesh.

lected to illustrate one of the limitations of the current method (or implementation). The domain shown has evolved significantly from the initial condition to reach a point where it changes topology due to the two ends of the “horse shoe” intersecting each other. The current implementation of the method does not automatically account for intersections and so requires manual intervention (*e.g.* by interpolating the solution on to a new mesh with a more appropriate connectivity) in order to proceed beyond this point. This is a common feature of techniques that rely on moving grids alone and is generally overcome by automatic detection of the topology change triggering a discrete remesh.

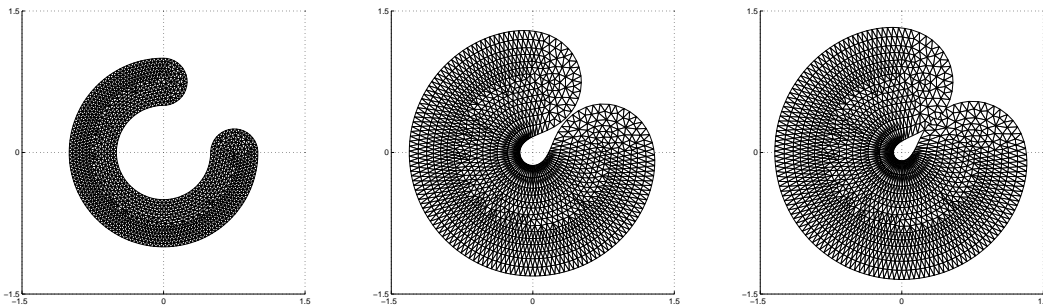


Fig. 7. Approximate solution surfaces at three different times for an initial non-convex domain which eventually overlaps with itself. The results were obtained using a uniform but unstructured 1534 node, 2850 cell mesh and the porous medium equation with $m = 1$.

5 Non-Mass-conserving Problems

Two types of non-mass-conserving problem are presented. In these cases $\dot{\theta}$ is not zero so θ also needs to be calculated as part of the solution procedure, as described in Section 3.

5.1 A Single-phase Stefan Problem

The single phase Stefan problem is given by the standard diffusion equation:

$$\frac{\partial u}{\partial t} = \nabla^2 u \quad (43)$$

with the boundary conditions:

$$\left. \frac{\partial u}{\partial n} \right|_{\Gamma_1} = C_L \dot{\mathbf{x}} \cdot \mathbf{n} \quad u|_{\Gamma_1} = u_B \quad (44)$$

Here $\partial\Omega = \Gamma_1(t) \cup \Gamma_2$ where $\Gamma_1(t)$ represents the moving boundary and Γ_2 represents any fixed portion of the computational boundary. In this work $\frac{\partial u}{\partial n}$ is imposed on the fixed boundary Γ_2 . The sign of C_L depends on whether freezing (positive) or thawing (negative) is being modelled by the problem, for which the single phase represented is the liquid phase.

The differences from the mass-conserving problems in the previous section arise mainly in the calculation of $\dot{\theta}$, which is no longer zero and therefore cannot be ignored. From (20), Equations (43) and (44) give

$$\begin{aligned} \dot{\theta}(t) &= \oint_{\partial\Omega(t)} (\nabla u + u\dot{\mathbf{x}}) \cdot \mathbf{n} \, ds \\ &= \oint_{\Gamma_1(t)} (C_L \dot{\mathbf{x}} + u\dot{\mathbf{x}}) \cdot \mathbf{n} \, ds + \oint_{\Gamma_2} (\nabla u + u\dot{\mathbf{x}}) \cdot \mathbf{n} \, ds \\ &= \oint_{\Gamma_1(t)} (C_L + u_B) \nabla \Phi \cdot \mathbf{n} \, ds + \oint_{\Gamma_2} \nabla u \cdot \mathbf{n} \, ds . \end{aligned} \quad (45)$$

For the final equality the constraint that $\dot{\mathbf{x}} \cdot \mathbf{n} = \mathbf{0}$ on the fixed part of the boundary, Γ_2 , has been used.

The equation for the velocity potential Φ at interior nodes ($i = 1, \dots, N$) is then given by (cf. (28))

$$c_i \dot{\theta}(t) + \int_{\Omega(t)} U \nabla \Phi \cdot \nabla W_i \, d\Omega = - \int_{\Omega(t)} \nabla U \cdot \nabla W_i \, d\Omega. \quad (46)$$

As in the previous examples, $\Phi = 0$ is imposed on the moving boundary $\Gamma_1(t)$ (i.e. the tangential component of $\dot{\mathbf{X}}$ is zero). However, Φ is not prescribed on the rest of the boundary so it is necessary to solve additional Φ equations for each node, b say, on Γ_2 :

$$c_b \dot{\theta}(t) + \int_{\Omega(t)} U \nabla \Phi \cdot \nabla W_b \, d\Omega = - \int_{\Omega(t)} \nabla U \cdot \nabla W_b \, d\Omega + \oint_{\Gamma_2} W_b \nabla u \cdot \mathbf{n} \, ds. \quad (47)$$

Note that $\frac{\partial \Phi}{\partial n} = 0$ on Γ_2 , so there is no boundary integral on the left-hand side of this equation. As explained in Section 3 the variable θ is updated together with $\dot{\mathbf{X}}$ at each time-step and new values fed back into (27) to find the new solution values U . The principle for moving the mesh is still that the proportion of the total mass associated with each node remains constant in time. The full method successively solves (27), (46)/(47) and (45), and (30) (with $\dot{\mathbf{X}}_b = 0$ for all nodes b on Γ_2).

We first consider a one-dimensional version of this problem with $C_L = -1$. This formulation permits the analytic solution

$$u(x, t) = \begin{cases} -1 + e^{-V(x-Vt)} & x \geq Vt \\ 0 & x < Vt, \end{cases} \quad (48)$$

for which V is the interface velocity and the initial conditions are set at $t_0 = 0$. Full details can be found in [1]. The case studied here takes $V = -1$ and an initial domain of $[-1, 0]$, however Figure 8 indicates that only first order accuracy is achieved in one dimension. The time-steps used were 0.0016 on the coarsest mesh, which was decreased by a factor of 4 at each refinement.

The two-dimensional test case models radially symmetric solidification and is known as the Frank spheres problem [1]. The exact solution is given by

$$u(r, t) = \begin{cases} u_\infty \left(1 - \frac{E_1(s^2/4)}{E_1(S^2/4)}\right) & s \geq S \\ 0 & s < S, \end{cases} \quad (49)$$

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

$$E_1(z) = \int_z^{\infty} \frac{e^{-t}}{t} dt. \quad (50)$$

The problem is parameterised 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

$$u_{\infty} = \frac{E_1(S^2/4)}{E_1'(S^2/4)}. \quad (51)$$

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

The two-dimensional results were obtained on a mesh covering an annulus, the moving boundary being the inner circle (which has radius S). An initial mesh was selected with higher resolution near the moving boundary than in the far field. The results of Figure 8 suggest that, unlike the one-dimensional case, better than first order accuracy is achieved. The reason for this is not clear, but it may be related to the fact that, in this case, the fixed boundary Γ_2 has been moved far enough away from the moving boundary (to $r = 10$) for it to have little effect: the solution derivatives being almost zero there. The time-step used was 0.01 on the coarsest mesh (reduced by a factor of 4 with each refinement). The evolution of the solution is illustrated in Figure 9. These results compare favourably with those in [1,12].

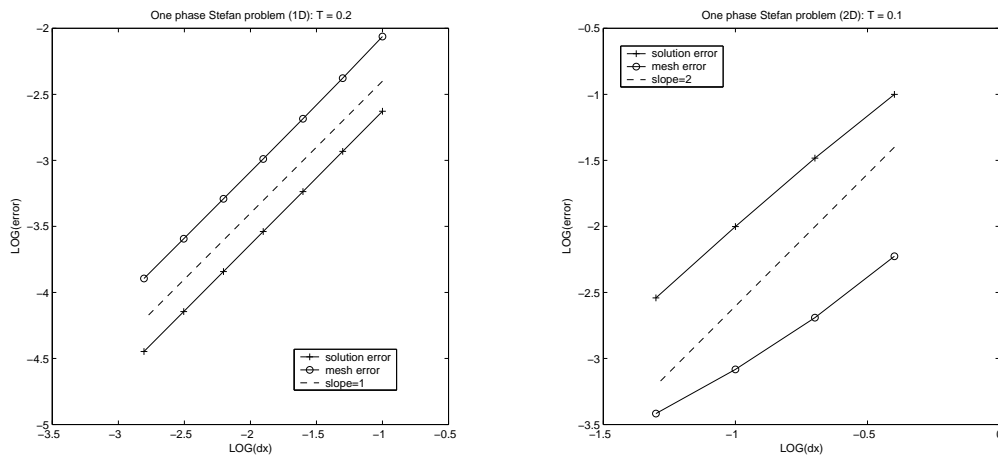


Fig. 8. Accuracy of the approximate solutions on a sequence of meshes at $T = 0.2$ in one dimension (left) and $T = 0.1$ in two dimensions (right). The solution error shown is in the L^1 norm while the mesh error in the maximum norm.

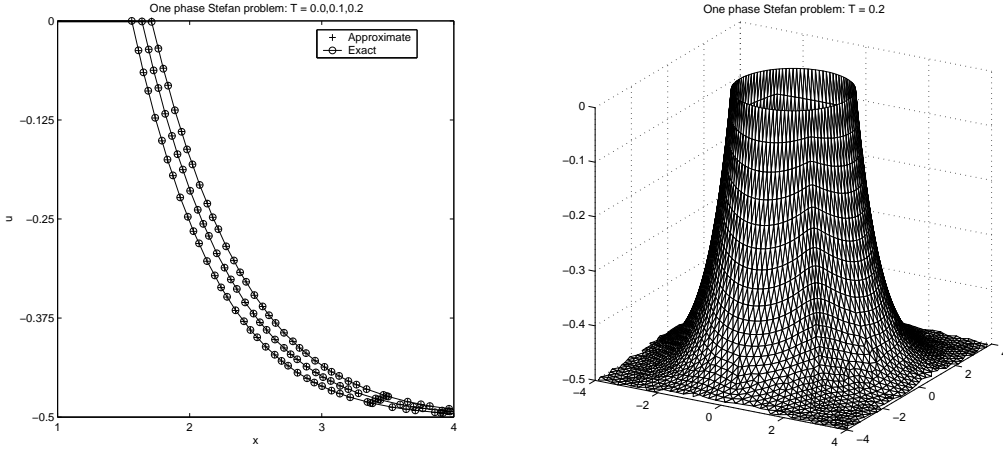


Fig. 9. Slices, along $y = 0$, of exact and approximate solutions at three different times (left) and the approximate solution surface at $T = 0.2$ (right). The results were obtained using a non-uniform 5258 node, 10275 cell mesh.

5.2 An Oxygen Diffusion/Absorption Problem

The final example included in this paper is a moving boundary problem with a source term, described in [5], which models the diffusion of oxygen in an absorbing medium. A straightforward generalisation of this can be considered in two dimensions, in which the whole of $\partial\Omega$ is a moving boundary. The problem can then be stated as

$$\frac{\partial u}{\partial t} = \nabla^2 u - 1 \quad \text{with} \quad u|_{\partial\Omega(t)} = \frac{\partial u}{\partial n}\Big|_{\partial\Omega(t)} = 0. \quad (52)$$

The calculation of $\dot{\theta}$ in (20) now takes the form

$$\begin{aligned} \dot{\theta}(t) &= \int_{\Omega(t)} \nabla \cdot (\nabla u + u\dot{\mathbf{x}}) - 1 \, d\Omega \\ &= \oint_{\partial\Omega(t)} (\nabla u + u\dot{\mathbf{x}}) \cdot \mathbf{n} \, ds - \int_{\Omega(t)} d\Omega \\ &= -|\Omega(t)|. \end{aligned} \quad (53)$$

The equations for the velocity potential are then given by (cf. (28))

$$c_i \dot{\theta}(t) + \int_{\Omega(t)} U \nabla \Phi \cdot \nabla W_i \, d\Omega = - \int_{\Omega(t)} \nabla U \cdot \nabla W_i \, d\Omega - \int_{\Omega(t)} W_i \, d\Omega \quad (54)$$

for $i = 1, \dots, N + B$. The full method successively solves (27), (54) and (53), and (30).

For the one-dimensional version of this problem an exact solution is known, [5]:

$$u(x, t) = \begin{cases} -x - t + e^{x+t-1} & x \leq 1 - t \\ 0 & x > 1 - t, \end{cases} \quad (55)$$

with initial conditions at $t_0 = 0$. In this one-dimensional case an additional boundary condition is also required at $x = 0$, and here the exact value of u_x (from (55)) is imposed. This appears in the numerical scheme as an extra term in both (53) and an additional equation of the form (54) associated with the boundary node at $x = 0$ (cf. the Γ_2 contributions to equations (45) and (47) from the previous example). Figure 10 shows that second order accuracy is again achieved for this problem.

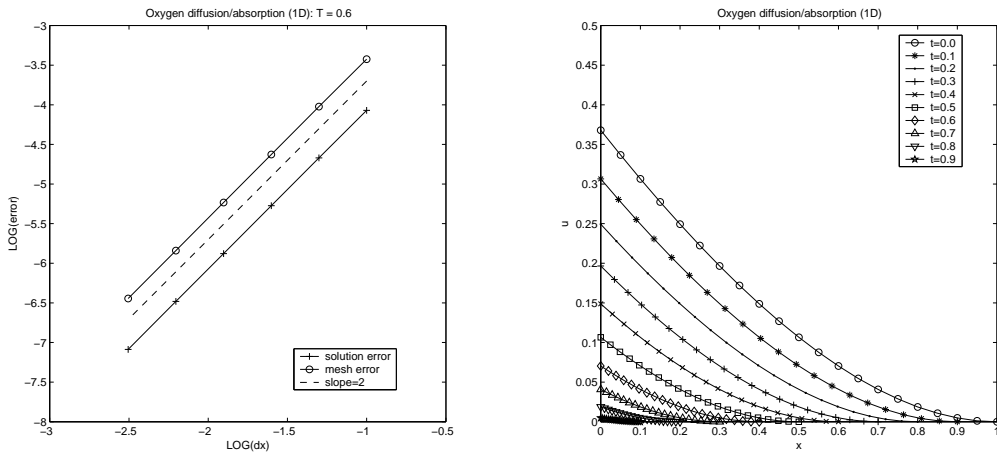


Fig. 10. Accuracy of the approximate solutions on a sequence of meshes at $T = 0.6$ (left) and the evolution of the exact and approximate solutions from $T = 0.0$ to $T = 0.9$ (right). The solution error shown is an L^1 approximation while the mesh error is an L^∞ approximation.

Figure 11 shows the evolution of the two-dimensional problem (for which no exact solution is available to the authors). The initial conditions are given by (55) at $t = 0$, with x replaced by r , the usual radial coordinate.

6 Conclusions

In this paper a moving mesh finite element numerical scheme has been described for the solution of moving boundary problems in one and two dimensions. The nodal velocities are controlled by maintaining the initial distribution

of local mass within the moving frame, which corresponds to scale invariance for the mass conserving applications described here. A range of numerical examples in one and two dimensions has been included in order to demonstrate empirically both the accuracy and the versatility of the proposed scheme.

The scale invariance and conservation properties of the porous medium equation have already been exploited on a local scale in [10], but this paper extends the idea to the invariance of a distributed conservation principle using linear finite elements. This allows a direct calculation of the nodal velocities in two dimensions, requiring only an assumption on their vorticity. Moreover, the solution is recovered without re-solving the differential equation. The extension of these ideas to the fourth order equation is also new. The application of the distributed conservation principle to the other applications shown is carried out for the first time. Although conservation is not present in these applications, it is shown here that a modified (or normalised) conservation principle can be constructed which allows the same methodology to be used. The advantages of using the conservation principle (13) in its non-distributed form are laid out in [10]. The choice of the distributed form is made in order to facilitate the application to higher dimensions. The fact that the method works without smoothing and with simple, forward Euler time-stepping means that the stiffness usually associated with moving mesh methods is well controlled.

The scheme is not restricted to one and two space dimensions since the theory in Section 2 holds for any number of dimensions and the implementation in Section 3 is readily generalised to moving tetrahedral meshes in three dimensions. Furthermore it is also possible to generalise the scheme to monitor functions other than the “mass” in Equations (12) and (13). These extensions will appear in a future paper.

From an analytical viewpoint, work is required to better understand and explain the observed orders of accuracy for the various test problems so far considered. Furthermore, both theoretical and practical extensions to higher order elements would make interesting and worthwhile investigations. Simi-

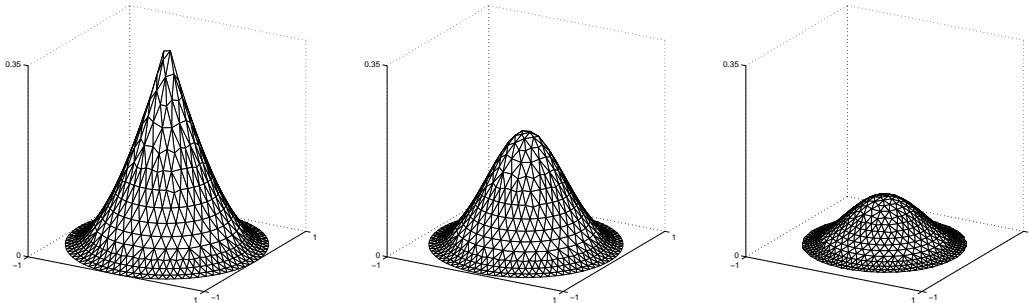


Fig. 11. Approximate solution surfaces at three different times for a two-dimensional problem with a negative source term. The results were obtained using a uniform but unstructured 615 node, 1149 cell mesh.

larly, a more complete understanding of the significance of an optimal mesh for the representation of the initial data would be desirable.

From a practical perspective, it would be useful to consider combining this moving mesh strategy with a discrete remeshing algorithm in order to improve the robustness of the software. This would ensure that the scheme may be used effectively without any *a priori* knowledge of how the solution is likely to evolve. It would also be possible to deal automatically with the type of self-intersection problem that is illustrated in Figure 7. Finally, work needs to be undertaken to assess the potential computational efficiency of the technique. As a prerequisite for this task an efficient implicit implementation of the solution of the ODE system (26) is required.

References

- [1] Almgren, R., Variational Algorithms and Pattern Formation in Dendritic Solidification, *J Comput Phys*, 106, 337-354 (1993).
- [2] Baines, M.J., *Moving Finite Elements*, OUP (1994).
- [3] Baines, M.J., Algorithms for Optimal Discontinuous Piecewise Linear and Constant L_2 Fits to Continuous Functions with Adjustable Nodes in One and Two Dimensions, *Math Comp*, 62, 645-669 (1994).
- [4] Beckett, G., Mackenzie, J.A. and Robertson, M.L., A Moving Mesh Finite Element Method for the Solution of Two-Dimensional Stefan Problems, *J Comput Phys*, 168, 500-518 (2001).
- [5] Berger, A.E., Ciment, M. and Rogers, J.C.W., Numerical Solution of a Diffusion Consumption Problem with a Free Boundary, *SIAM J Numer Anal*, 12, 646-672 (1975).
- [6] Blake, K.W., *Moving Mesh Methods for Nonlinear Partial Differential Equations*, PhD thesis, Dept of Mathematics, University of Reading, UK (2001).
- [7] Blake, K.W. and Baines, M.J., *Moving Mesh Methods for Nonlinear Partial Differential Equations*, Numerical Analysis Report 7/01, Dept of Mathematics, University of Reading, UK (2001)
- [8] Bonnerot, R. and Jamet, P., Numerical Computation of the Free Boundary for Two-Dimensional Stefan Problem by Space-Time Finite Elements, *J Comput Phys*, 25, 163-181 (1977).
- [9] Budd, C.J., Huang, W. and Russell, R.D., Moving Mesh Methods for Problems with Blow-up, *SIAM J Sci Stat Comput*, 17, 305-327 (1996).
- [10] Budd, C.J. and Piggott, M., The Geometric Integration of Scale-Invariant Ordinary and Partial Differential Equations. *J Comput Appl Math*, 128, 399-422 (2001).

- [11] Cao, W., Huang, W. and Russell, R.D., A Moving Mesh Method based on the Geometric Conservation Law, *SIAM J Sci Comp*, 24, 118-142 (2002).
- [12] Chen, S., Merriman, B., Osher, S. and Smereka, P., A Simple Level Set Method for Solving Stefan Problems, *J Comput Phys*, 135, 8-29 (1997).
- [13] Diez, J.A., Kondic, L. and Bertozzi, A., Global Models for Moving Contact Lines, *Phys Rev E*, 63, 011208 (2001).
- [14] Devine, K.D. and Flaherty, J.E., Parallel Adaptive hp-Refinement Techniques for Conservation Laws, *Appl Numer Math*, 20, 367-386 (1996).
- [15] Gelinias, R.J., Doss, S.K. and Miller, K., The Moving Finite Element Method: Application to General Partial Differential Equations with Multiple Large Gradients, *J Comput Phys*, 40, 202-249 (1981).
- [16] Harten, A. and Hyman, J.M., Self-Adjusting Grid Methods for One-Dimensional Hyperbolic Conservation Laws, *J Comput Phys*, 50, 235-269 (1983).
- [17] Huang, W., Ren, Y. and Russell, R.D., Moving Mesh Partial Differential Equations (MMPDEs) Based on the Equidistribution Principle, *SIAM J Numer Anal*, 31, 709-730 (1994).
- [18] Jimack, P.K. and Wathen, A.J., Temporal Derivatives in the Finite Element Method on Continuously Deforming Grids, *SIAM J Numer Anal*, 28, 990-1003 (1991).
- [19] Liao, G. and Anderson, D.A., A New Approach to Grid Generation, *Appl Anal*, 44, 285-411 (1992).
- [20] Liao, G., Liu, F., de la Pena, G., Peng, D. and Osher, S., Level-Set-Based Deformation Methods for Adaptive Grids, *J Comput Phys*, 159, 103-122 (2000).
- [21] Lynch, D.R., A Unified Approach to Simulation on Deforming Elements with Applications to Phase Change Problems, *J Comput Phys*, 47, 387-411 (1982).
- [22] Miller, K. and Miller, R.N., Moving Finite Elements, Part I, *SIAM J Numer Anal*, 18, 1019-1057 (1982).
- [23] Murray, J.D., *Mathematical Biology: An Introduction* (3rd edition), Springer (2002).
- [24] Peterson, R.C., Jimack, P.K. and Kelmanson, M.A., The Solution of Two-Dimensional Free-Surface Problems Using Automatic Mesh Generation, *Int J Numer Meth Fluids*, 31, 937-960 (1999).
- [25] Tourigny, Y. and Baines, M.J., Analysis of an Algorithm for Generating Locally Optimal Meshes for L-2 Approximation by Discontinuous Piecewise Polynomials *Math Comp*, 66, 623-650 (1997).
- [26] Thomas, P.D. and Lombard, C.K., The Geometric Conservation Law and its Application to Flow Computations on Moving Grids, *AIAA J*, 17, 1030-1037 (1979).