This is a repository copy of *Consistent dirichlet boundary conditions for numerical solution of moving boundary problems*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/8483/

Consistent Dirichlet Boundary Conditions for Numerical Solution of Moving Boundary Problems

M.E.Hubbard, M.J.Baines, P.K.Jimack

Abstract

We consider the imposition of Dirichlet boundary conditions in the finite element modelling of moving boundary problems in one and two dimensions for which the total mass is prescribed. A modification of the standard linear finite element test space allows the boundary conditions to be imposed strongly whilst simultaneously conserving a discrete mass. The validity of the technique is assessed for a specific moving mesh finite element method, although the approach is more general. Numerical comparisons are carried out for mass-conserving solutions of the porous medium equation with Dirichlet boundary conditions and for a moving boundary problem with a source term and time-varying mass.

1 Introduction

Moving boundary problems are very common in the mathematical modelling of physical processes [12]. Compared with fixed boundary problems they require extra boundary conditions to determine the motion of the boundary. For example, standard Dirichlet conditions for a fixed boundary problem are often supplemented by additional Neumann conditions. An appropriate discretisation for these problems is the finite element method with moving nodes, and many such techniques have been proposed in recent years [1, 3, 7, 8, 9, 10, 13, 15]. It should be noted however that the additional boundary conditions required to ensure well-posedness of the moving boundary problem introduce extra constraints on the treatment of the finite element equations at the boundary. In this paper we discuss the imposition of strong constraints (specifically, Dirichlet boundary conditions) on sets of moving mesh finite element equations with constraints on the total mass.

Solutions of these problems often exhibit global properties such as mass conservation, *i.e.* the preservation of an integral of the dependent variable in time. Such properties are a consequence of both the governing equation and the boundary conditions. The usual approach to the imposition of Dirichlet boundary conditions is to overwrite boundary values

1

and disregard the finite element equations arising from the test functions associated with those nodes. However, this step also destroys mass conservation in general, which may be a key property.

In this paper we discuss ways of preserving both Dirichlet conditions and conservation of mass by a modification of the test space. The procedure is described in the context of schemes which use a general ALE (Arbitrary Lagrangian-Eulerian) framework, after which a computational assessment is undertaken for one such scheme [1, 2]. In the remainder of this section we describe the class of moving boundary problems and boundary conditions considered here. A general form of the moving mesh finite element equations is then stated and the problem of simultaneously implementing Dirichlet boundary conditions and conservation is discussed. In the following section, ways of overcoming these problems are described, in both one and two space dimensions. The resulting algorithm is then tested on a selection of problems in one and two dimensions, using the moving mesh finite element method given in [1, 2]. We conclude with a short discussion.

1.1   Moving Boundary Problems and ALE Finite Elements

Consider a moving boundary problem of the form

$$u_t = \nabla \cdot \mathcal{F}u \qquad \text{for } \mathbf{x} \in \mathcal{R}(t), \tag{1}$$

where $\mathcal{F}$ is a spatial operator, with Dirichlet boundary conditions $u = g$ on the moving boundary $\partial\mathcal{R}(t)$. We shall mainly be concerned with problems for which the total mass,

$$\int_{\mathcal{R}(t)} u(t, \mathbf{x}) \, d\mathbf{x}, \tag{2}$$

remains constant in time: this class of problems includes a wide variety of nonlinear diffusion equations [1, 3, 6, 12]. However, as will be demonstrated, the ideas presented will apply to more general problems too.

By the Reynolds Transport Theorem [14] and (1),

$$
\begin{aligned}
\frac{d}{dt} \int_{\mathcal{R}(t)} u(t, \mathbf{x}) \, d\mathbf{x} &= \int_{\mathcal{R}(t)} u_t \, d\mathbf{x} + \oint_{\partial\mathcal{R}(t)} u \, \mathbf{v} \cdot \mathbf{n} \, dS \\
&= \int_{\mathcal{R}(t)} \nabla \cdot \mathcal{F}u \, d\mathbf{x} + \oint_{\partial\mathcal{R}(t)} u \, \mathbf{v} \cdot \mathbf{n} \, dS \\
&= \oint_{\partial\mathcal{R}(t)} (\mathcal{F}u + u \, \mathbf{v}) \cdot \mathbf{n} \, dS
\end{aligned}
\tag{3}
$$

where $\mathbf{v}(t)$ is the velocity of the boundary. Thus, the total mass (2) is conserved if and only if the boundary integral in (3) is zero. This, in turn, is true if u and $\mathbf{v}$ satisfy the boundary condition

$$(\mathcal{F}u + u \, \mathbf{v}) \cdot \mathbf{n} = 0. \tag{4}$$

Now consider a piecewise linear moving mesh finite element approach based upon a weak form of the Reynolds Transport Theorem. In order to do this let $\Omega(t)$ be a moving polygonal approximation to $\mathcal{R}(t)$ and assume a moving triangulation of $\Omega(t)$ with $N + B$ vertices, the first N being in the interior and the remainder being on the moving boundary. Let the test functions $W_i(t, \mathbf{x})$, $(i = 1, ..., N + B)$, be piecewise linear basis functions forming a partition of unity, and define piecewise linear approximations, $U \approx u$ and $\mathbf{V} \approx \mathbf{v}$, in the moving frame by

$$U(t, \mathbf{x}) = \sum_{j=1}^{N+B} U_j(t) W_j(t, \mathbf{x}), \qquad \mathbf{V}(t, \mathbf{x}) = \sum_{j=1}^{N+B} \mathbf{V}_j(t) W_j(t, \mathbf{x}) \tag{5}$$

(for further details see [1]). Then (*cf.* (3))

$$
\begin{aligned}
\frac{d}{dt} \int_{\Omega(t)} W_i(t, \mathbf{x}) \, U(t, \mathbf{x}) \, d\mathbf{x} &= \int_{\Omega(t)} \frac{\partial}{\partial t} (W_i \, U) \, d\mathbf{x} + \oint_{\partial\Omega(t)} W_i \, U \, \mathbf{V} \cdot \mathbf{n} \, dS \\
&= \int_{\Omega(t)} W_i \, \nabla \cdot (\mathcal{F}U) \, d\mathbf{x} + \int_{\Omega(t)} W_i \, \nabla \cdot (U \, \mathbf{V}) \, d\mathbf{x} \\
&= \int_{\Omega(t)} W_i \, \nabla \cdot (\mathcal{F}U + U \, \mathbf{V}) \, d\mathbf{x} \tag{6}
\end{aligned}
$$

for $i = 1, ..., N + B$, using (1) and the divergence theorem. Note that the second equality above uses the fact that the test function W_i moves with $\mathbf{V}$ (*i.e.* satisfies $\partial W_i/\partial t + \mathbf{V} \cdot \nabla W_i = 0$), as shown in [1] for example, and U is regularised as necessary to lie in the domain of $\mathcal{F}$.

Using integration by parts on (6), assuming that the discrete form of the boundary condition (4) holds on the boundary, the ALE finite element form is

$$\frac{d}{dt} \int_{\Omega(t)} W_i(t, \mathbf{x}) \, U(t, \mathbf{x}) \, d\mathbf{x} = -\int_{\Omega(t)} \nabla W_i \cdot (\mathcal{F}U + U \, \mathbf{V}) \, d\mathbf{x} . \tag{7}$$

Any $\mathbf{V}$ which satisfies the discrete form of (4) can be used in (7), leading via time-integration to the specification of

$$\int_{\Omega(t)} W_i(t, \mathbf{x}) \, U(t, \mathbf{x}) \, d\mathbf{x} \tag{8}$$

and subsequently to the evaluation of $U(t, \mathbf{x})$ (see below). It is the imposition of boundary conditions on this quantity that is the subject of this paper.

1.2 Mass Conservation

Note that the total discrete mass is given by

$$\int_{\Omega(t)} U(t, \mathbf{x}) \, d\mathbf{x} . \tag{9}$$

Furthermore, summing equations (7) over $i = 1, ..., N + B$ yields

$$\frac{d}{dt} \int_{\Omega(t)} U(t, \mathbf{x}) \, d\mathbf{x} = 0 \tag{10}$$

since the functions $W_i(t, \mathbf{x})$ form a partition of unity on $\Omega(t)$; that is $\sum_{i=1}^{N+B} W_i(t, \mathbf{x}) \equiv 1$ for all t. This implies that for any ALE finite element method of the form (7), which is constrained to satisfy the discrete form of (4) on the whole boundary, we obtain discrete mass conservation.

However, this argument holds only for weak imposition of Dirichlet boundary conditions on U. When Dirichlet boundary conditions are imposed strongly the test functions $W_i(t, \mathbf{x})$ no longer form a partition of unity, since (7) only holds for $i = 1, ..., N$, and so summation of these equations no longer yields conservation of the total discrete mass. In Section 2 below we describe a technique for modifying this test space which allows Dirichlet boundary conditions to be imposed strongly whilst simultaneously maintaining discrete conservation of mass.

In order to simplify the description of this technique we shall present it for one specific choice of ALE finite element method, that described in [1] (see below). However, the generalisation to any ALE solver is straightforward and is described in the Discussion at the end of this paper. For completeness a brief outline of the method of [1] is provided in Appendix A. The idea upon which it is based is to drive the mesh movement according to a *distributed* conservation principle

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

where each C_i is a constant in time, retaining its initial value at time $t = t_0$, given by

$$C_i \;=\; \int_{\Omega(t_0)} W_i(t_0, \mathbf{x}) \, U(t_0, \mathbf{x}) \, d\mathbf{x} \,, \tag{12}$$

thus specifying (8) directly.

Expanding $U(t, \mathbf{x})$ in terms of the $W_i(t, \mathbf{x})$, as in (5), Equation (11) leads to a mass matrix equation of the form

$$M \underline{U} \;=\; \underline{C} \,, \tag{13}$$

where M is the mass matrix, $\underline{U}$ is the unknown vector of coefficients in the expansion (5) and $\underline{C}$ is a known vector of the constants C_i. The symmetric positive definite mass matrix M has entries

$$M_{ij} \;=\; \int_{\Omega(t)} W_i(t, \mathbf{x}) \, W_j(t, \mathbf{x}) \, d\mathbf{x} \,. \tag{14}$$

As in the case of equations (7), summing (11) over $i = 1, ..., N + B$ (or equivalently summing the rows of (13)) gives

$$\sum_{i=1}^{N+B} \left[\int_{\Omega(t)} W_i(t, \mathbf{x}) \, U(t, \mathbf{x}) \, d\mathbf{x} \right] \;=\; \int_{\Omega(t)} U(t, \mathbf{x}) \, d\mathbf{x} \;=\; \int_{\Omega(t_0)} U(t_0, \mathbf{x}) \, d\mathbf{x} \,, \tag{15}$$

which is consistent with conservation of the total discrete mass (9). Also, as in the case of equation (7), when Dirichlet boundary conditions are imposed strongly on (13) in the usual way, by discarding those rows of the mass matrix (and the C_i) which correspond to the boundary test functions, mass conservation is destroyed.

2 Strong Boundary Conditions and Mass Conservation

As we have seen in the previous section, strong Dirichlet boundary conditions and mass conservation are, separately, easy to satisfy.

- To impose strong boundary conditions on (7), the relevant boundary values are over-written and the corresponding equations $(i = N + 1, ..., N + B)$ are removed. The remaining equations are solved for internal nodes only, although, as a result, mass conservation is destroyed.

- To impose mass conservation on (7) the full set of equations are solved for all nodes with no overwriting.

In the first situation, when one or more of the values of U_i on the boundary are constrained, the dimension of the approximation space is reduced, the associated equations are ignored (becoming "inactive") and the remaining "active" basis functions no longer form a partition of unity. Hence the total discrete mass is no longer conserved. Conservation of the integral of U can be restored, however, by modifying the *test* space so that the basis functions remain a partition of unity. This can be achieved by combining the inactive test functions with the active ones to form a modified test space.

For the remainder of this section we focus on the ALE method of [1], which solves each of the local conservation equations (11) in order to recover U at each point in time. When Dirichlet conditions are to be imposed, we now propose that each individual equation which corresponds to an inactive test function may be added to an active equation (or distributed over a subset of the active equations) prior to overwriting the constrained values. This technique conserves mass globally by associating the information carried by the boundary equations with internal nodes in a manner which preserves the integral of U (and retains an invertible system). The approach is therefore to solve the standard equations (11) but with modified test functions $\widetilde{W}_i$, *i.e.* to replace (11) at any given time by

$$\int_{\Omega(t)} \widetilde{W}_i(t, \mathbf{x}) \, U(t, \mathbf{x}) \, d\mathbf{x} \; = \; \widetilde{C}_i \tag{16}$$

say, where the $\widetilde{W}_i(t, \mathbf{x})$ are chosen to enforce both the Dirichlet and the mass conservation conditions, and the $\widetilde{C}_i$ are new constants, evaluated at $t = t_0$,

$$\widetilde{C}_i \; = \; \int_{\Omega(t_0)} \widetilde{W}_i(t_0, \mathbf{x}) \, U(t_0, \mathbf{x}) \, d\mathbf{x} \,. \tag{17}$$

Consequently, the test and trial spaces differ, as in the Petrov-Galerkin approach.

2.1 One Dimension

In one dimension we use the more natural notation that the boundary nodes are numbered 0 and $N + 1$ (with the interior nodes still numbered from 1 to N). In this case, the twin

Figure 1: Test functions for the recovery of U via mass conservation in one dimension which are compatible with strong imposition of Dirichlet boundary conditions.

goals of Dirichlet conditions and mass conservation are easily achieved by taking

$$\widetilde{W}_1(t,x) \;=\; W_0(t,x) + W_1(t,x) \qquad \text{and} \qquad \widetilde{W}_N(t,x) \;=\; W_N(t,x) + W_{N+1}(t,x), \quad (18)$$

leaving the remaining $W_i(t,x)$ unaltered. The family of modified piecewise linear test functions is illustrated in Figure 1 and leads to the set of equations

$$\int_{X_0(t)}^{X_{N+1}(t)} (W_0 + W_1)\, U \; dx \;=\; \widetilde{C}_1 \;=\; C_0 + C_1$$

$$\int_{X_0(t)}^{X_{N+1}(t)} W_i\, U \; dx \;=\; \widetilde{C}_i \;=\; C_i \qquad \text{for} \quad i = 2, \ldots, N-1$$

$$\int_{X_0(t)}^{X_{N+1}(t)} (W_N + W_{N+1})\, U \; dx \;=\; \widetilde{C}_{N+1} \;=\; C_N + C_{N+1}. \qquad (19)$$

The trial space is based upon the usual piecewise linear basis, with the coefficients of W_0 and W_{N+1} overwritten by the Dirichlet boundary conditions. Adding together equations (19) shows that this approach retains global mass conservation, albeit with modified local masses that have the same total mass. Importantly, the additional terms make the mass matrix more diagonally dominant.

2.2   Two Dimensions

In two dimensions the test functions can be similarly modified, in such a way that they retain the same support as the original hat functions but with the test functions adjacent to the boundary distributed to interior test functions. The mapping is non-unique and two possible choices will be described here.

i. The first approach is derived by decomposing the mass associated with any test function corresponding to a boundary node into components obtained by integrating the overlapping trial functions over each of the neighbouring cells. For example, consider a node, I say, on the boundary of the polygonal domain $\Omega(t)$. According to the connectivity shown in Figure
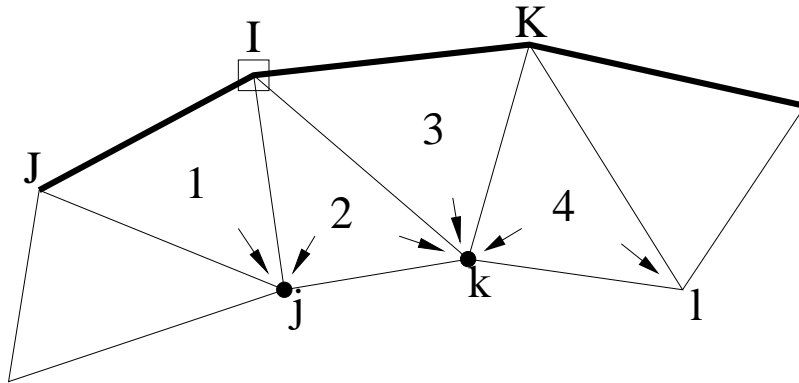
6

Figure 2: Boundary nodes for two-dimensional boundary conditions. The thicker line represents the boundary edges.

2, the mass associated with node I is given by

$$
\begin{aligned}
\int_{\Omega(t)} W_I(t, \mathbf{x})\, U(t, \mathbf{x})\, d\mathbf{x} \;=\;& \int_{\triangle_1} W_I \left(W_I\, U_I + W_J\, U_J + W_j\, U_j \right) d\mathbf{x} \\
+\;& \int_{\triangle_2} W_I \left(W_I\, U_I + W_j\, U_j + W_k\, U_k \right) d\mathbf{x} \\
+\;& \int_{\triangle_3} W_I \left(W_I\, U_I + W_k\, U_k + W_K\, U_K \right) d\mathbf{x} .
\end{aligned} \tag{20}
$$

The upper case subscripts indicate boundary nodes while lower case subscripts signify internal nodes. This is more complicated than the one-dimensional case because many of the boundary nodes have more than one adjacent node to which their contributions can be transferred. Furthermore, even though the values of $U(t, \mathbf{x})$ on the boundary remain constant during the time evolution, their contribution to the integral in (20) does not (because the nodal positions in the moving mesh method may change), so they cannot be ignored (this is also true in one dimension).

The boundary cells can clearly be split into two categories, one of which is far simpler to deal with.

- When a triangular cell has two vertices on the boundary and one in the interior, the full mass of the cell can be transferred to the internal node.

- When a triangular cell has one vertex on the boundary and two in the interior, the mass of the cell can be associated with the two internal nodes. However, it is not obvious how the mass of the cell should be distributed between the two: possibilities include splitting it equally, or weighting the split in proportion to the associated nodal values of the dependent variable. The latter option is discussed below.

The resulting split is more clearly illustrated by the annotation of the terms in (20) given by

$$\int_{\Omega(t)} W_I(t,\mathbf{x})\, U(t,\mathbf{x})\, d\mathbf{x} \;=\; \underbrace{\int_{\triangle_1} W_I\,(W_I\,U_I + W_J\,U_J + W_j\,U_j)\, d\mathbf{x} \;+\; \int_{\triangle_2} W_I\,(W_j\,U_j)\, d\mathbf{x}}_{\text{add to equation } j}$$

$$+\; \underbrace{\int_{\triangle_3} W_I\,(W_I\,U_I + W_k\,U_k + W_K\,U_K)\, d\mathbf{x} \;+\; \int_{\triangle_2} W_I\,(W_k\,U_k)\, d\mathbf{x}}_{\text{add to equation } k}$$

$$+\; \underbrace{\int_{\triangle_2} W_I\,(W_I\,U_I)\, d\mathbf{x}}_{\text{split between } j \text{ and } k}\;. \tag{21}$$

The right-hand side constants C_i in Equation (11) are calculated from a piecewise linear representation of the dependent variable, and can therefore be partitioned in the same way as in (21). The test functions indicated by (21), examples of which are provided in Figure 3, are clearly discontinuous, but this has no adverse effects on the computation, since no derivatives are required. Figure 3 also shows that the test functions for nodes which are not adjacent to the boundary remain unchanged.

In algebraic terms, the test function associated with node k in Figure 2 would take the form

$$\widetilde{W}_k \;=\; W_k + W_I|_{\triangle_3} + W_K|_{\triangle_3} + \frac{\lambda_{2,k}}{\lambda_{2,j} + \lambda_{2,k}}\, W_I|_{\triangle_2} + \frac{\lambda_{4,k}}{\lambda_{4,k} + \lambda_{4,l}}\, W_K|_{\triangle_4}\;, \tag{22}$$

in which $W|_{\triangle}$ indicates the test function restricted to the specified triangular mesh cell, and the λ are chosen appropriately. Since the support for each of the modified test functions is the same as that of the corresponding standard linear test function, this will be referred to as the *compact modified approach*. Furthermore, this method adds to the diagonal of the mass matrix, making it more diagonally dominant and potentially easier to invert.

The special case in which *zero* Dirichlet boundary conditions are applied is considerably simpler. In this situation, the mass associated with node I, according to the connectivity shown in Figure 2, is given by

$$\int_{\Omega(t)} W_I(t,\mathbf{x})\, U(t,\mathbf{x})\, d\mathbf{x}$$

$$=\; \int_{\triangle_1} W_I\,U\, d\mathbf{x} \;+\; \int_{\triangle_2} W_I\,U\, d\mathbf{x} \;+\; \int_{\triangle_3} W_I\,U\, d\mathbf{x}$$

$$=\; \int_{\triangle_1} W_I\,(W_j\,U_j)\, d\mathbf{x} \;+\; \int_{\triangle_2} W_I\,(W_j\,U_j + W_k\,U_k)\, d\mathbf{x} \;+\; \int_{\triangle_3} W_I\,(W_k\,U_k)\, d\mathbf{x}$$

$$=\; \underbrace{\int_{\Omega(t)} W_I\,(W_j\,U_j)\, d\mathbf{x}}_{\text{add to equation } j} \;+\; \underbrace{\int_{\Omega(t)} W_I\,(W_k\,U_k)\, d\mathbf{x}}_{\text{add to equation } k}\;. \tag{23}$$

As suggested by the final line of (23), the contributions can be used to augment the equations associated with the nodes adjacent to the boundary in a manner analogous to that carried
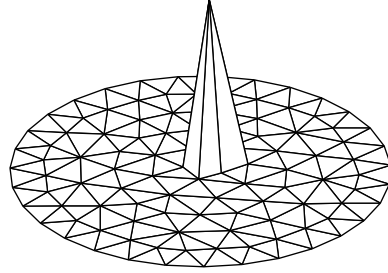
8
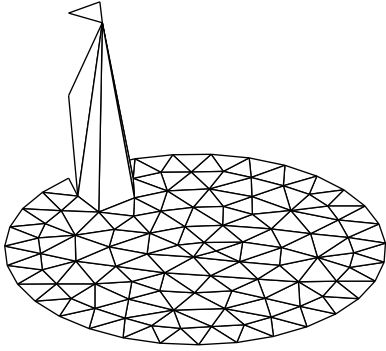
Figure 3: Representative test functions for the recovery of U via mass conservation in two dimensions which are compatible with strong imposition of Dirichlet boundary conditions using the "compact" approach.

out for one dimension in (19). The boundary values may then be overwritten with zero Dirichlet conditions and the internal equations solved for the internal values of U without losing mass. A similar approach can be followed however many internal nodes are adjacent to I. (In the special case where node I has no adjacent internal nodes, the boundary condition $U = 0$ ensures that node I does not contribute to the overall mass and can therefore be ignored.) Exactly the same partition can be applied to (12) to obtain (17).

The distribution of the mass indicated by (23) can be achieved using modified test functions of the form (22) in which $\lambda = U$. Since

$$\int_{\triangle_2} W_I W_k \, d\Omega \;=\; \int_{\triangle_2} W_I W_j \, d\Omega \tag{24}$$

it is simple to show that

$$\int_{\triangle_2} \frac{U_k}{U_j + U_k} W_I U \, d\Omega \;=\; \int_{\triangle_2} W_I \left(W_k U_k \right) d\Omega \,, \tag{25}$$

with a similar equality for U_j, from which the equivalence of the schemes based on (22) and (23) follows immediately. In the presence of non-homogeneous Dirichlet boundary conditions, (25) no longer holds, and using a test function of the form (22) in (16) would seem to require the inversion of a nonlinear system of equations to recover the nodal values of U. However, this can be easily avoided by lagging the values of U, using the values from the previous iteration in the definition of the test function (22). In this paper the effect of the modification will be illustrated through its application to problems in which homogeneous Dirichlet boundary conditions are imposed on the moving boundary, but the same approach has been used successfully in the modelling of systems with non-homogeneous boundary conditions.
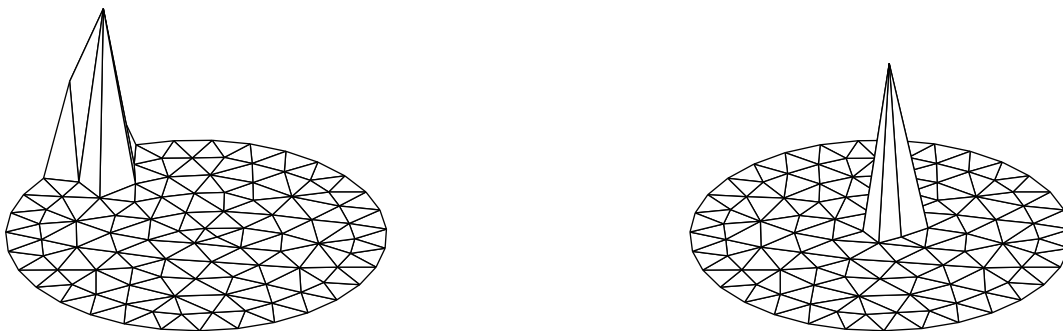
Figure 4: Representative test functions for the recovery of U via mass conservation in two dimensions which are compatible with strong imposition of Dirichlet boundary conditions using the "averaged" approach.

ii. The second approach considered here modifies the test functions directly simply by redistributing the test functions associated with the boundary nodes equally between the adjacent internal nodes to give, in the case of node k in Figure 2,

$$\widetilde{W}_k \;=\; W_k + \frac{1}{N_I}W_I + \frac{1}{N_K}W_K\,, \tag{26}$$

in which N_I and N_K represent the number of internal nodes connected by the mesh to boundary nodes I and K, respectively. This scheme will be referred to as the *averaged modified approach.* The logic required to implement this form is simpler than for the first, and the resulting system given by (16) is always linear, but the modification is not guaranteed to increase the diagonal dominance of the mass matrix and it can extend the stencil of the scheme, especially in three dimensions (and hence reduce the sparsity of the mass matrix where a boundary node has more than d internal neighbours, d being the number of space dimensions).

3   Numerical Experiments

In this section we compare the results obtained from the particular ALE method [1], applied to two different moving boundary problems, in the following cases:

A Dirichlet conditions are imposed weakly and mass is conserved exactly (as in [1]),

B Dirichlet conditions are imposed strongly and mass is conserved exactly using the modified test functions (as described in the previous section). In two dimensions, the two different forms described in Section 2.2 are used:

    i the compact modified approach;

ii the averaged modified approach.

3.1   The Porous Medium Equation

We first present numerical results for a problem of the form (1) which satisfies the boundary condition (4) on its whole boundary. The porous medium equation (PME) takes the form

$$\frac{\partial u}{\partial t} \;=\; \nabla \cdot (u^n \nabla u), \tag{27}$$

where $n > 0$ is an integer exponent, on a moving domain $\mathcal{R}(t)$ with the Dirichlet boundary condition $u = 0$ on $\partial \mathcal{R}(t)$. It is well known that solutions exist with finite but time-dependent support in which mass (the integral of the dependent variable u over the whole domain) is conserved. In particular, equation (27) admits a family of compact-support similarity solutions with moving boundaries at which $u = 0$ (see, for example, [4, 11]). Two of these solutions are considered here, one with exponent $n = 1$ (for which the slope of the self-similar solution normal to the moving boundary is finite) and the other with $n = 3$ (for which the slope normal to 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

$$u(r,t) \;=\; \begin{cases} \frac{1}{\lambda^d(t)} \left(1 - \left(\frac{r}{r_0 \lambda(t)} \right)^2 \right)^{\frac{1}{n}} & |r| \le r_0 \lambda(t) \\[2mm] 0 & |r| > r_0 \lambda(t) , \end{cases} \tag{28}$$

in which r is the usual radial coordinate, and where

$$\lambda(t) \;=\; \left(\frac{t}{t_0} \right)^{\frac{1}{2+dn}} \qquad \text{and} \qquad t_0 \;=\; \frac{r_0^2 n}{2(2 + dn)} . \tag{29}$$

Symmetry is not assumed in the numerical calculations carried out here, so in one dimension, for example, the whole of the solution support $[-r, r]$ is modelled. 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 computed for four test cases, two each in one and two space dimensions.

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

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

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

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

All the results have been obtained on quasi-uniform meshes, for which the representative *initial* mesh size is repeatedly halved to estimate orders of accuracy. The time-step (using Heun's scheme, as discussed in Appendix A) is divided by four with each mesh refinement.
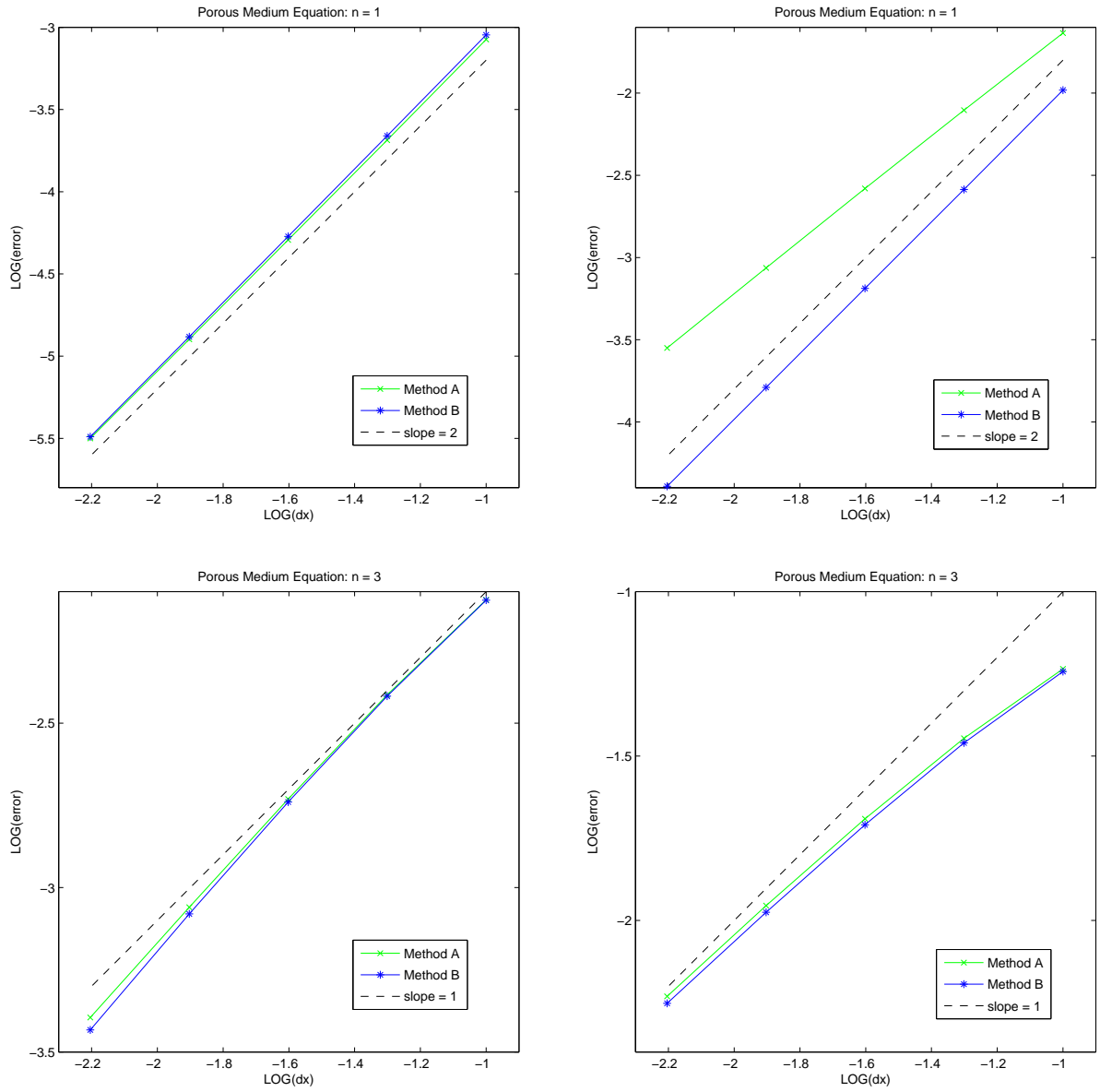
11

Figure 5: Comparison of L^2 errors in the solution (left) and the magnitudes of the errors in the boundary node positions (right) for the different boundary approximations in one space dimension.
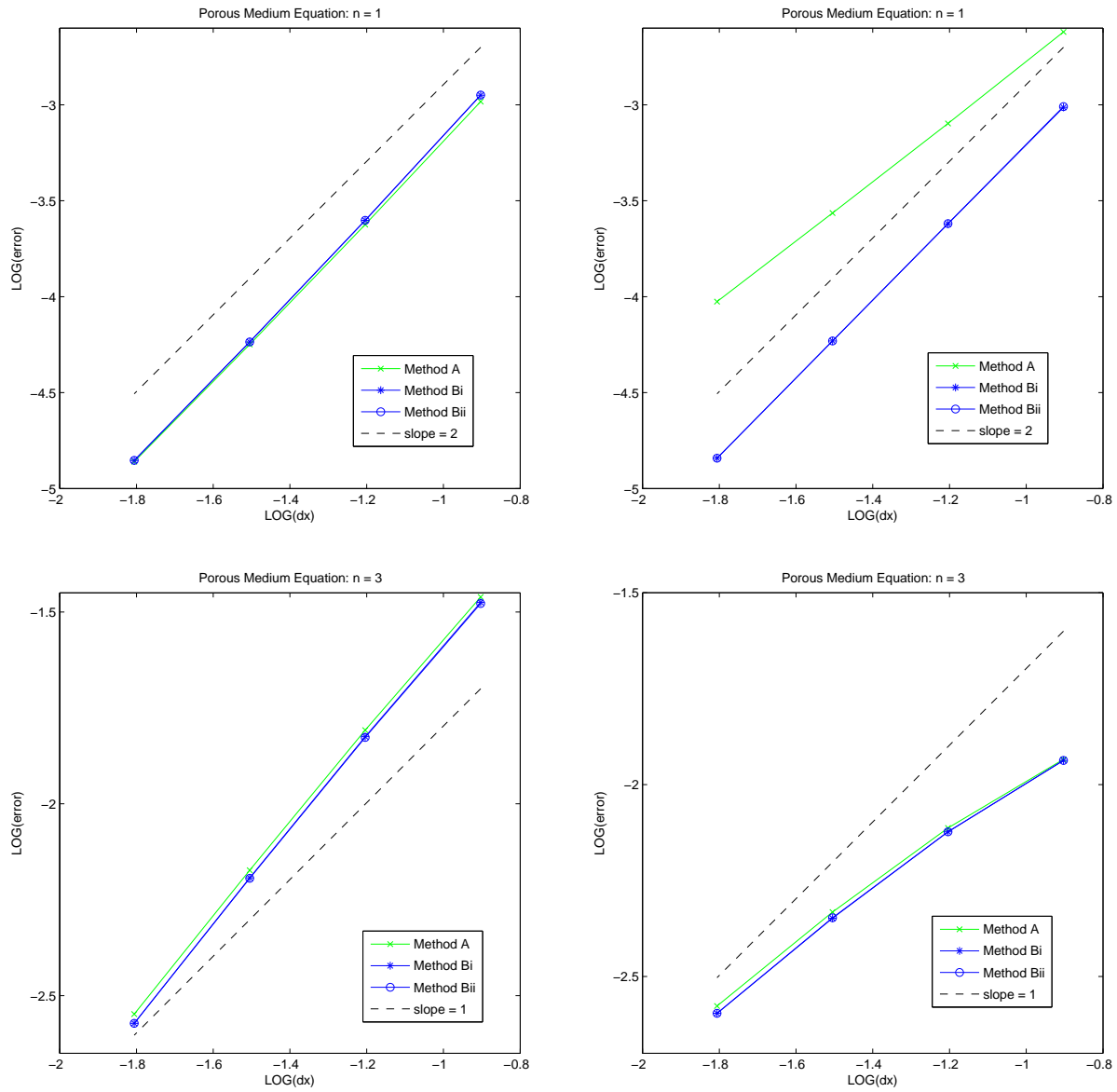
Figure 6: Comparison of L^2 errors in the solution (left) and boundary node positions (right) for the different boundary approximations in two space dimensions.

Table 1: Numerical estimates of the orders of accuracy of the scheme for the dependent variable U and the boundary positions X_b or R_b 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	X_b	U	X_b	Method	U	R_b	U	R_b
A	2.00	1.62	1.11	0.92	**A**	2.03	1.53	1.25	0.81
B	2.02	2.00	1.17	0.92	**Bi**	2.05	2.03	1.26	0.82
					Bii	2.05	2.03	1.26	0.83

The values of various error measures are shown in Figures 5 and 6, while numerically estimated orders of accuracy are given in Table 1.

These results lead to a number of observations.

- The results obtained using methods **Bi** and **Bii** in the two-dimensional cases are almost indistinguishable in terms of speed and accuracy. In the discussion below they will be considered together and referred to simply as method **B**.

- There can be significant differences between the errors in approximating the boundary position in the two methods, 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 or slightly better when $n=3$, whichever approach is used.

- Overall, the least accurate of the approaches seems to be the one which only imposes $U=0$ weakly (method **A**). This is most noticeable when considering the error in the boundary position, where for $n=1$ the numerical evidence suggests that the order of accuracy drops by about 0.5.

- In terms of computational speed, the methods are very similar, though method **B** is typically slightly faster, partly because it is inverting a smaller system when recovering the values of the dependent variable, and partly because the adjustments made to the mass matrix tend to improve its conditioning and so accelerate the convergence of the conjugate gradient algorithm used to invert the mass matrix.

- Neither method constrains the dependent variable from becoming negative, but the nature of the solutions tested has meant that this has only happened when the Dirichlet boundary condition is enforced weakly (method **A**). The negative values (when they appear) are only small but there may be practical circumstances in which even this would be unacceptable.

- The solution was initialised with the piecewise linear interpolant of the exact solution values at the nodes of the initial mesh. The same experiments have been run using a constrained best least squares fit to provide the initial values for the dependent variable. Similar orders of accuracy were seen when using method **B**, but typically the least squares fit gave better solutions for $n = 3$, whilst the exact nodal values were better for $n = 1$.

Of the two general approaches, method **B**, which allows exact mass conservation and Dirichlet boundary conditions to be achieved simultaneously, clearly has the better combination of accuracy, robustness, speed and physical realism, when applied in the context of this moving mesh method. Hence this approach is recommended for future use. Since there is little to distinguish between methods **Bi** and **Bii** in terms of speed and accuracy, either would be an appropriate choice in two-or three-dimensional simulations, although **Bii** has the attraction of greater simplicity, even though the modified test functions may have a greater support.

3.2   A Diffusion-Absorption Problem

In addition to examples of the form (1), or for which the total mass is constant in time, we illustrate here that the technique proposed may also be applied to more general problems for which both mass conservation and Dirichlet boundary conditions are still important. This test case represents a moving boundary problem with a source term, described in [5], which models the diffusion of oxygen in an absorbing medium. The equation takes the form

$$\frac{\partial u}{\partial t} = \nabla^2 u - 1 \qquad \text{with} \qquad u|_{\partial\Omega(t)} = \frac{\partial u}{\partial n}\bigg|_{\partial\Omega(t)} = 0, \qquad (30)$$

in which $\partial\Omega(t)$ represents the moving boundary only. Details of how the moving mesh finite element approach can be modified to approximate this system can be found in [1].

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

$$u(x,t) = \begin{cases} -x - t + e^{x+t-1} & x \leq 1 - t \\ 0 & x > 1 - t, \end{cases} \qquad (31)$$

with initial conditions at $t_0 = 0$. In this one-dimensional case an additional boundary condition is also required at the fixed boundary $x = 0$, and here the exact value of $u_x(0,t)$ (from (31)) is imposed, as in [1]. Note that it is now the *proportion* of the total mass associated with each node which is held constant by the algorithm. The rate of change of mass for this problem is known, and can be used to modify the right-hand side of (11) and (16) appropriately.

Figure 7 shows that second order accuracy is again achieved for this problem with the new approach to modelling the boundary conditions. At the mesh resolutions shown the non-conservative approach gives more accurate solutions but the order of accuracy is slightly lower: on the finest meshes tested the numerical estimates of the orders of accuracy are

- 1.92 for method **A** compared with 1.99 for method **B** for the dependent variable u.

- 1.98 for method **A** compared with 2.00 for method **B** for the moving boundary position x_{N+1}.

Figure 8 shows the evolution of the corresponding two-dimensional problem (for which no exact solution is available to the authors) obtained using method **Bi**. The initial conditions are given by (31) at $t = 0$, with x replaced by r, the usual radial coordinate.
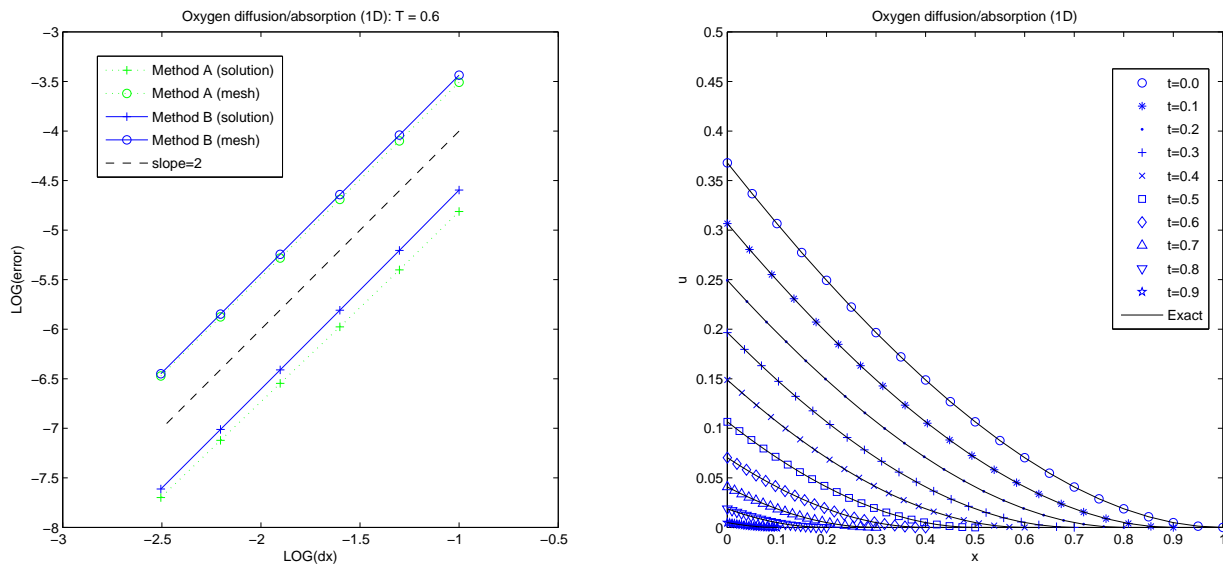

Figure 7: Accuracy of the approximate solutions on a sequence of meshes at $T = 0.6$ (left) and the evolution of the exact and approximate solutions from $T = 0.0$ to $T = 0.9$ (right). The solution error shown is an L^2 approximation while the mesh error is simply the magnitude of the error in the boundary position.

4   Discussion

In this paper we have discussed the strong imposition of Dirichlet boundary conditions at the same time as conserving mass in the finite element modelling of moving boundary problems. It has been demonstrated that both constraints can be made to hold simultaneously if the test space is modified in a relatively simple way. In one dimension the modification is immediate, in two dimensions slightly more complicated, but can be extended simply to three dimensions.

The effect of imposing both conditions, as opposed to using a standard linear test space and imposing the boundary conditions weakly, has been assessed for a moving mesh finite element method based on the distributed Lagrangian conservation principle (11), as applied to the porous medium equation in one and two space dimensions. The results suggest

16

that an approach which imposes both mass conservation and boundary conditions strongly is superior. The improved conditioning of the mass matrix also makes the method more efficient and this approach is recommended. This recommendation includes problems for which the total mass is not constant since, for the problem defined in (30), in which the mass varies in a known manner due to the influence of a source term, the new approach improved the order of accuracy of the approximation.

As described in Section 1, the problem of imposing both strong Dirichlet boundary conditions and mass conservation arises for any ALE formulation which correctly captures the motion of the moving boundary (see Equation (4)). In particular, when Dirichlet conditions are imposed in the usual finite element manner, so that equations (7) are solved only for $i = 1, ..., N$, mass is not conserved. The technique proposed here, in which the test functions are modified as described in Section 2, may be applied in any such situation. Specifically, if $\mathbf{V}(t)$ is known, and satisfies the discrete form of the boundary condition (4), the right-hand side of (6) or (7) can be evaluated and used to give the values of (8) at the new time level, as described in [3] for example. Modification of the test functions as illustrated by (22) or (26), can then be applied to allow simultaneous application of boundary conditions and mass conservation. In the case where $\lambda = U$ is used in (22) a similar issue arises as when non-homogeneous Dirichlet boundary conditions are considered, so the nodal values of U must be lagged in the definition of the test function.

In Section 2.2 the modification was considered in terms of redistributing the mass associated with each boundary node to internal nodes. Viewed in this context, the particular moving mesh finite element method used above calculates the constants C_i in Equation (11) from an initial piecewise linear representation of the dependent variable, so they could be distributed in the manner described by Equation (21). Now, when considering the mesh connectivity shown in Figure 2, the updated values of the quantities on the right-hand side
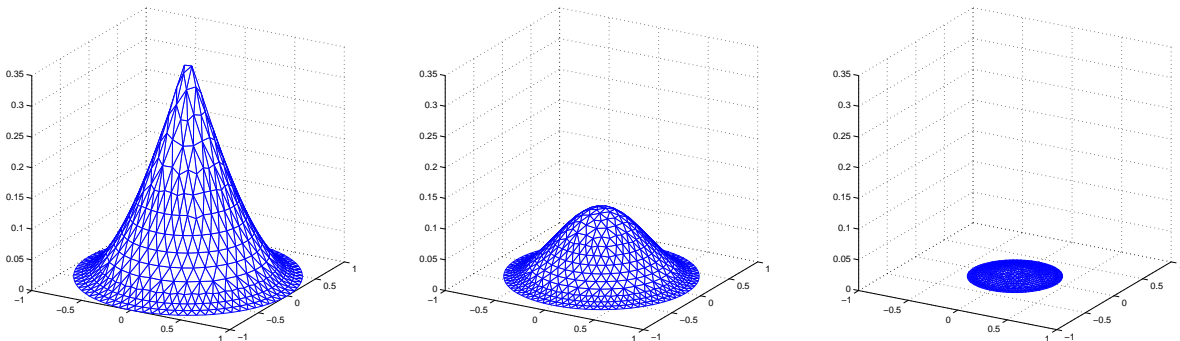

Figure 8: Approximate solution surfaces at three different times ($t = 0.0$, 0.05 and 0.1) for a two-dimensional problem with a negative source term. The results were obtained using a uniform but unstructured 615 node, 1149 cell mesh.

of (6) can be broken up no further than

$$\int_{\Omega(t)} W_I f \, d\mathbf{x} \;=\; \int_{\triangle_1} W_I f \, d\mathbf{x} \;+\; \int_{\triangle_2} W_I f \, d\mathbf{x} \;+\; \int_{\triangle_3} W_I f \, d\mathbf{x}. \tag{32}$$

Here the precise definition of f depends upon the time integration scheme used to solve (6) to obtain (8) at the new time level, and U is regularised as necessary. As before, the contributions from triangles 1 and 3 can, in each case, be transferred to the unique internal vertex (as can also be done in one dimension). The contribution from triangle 2 can be split and, for consistency with the process illustrated by Equation (23), should be carried out as follows (*cf.* Equation (22)):

$$\int_{\triangle_2} W_I f \, d\mathbf{x} \;=\; \underbrace{\frac{f_j}{f_j + f_k} \int_{\triangle_2} W_I f \, d\mathbf{x}}_{\text{add to equation } j} \;+\; \underbrace{\frac{f_k}{f_j + f_k} \int_{\triangle_2} W_I f \, d\mathbf{x}}_{\text{add to equation } k}. \tag{33}$$

The solution U then preserves the integral of f and therefore the total mass.

Interestingly, the simple idea presented in this paper may also be applied in a range of different contexts. For example, it is applicable when mass conservation is required alongside any specific constraint on the value of u in equations such as (1), not only constraints on the boundary of the domain. A similar situation arises in finding a constrained L^2 best fit to a fixed function in any number of space dimensions when, for example, mass conservation is required.

5   Acknowledgements

References

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

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

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

[4] G.I.Barenblatt, Scaling, CUP, 2003.

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

[6] C.J.Budd, G.J.Collins, W.Huang and R.D.Russell, Self-similar numerical solutions of the porous-medium equation using moving mesh methods, *Phil. Trans. Roy. Soc. London*, **357**:1047–1077, 1999.

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

[8] W.Huang, Y.Ren and R.D.Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.*, **113**:279–290, 1994.

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

[10] J.A.Mackenzie and M.L.Robertson, A moving mesh method for the solution of the one-dimensional phase-field equations, *J. Comput. Phys.*, **181**(2):526–544, 2002.

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

[12] J. Ockendon, S. Howison, A Lacey and A Movchan, Applied Partial Differential Equations (revised edition), Oxford University Press, 2003.

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

[14] P.Wesseling, Computational Fluid Dynamics, Springer, 2000.

[15] P.Zegeling, Moving Grid Techniques, in *Handbook of Grid Generation*, J.F.Thompson, B.K.Soni, N.P.Weatherill (Eds.), CRC Press, 37-1 – 37-18, 1999.

A   Derivation of the Nodal Velocities

The mesh velocities for the moving mesh finite element method of [1] are derived by differentiating the integral in (11) with respect to time. Assuming the existence of a velocity potential, and integrating by parts, this leads to an equation for a piecewise linear approximation, Φ, to the velocity potential which, with zero Dirichlet boundary conditions for U and $\mathcal{F}U$ (for simplicity of this exposition), is

$$\int_{\Omega(t)} U(t, \mathbf{x}) \, \nabla \Phi \cdot \nabla W_i \, d\mathbf{x} \; = \; - \int_{\Omega(t)} \nabla W_i \cdot \mathcal{F}U \, d\mathbf{x} \, . \tag{34}$$

The velocity $\mathbf{V}$ is obtained by a best L^2 fit to $\nabla\Phi$,

$$\int_{\Omega(t)} W_i(t, \mathbf{x})\, \mathbf{V}(t, \mathbf{x})\, d\mathbf{x} \;=\; \int_{\Omega(t)} W_i(t, \mathbf{x})\, \nabla\Phi\, d\mathbf{x}\,. \tag{35}$$

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

1. Given a set of mesh nodes with positions $\mathbf{X}_i$ solve (11) to obtain the solution values U_i at the mesh nodes, $i = 1, \ldots, N + B$. The constants C_i are calculated from the initial solution and mesh. This is the stage where the techniques of this paper may be applied.

2. Use the values of U_i in (34) to obtain the velocity potentials Φ_i at the mesh nodes $i = 1, \ldots, N + B$. In order to ensure that (34) has a unique solution, $\Phi = 0$ is fixed at one point in the computational domain, with the remaining equations solved for every other node.

3. Use the values of Φ_i in (35) to obtain the mesh node velocities $\mathbf{V}_i$ for $i = 1, \ldots, N + B$, without imposing boundary conditions on $\mathbf{V}$.

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

$$\frac{d\vec{\mathbf{X}}}{dt} \;=\; \vec{F}(\vec{\mathbf{X}}) \;=\; \vec{\mathbf{V}}\,. \tag{36}$$

This system of ordinary differential equations in time can then be integrated using any appropriate method. In [1], a simple forward Euler discretisation was used, but in this paper Heun's scheme, a second order TVD Runge-Kutta discretisation, is used to ensure that the errors due to approximating the time derivative are negligible compared to those incurred by the spatial discretisation. The qualitative features of the results presented in this paper are not changed by switching to the first order scheme, but the quantitative differences can be assessed using the comparison presented in Figures 9, 10 and 11. It is clear from the figures that the second order scheme is accurate enough to ensure that the errors due to the temporal discretisation are negligible for almost any value of the time-step which avoids mesh tangling. A fourth order Runge-Kutta scheme was also used, but the additional accuracy was unnecessary for this work.

The solid circles indicate the sets of results presented in Figures 5, 6 and 7. It is worth noting that in some cases the forward Euler scheme produces results which are more accurate than those of the Runge-Kutta scheme at this mesh resolution. This is due to cancellation between the errors in the discretisation of the temporal and spatial derivatives and it can be seen (where stability allows) that the overall errors for the first order approach do become significantly larger than those obtained using the second order scheme once the time-step is large enough for the temporal errors to dominate the spatial errors.
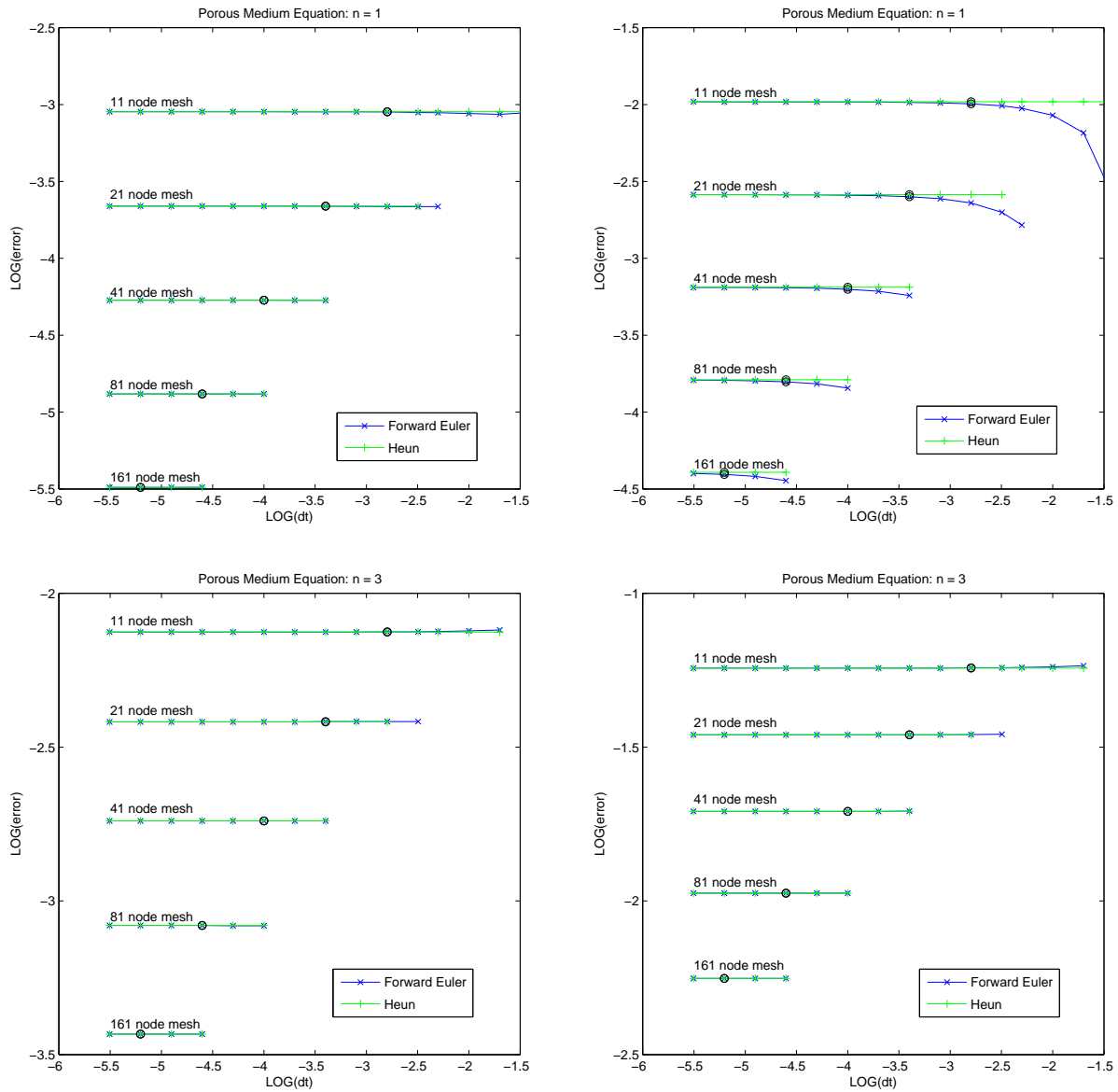
Figure 9: Comparison of L^2 errors in the solution (left) and the magnitudes of the errors in the boundary node positions (right) for the different time derivative approximations used with the one-dimensional porous medium equation. For each mesh, increasing the value of the time-step by a factor of 2 from that used to obtain the rightmost point leads to mesh tangling.
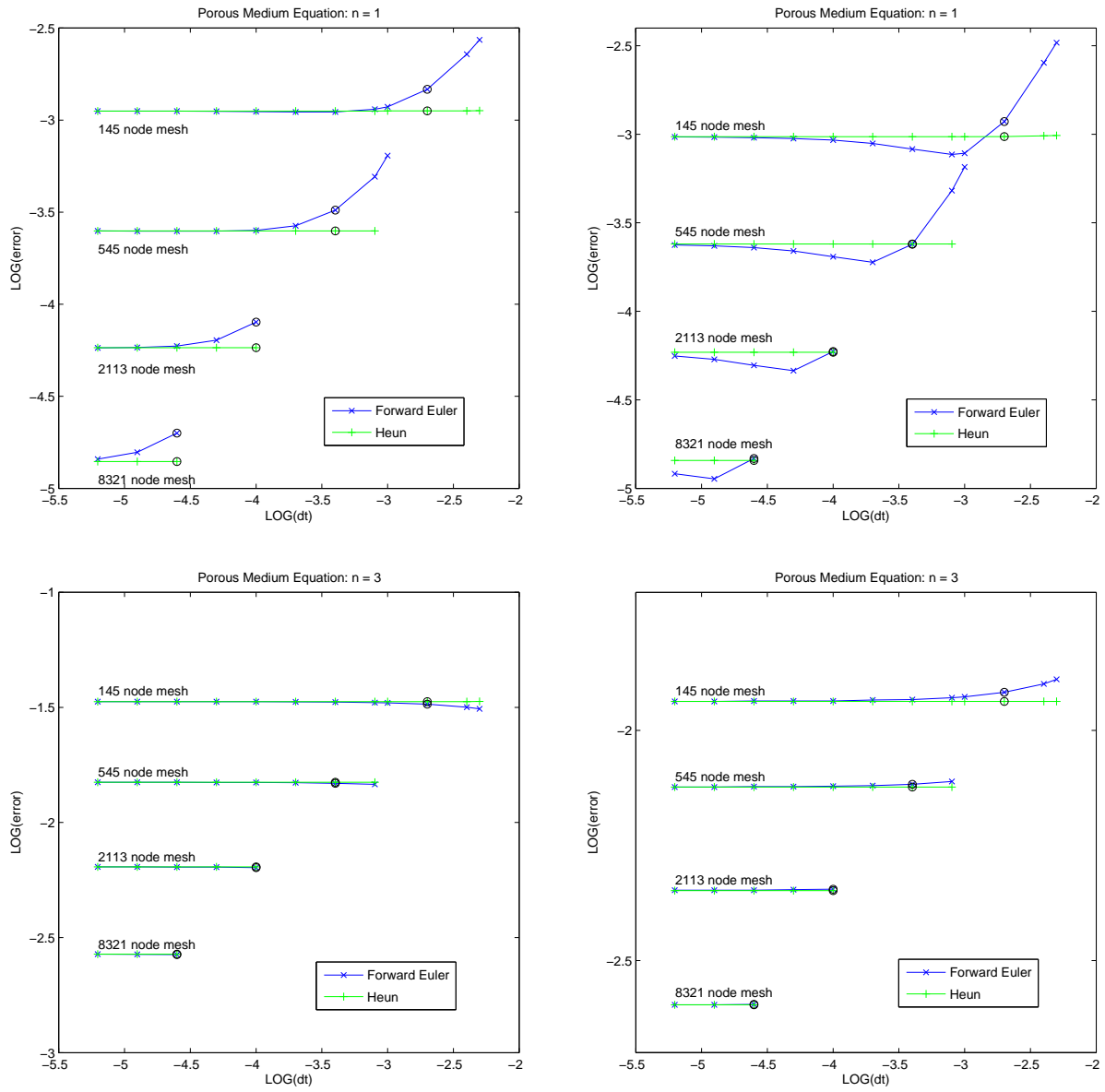
Figure 10: Comparison of L^2 errors in the solution (left) and boundary node positions (right) for the different time derivative approximations used with the two-dimensional porous medium equation. For each mesh, increasing the value of the time-step by a factor of 2 from that used to obtain the rightmost point leads to mesh tangling.
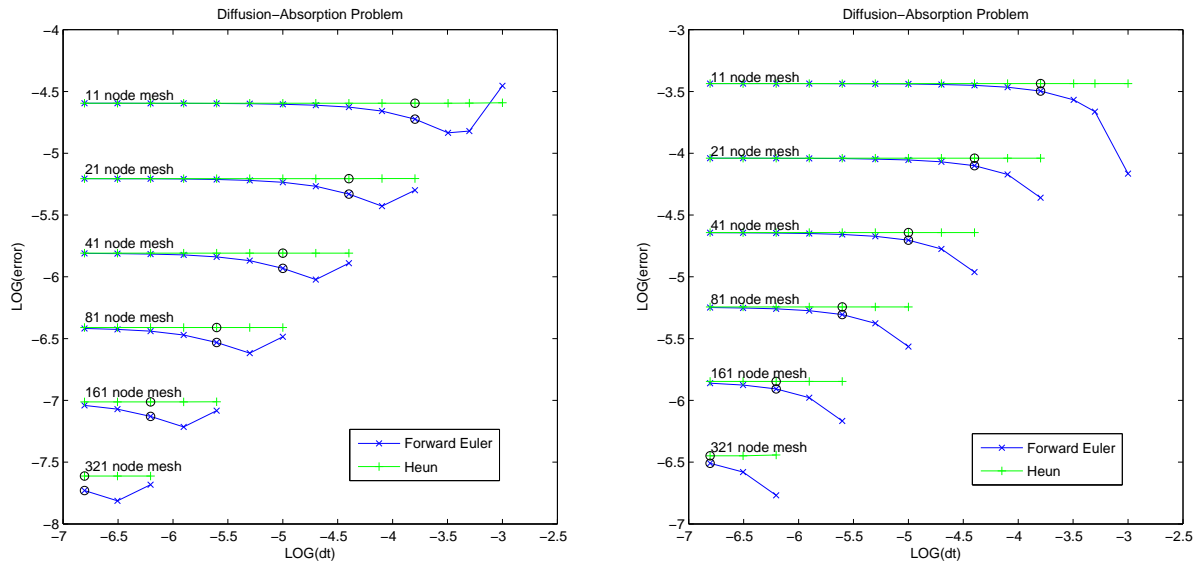
Figure 11: Comparison of L^2 errors in the solution (left) and the magnitudes of the errors in the boundary node positions (right) for the different time derivative approximations used with the one-dimensional diffusion-absorption problem. For each mesh, increasing the value of the time-step by a factor of 2 from that used to obtain the rightmost point leads to mesh tangling.