University of Leeds
SCHOOL OF COMPUTING
RESEARCH REPORT SERIES
Report 2003.04

A Moving Finite Element Method using Monitor Functions

by

M J Baines, M E Hubbard & P K Jimack

February 2003

Abstract

The recent study of geometric integration using scale invariance has further motivated the solution of nonlinear partial differential equations on moving meshes. Many mesh movement algorithms use monitor functions which can be understood both in relation to scale invariance and the Geometric Conservation Law. In this report we review the background and generalise the approach so as to construct a Moving Finite Element method using monitor functions, applicable to multidimensional nonlinear scalar PDEs.

The method is illustrated on nonlinear diffusion equations with moving boundaries using mass conservation as the monitor.

1 Introduction

Recent interest in geometric methods for the solution of differential equations has stimulated new approaches to numerical methods. Many partial differential equations (PDEs) of interest have invariance properties, seen through the study of invariance groups and self-similar solutions, which have prompted the construction of numerical methods which preserve them. Moreover, physically based PDEs are invariant to scalings of both independent and dependent variables, suggesting that these variables should be treated alike in numerical methods. A natural consequence of these viewpoints in the numerical solution of evolutionary problems is the use of moving meshes and these developments have given fresh impetus to their study.

Budd and co-authors (for references see [4]) have used the invariance properties of PDEs to motivate moving mesh methods. One of their applications is the solution of the one-dimensional Porous Medium Equation (PME) with a moving boundary. A global invariance property (conservation of mass) is used locally to determine a monitor function which generates a moving mesh that maintains scale invariance. A radial self-similar solution is used to test the method. A discrete form of the Oleinik comparison principle [8] is demonstrated in which numerical solutions with general initial data are trapped between self-similar solutions which approach an attractor as time evolves.

In [5] Cao, Huang and Russell discuss a link between monitor functions and the Geometric Conservation Law (GCL) [9]. They also introduce a mesh velocity potential that satisfies an elliptic problem. We shall follow their approach in our finite element method.

In [2, 3] Blake and Baines use the ideas in [4] to generate an ordinary differential equation system for the mesh velocity and solution, which is solved by a finite volume method. The method is illustrated on several problems in 1-D including the PME and a problem involving blow up. In the present report we shall generalise this approach to finite elements, allowing the generalisation to higher dimensional problems to be treated simply.

The report is organised as follows. In the next two sections the application of the local invariance property of [4] is described in terms of mass conservation and the use of more general monitor functions in one and two dimensions. In two dimensions an elliptic problem is derived using the approach in [5] which generates equations for a monitor velocity potential and a stream function.

The approach is then generalised to incorporate a weighted form of the invariance equation, resulting in a finite element method with moving nodes.

The method is tested against a radial self-similar solution of the 2-D PME with a moving boundary and is shown to be of second order. It is then applied to non-radially symmetric problems in order to test the comparison principle referred to in [4]. Not only is the comparison principle upheld but a comparison principle also holds for the moving boundary.

In an Appendix the principle on which this moving finite element method is based is compared with that of the Moving Finite Element of Miller [7, 1].

2 Background

2.1 Local Mass Conservation

In [4], using invariance arguments, conservation of the total mass is applied locally in 1-D on an arbitrary moving interval $\Delta x(t)$ in the form

$$\frac{d}{dt}\int_{\Delta x(t)} u\,dx = 0 \tag{1}$$

in order to construct a moving mesh method. From (1)

$$\int_{\Delta x(t)} u\,dx = \text{constant in time.} \tag{2}$$

It follows from (1) that

$$\frac{d}{dt}\int_{\Delta x(t)} u\,dx = \int_{\Delta x(t)} u_t\,dx + [u x_t]_{\Delta x(t)} = 0 \tag{3}$$

where $\Delta x(t)$ is the moving interval, $[\cdot]$ denotes the jump in the argument across the interval and x_t is the distortional velocity.

The integrand u_t is given in terms of x and u by the PDE. For example, in the case of the one-dimensional Porous Medium Equation

$$u_t = (u^m u_x)_x \tag{4}$$

equation (3) gives

$$-[u x_t] = [u^m u_x] \tag{5}$$

which leads to

$$-u x_t = u^m u_x + \text{constant}. \tag{6}$$

In reference [4] the mesh velocity x_t obtained from (3) is substituted back into a transformed PDE (equation (77) of the Appendix) in order to determine the solution for u. However in this report we shall use a much simpler reconstruction based on (2).

2.2 Use of Monitor Functions

Conservation of the integral of a more general monitor function $M(u)$ can also be applied to the moving interval $\Delta x(t)$ in the form

$$\frac{d}{dt} \int_{\Delta x(t)} M(u)\, dx = 0 \tag{7}$$

in order to construct a more general moving mesh method.

From (7)

$$\int_{\Delta x(t)} M(u)\, dx = \text{constant in time} \tag{8}$$

which is consistent with the standard equidistribution statement

$$\int_{\Delta x} M(u)\, dx = \Theta \int_{\cup \Delta x} M(u)\, dx$$

(see [5]) so long as the total integral of $M(u)$ on the right hand side of this equation remains constant.

It follows that

$$\frac{d}{dt} \int_{\Delta x(t)} M(u)\, dx = \int_{\Delta x(t)} M(u)_t\, dx + [M(u) x_t]_{\Delta x(t)} = 0\,. \tag{9}$$

In conservation form (9) may be written as

$$\int_{\Delta x(t)} \left(\frac{\partial M(u)}{\partial t} + \frac{\partial (M(u)x_t)}{\partial x} \right) dx = \int_{\Delta x(t)} \left(\frac{\partial J}{\partial t} + \frac{\partial (Jx_t)}{\partial x} \right) dx = 0 \qquad (10)$$

where J is the Jacobian of the transformation from the fixed frame to the moving frame. In the case of the mass monitor $M(u) = u = \xi_x = J$ (see [1]), hence

$$\frac{d}{dt} \int_{\Delta x(t)} J \, dx = \frac{d}{dt} \int_{\Delta \xi} d\xi = 0 \, . \qquad (11)$$

As pointed out in [5], equation (11) is equivalent to the integral form of a Geometric Conservation Law (GCL)[9], which expresses the necessary condition that an arbitrary volume should be conserved under the distortional velocity x_t. The monitor function $M(u)$ in (7) can therefore be interpreted as the Jacobian $J = \xi_x$ of the transformation from the moving frame to the fixed frame.

In higher dimensions (10) and (11) generalise to

$$\int_{\Omega(t)} \left(\frac{\partial J}{\partial t} + \nabla \cdot (J\mathbf{x}_t) \right) d\Omega = \frac{d}{dt} \int_{\Omega(t)} J \, d\Omega = \frac{d}{dt} \int_{\Sigma} d\Sigma = 0 \qquad (12)$$

where $d\Sigma$ and $d\Omega$ are elements in the fixed and moving frames, respectively, and, as in [5], this is a form of GCL. The monitor function $M(u)$ can again be identified with the Jacobian J of the transformation from the moving frame to the fixed frame.

In this report we restrict attention to the mass monitor $M(u) = u$ although the method can be generalised to other monitor functions, including those depending on u_x [2, 3]

2.3 A Numerical Approach in 1-D

In contrast to [4] Blake and Baines in [2, 3] solve the velocity equation (3) directly. Taking the interval $\Delta x(t)$ to be a finite volume with end points $X_{i-1}(t), X_i(t)$ and u to be a piecewise linear approximation U, equation (3) gives

$$-[U(X)_t]_{X_{i-1}}^{X_i} = \int_{X_{i-1}}^{X_i} \frac{\partial U}{\partial t} \, dx \qquad (13)$$

$\forall i$. In the case of the PME, for example, (13) can be written

$$-[UX_t]_{X_{i-1}}^{X_i} = [U^m U_x]_{X_{i-1}}^{X_i} \qquad (14)$$

which implies that

$$-[UX_t]_i = [U^m U_x]_i \tag{15}$$

(as long as U_x and X_t vanish at the same location).

Coupled with equation (1) in the discretised algebraic form

$$\frac{1}{2}(X_i - X_{i-1})(U_{i-1} + U_i) = c_i \tag{16}$$

$\forall i$ (as in [4]), equation (13) can be written as an ODE system for the mesh locations $X_i(t)$, which may be solved by a BDF package (see [2, 3]), and the U_i values can then be recovered from (16).

For the PME equation (4) resolution of the steep front is poor in the case of the mass monitor if the initial mesh is evenly spaced, but is improved if the nodes are packed in close to the front initially. Other monitors are considered in [2, 3] to improve the resolution. A blow-up equation is also treated.

3 Local Mass Conservation in Higher Dimensions

Applying invariance of the total mass locally to an arbitrary moving element Ω in any number of dimensions we have (cf. (1))

$$\frac{d}{dt}\int_{\Omega(t)} u \, d\Omega = 0 \tag{17}$$

or, equivalently,

$$\int_{\Omega(t)} u \, d\Omega = \text{constant in time} \tag{18}$$

It follows from (17) that

$$\frac{d}{dt}\int_{\Omega(t)} u \, d\Omega = \int_{\Omega(t)} u_t \, d\Omega + \oint_{\partial\Omega(t)} u\mathbf{x}_t \cdot d\mathbf{S} = 0 \tag{19}$$

(cf. (3)), where $\mathbf{x}_t$ is the distortional velocity. This leads, via the Divergence Theorem, to

$$-\int_{\Omega(t)} \nabla \cdot (u\mathbf{x}_t) \, d\Omega = \int_{\Omega(t)} \frac{\partial u}{\partial t} \, d\Omega \tag{20}$$

where $\frac{\partial u}{\partial t}$ is given by the PDE.

In the case of the multidimensional PME,

$$u_t = \nabla \cdot (u^m \nabla u) \tag{21}$$

equation (20) becomes (cf. (5))

$$-\int_{\Omega(t)} \nabla \cdot (u\mathbf{x}_t)\, d\Omega = \int_{\Omega(t)} \nabla \cdot (u^m \nabla u)\, d\Omega\,. \tag{22}$$

In the radially symmetric case (22) becomes

$$-\int_{\Delta r(t)} \left(\frac{1}{r}\frac{\partial}{\partial r}(rur_t)\right) r\, dr = \int_{\Delta r(t)} \frac{1}{r}\frac{\partial}{\partial r}\left(ru^m \frac{\partial u}{\partial r}\right) r\, dr \tag{23}$$

leading to (cf. (5))

$$-[rur_t] = [ru^m u_r]\,. \tag{24}$$

3.1 The Mesh Potential and Stream Function

In [5] Cao, Huang and Russell point out that equation (20) is insufficient by itself to determine $\mathbf{x}_t$. However, by adding a *curl* condition of the form

$$curl(\omega \mathbf{x}_t) = \zeta \tag{25}$$

where ω is a positive function and ζ is a prescribed vorticity, and applying a suitable boundary condition on $\mathbf{x}_t$, they derive an elliptic problem which has a unique solution.

In [5] the vorticity is written

$$\zeta = curl(\omega \mathbf{v}) \tag{26}$$

where ω is an arbitrary positive weight function and $\mathbf{v}$ is an arbitrary prescribed velocity. It then follows from (25) and (26) that there exists a potential function ϕ such that

$$\omega \mathbf{x}_t = \omega \mathbf{v} + \nabla \phi\,. \tag{27}$$

Substitution of $\mathbf{x}_t$ into (20) yields an equation for the potential

$$-\int_{\Omega(t)} \nabla \cdot (\omega^{-1} u \nabla \phi)\, d\Omega = \int_{\Omega(t)} \frac{\partial u}{\partial t}\, d\Omega + \int_{\Omega(t)} \nabla \cdot (u\mathbf{v})\, d\Omega \tag{28}$$

which is the integral form of the elliptic equation

$$-\nabla \cdot (\omega^{-1} u \nabla \phi) = \frac{\partial u}{\partial t} + \nabla \cdot (u\mathbf{v})\,. \tag{29}$$

For the PME this becomes

$$-\nabla \cdot (\omega^{-1} u \nabla \phi) = \nabla \cdot (u^m \nabla u) + \nabla \cdot (u\mathbf{v}) \tag{30}$$

which can be solved for ϕ with suitable boundary conditions. For example, if the normal component of $\mathbf{x}_t$ is given on the boundary, $\frac{\partial \phi}{\partial n}$ is prescribed while if the tangential component of $\mathbf{x}_t$ is given, ϕ is prescribed. If neither is given then no boundary condition is imposed on ϕ, but the solution is only unique to within a constant in such a case.

Another route to an elliptic equation in 2-D when the PDE is in conservation form is through the stream function ψ associated with (20). If

$$u_t = \nabla \cdot \mathbf{F} \tag{31}$$

equation (20) has the pointwise form

$$\nabla \cdot (u\mathbf{x}_t + \mathbf{F}) = 0 \tag{32}$$

showing that there exists a stream function ψ such that

$$-u\mathbf{x}_t = \mathbf{F} - \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x} \right)^{\mathrm{T}}\,. \tag{33}$$

Substitution into (25) yields

$$\mathbf{k} \cdot \nabla \times \left(-\omega u^{-1} \mathbf{F} + \omega u^{-1} \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x} \right)^{\mathrm{T}} \right) = \zeta \tag{34}$$

where $\mathbf{k}$ is perpendicular to the plane, giving

$$\nabla \cdot (\omega u^{-1} \nabla \psi) = \zeta + \mathbf{k} \cdot \nabla \times \left(\omega u^{-1} \mathbf{F} \right)\,. \tag{35}$$

The boundary conditions are the opposite of those for the previous elliptic equation. The streamlines are the trajectories of the points in the moving frame.

For the PME (35) becomes

$$\nabla \cdot (\omega u^{-1} \nabla \psi) = \zeta + \mathbf{k} \cdot \nabla \times \left(\omega u^{m-1} \nabla u \right)\,. \tag{36}$$

3.2 A Numerical Method

In two dimensions $u, \mathbf{x}$ may be taken to be piecewise linear functions $U, \mathbf{X}$ on a triangulation of the region with $\Omega_i(t)$ as a typical triangle. Higher dimensions may be treated similarly.

Equations (17) and (18) become

$$\frac{d}{dt}\int_{\Omega_i(t)} U\, d\Omega = 0 \qquad\qquad \int_{\Omega_i(t)} U\, d\Omega = c_i \tag{37}$$

while the velocity equation (20) is

$$-\int_{\Omega_i(t)} \nabla\cdot(U\mathbf{X}_t)\ d\Omega = \int_{\Omega_i(t)} \frac{\partial U}{\partial t}\, d\Omega\,. \tag{38}$$

For the PME the mesh potential equation (28) becomes

$$-\int_{\Omega_i(t)} \nabla\cdot(\omega^{-1}U\nabla\Phi)\, d\Omega = \int_{\Omega_i(t)} \nabla\cdot(U^m\nabla U)\ d\Omega + \int_{\Omega_i(t)} \nabla\cdot(U\mathbf{V})\, d\Omega \tag{39}$$

and the mesh velocity is then given by

$$\omega\mathbf{X}_t = \omega\mathbf{V} + \nabla\Phi\,.$$

However this finite volume approach is less successful in generalising to higher dimensions than the finite element approach described below.

4 A Moving Finite Element Method

We now describe a moving finite element approach in which the invariance property (1) or (17) is distributed in space by weight functions w_i which collectively form a partition of unity at each t so as to be consistent with global conservation.

We choose the functions w_i to be the linear finite element hat or tent functions $w_i(\mathbf{x}, \vec{\mathbf{X}}(t))$ associated with node i, where $\vec{\mathbf{X}}(t)$ is a vector of nodal positions, and $\Omega(t)$ to be $\Pi_i(t)$, the union of the elements abutting node i.

With u approximated by a piecewise linear function U the distributed local form of mass conservation can be written (cf. (1) or (17))

$$\frac{d}{dt}\int_{\Pi_i(t)} w_i(\mathbf{x}, \vec{\mathbf{X}}(t))\, U(\mathbf{x}, t)\, d\Omega = 0 \tag{40}$$

or, equivalently,

$$\int_{\Pi_i(t)} w_i(\mathbf{x}, \vec{\mathbf{X}}(t))\, U(\mathbf{x}, t)\, d\Omega = c_i\,. \tag{41}$$

From (40)

$$\begin{aligned} 0 &= \int_{\Pi_i(t)} \left(w_i \frac{\partial U}{\partial t} + U \frac{\partial w_i}{\partial t} \right) d\Omega + \oint_{\partial\Pi_i(t)} w_i U \mathbf{X}_t \cdot d\mathbf{S} \\ &= \int_{\Pi_i(t)} \left(w_i \frac{\partial U}{\partial t} + U \frac{\partial w_i}{\partial t} \right) d\Omega\,, \end{aligned} \tag{42}$$

where $\mathbf{X}_t$ is the distortional velocity, given by

$$\mathbf{X}_t = \sum_j \dot{\mathbf{X}}_j w_j\,, \tag{43}$$

where $\dot{\mathbf{X}}_j$ is the velocity of node j, and the boundary integral vanishes because either w_i or U is assumed to be zero on $\partial\Pi_i(t)$.

Now

$$\frac{\partial w_i}{\partial t} = \sum_j \frac{\partial w_i}{\partial \mathbf{X}_j} \dot{\mathbf{X}}_j \tag{44}$$

and, for linear basis functions w_i,

$$\frac{\partial w_i}{\partial \mathbf{X}_j} = -(\nabla w_i)\, w_j \tag{45}$$

(see [6]), so

$$\frac{\partial w_i}{\partial t} = -\sum_j (\nabla w_i) \cdot (w_j \dot{\mathbf{X}}_j) = -\mathbf{X}_t \cdot \nabla w_i\,. \tag{46}$$

Hence (42) becomes

$$\int_{\Pi_i(t)} \left(w_i \frac{\partial U}{\partial t} - U \mathbf{X}_t \cdot \nabla w_i \right) d\Omega = 0 \tag{47}$$

leading to

$$\int_{\Pi_i(t)} w_i \left(\frac{\partial U}{\partial t} + \nabla \cdot (U \mathbf{X}_t) \right) d\Omega = 0\,. \tag{48}$$

since the boundary integral again vanishes (cf. (43)). Equation (48) is a weighted form of the velocity equation (20). By introducing a discretised potential function Φ defined by

$$\omega \mathbf{X}_t = \omega \mathbf{V} + \nabla \Phi \tag{49}$$

(cf. (27)), equation (48) becomes the mesh potential equation, $\forall i$:

$$\int_{\Pi_i(t)} \omega^{-1} U (\nabla \Phi) \cdot (\nabla w_i) \, d\Omega = \int_{\Pi_i(t)} \left(w_i \frac{\partial U}{\partial t} - U \mathbf{V} \cdot (\nabla w_i) \right) d\Omega \, . \tag{50}$$

In the case of the PME, by using the weak form of (21) equation (50) becomes $\forall i$:

$$\int_{\Pi_i(t)} \omega^{-1} U (\nabla \Phi) \cdot (\nabla w_i) \, d\Omega = \int_{\Pi_i(t)} (U^m \nabla U + U \mathbf{V}) \cdot (\nabla w_i) \, d\Omega \tag{51}$$

4.1 Matrix Forms

Expanding Φ in the basis functions w_j as

$$\Phi = \sum_j \Phi_j w_j \tag{52}$$

equation (50) leads in the standard way to the matrix equation

$$K(\vec{\mathbf{X}}, \vec{U}) \vec{\Phi} = \vec{f} \tag{53}$$

where $\vec{\Phi}$ is the vector of coefficients Φ_j. Here

$$K(\vec{\mathbf{X}}, \vec{U}) = \{K_{ij}\} \qquad K_{ij} = \int_{\Pi_i(t)} \omega^{-1} U (\nabla w_j) \cdot (\nabla w_i) \, d\Omega \tag{54}$$

is a weighted stiffness matrix and

$$\vec{f} = \{f_i\} \qquad f_i = \int_{\Pi_i(t)} \left(w_i \frac{\partial U}{\partial t} - U \mathbf{V} \cdot \nabla w_i \right) d\Omega \tag{55}$$

Expanding U in the same basis functions

$$U = \sum_j U_j w_j \tag{56}$$

equation (41) gives

$$A(\vec{\mathbf{X}}) \vec{U} = \vec{c} \tag{57}$$

where $\vec{U}$ is the vector of coefficients U_j. Here

$$A(\vec{\mathbf{X}}) = \{A_{ij}\} \qquad A_{ij} = \int_{\Pi_i(t)} w_i w_j \, d\Omega \tag{58}$$

is a mass matrix and

$$\vec{c} = \{c_i\} \qquad\qquad c_i = \text{constant}$$

are values determined by the initial distribution.

The boundary conditions on (58) are not imposed strongly, otherwise the result is inconsistent with mass conservation and also with the velocity equation (54). The value at the boundary is therefore only achieved weakly.

A recovery step for deducing $\dot{\mathbf{X}}$ from Φ is required. Since Φ is piecewise linear a piecewise linear $\dot{\mathbf{X}}$, where $\mathbf{X}$ is of the form

$$\mathbf{X} = \sum_j \mathbf{X}_j w_j \,, \tag{59}$$

must be recovered weakly. A suitable recovery step is the weak equation

$$\int_{\Pi_i(t)} w_i \left((\dot{\mathbf{X}} - \mathbf{V}) - \omega^{-1} \nabla \Phi \right) d\Omega = 0 \tag{60}$$

(cf. (27)), which yields the matrix equations

$$B(\vec{\mathbf{X}}) \vec{X}_t^{(1)} = \vec{b}^{(1)} \qquad\qquad B(\vec{\mathbf{X}}) \vec{X}_t^{(2)} = \vec{b}^{(2)} \tag{61}$$

where $\vec{X}^{(d)}$ is the vector of coefficients $X_j^{(d)}$ for $d = 1, 2$, and each component of $\vec{\mathbf{X}}$ is given by $\mathbf{X}_j = (X_j^{(1)}, X_j^{(2)})^{\mathrm{T}}$. Here

$$B(\vec{\mathbf{X}}) = \{B_{ij}\} \qquad\qquad B_{ij} = \int_{\Pi_i(t)} w_i w_j \, d\Omega \tag{62}$$

is a weighted mass matrix and the right hand side is

$$\vec{\mathbf{b}} = \{\mathbf{b}_i\} \qquad\qquad \mathbf{b}_i = \int_{\Pi_i(t)} w_i (\mathbf{V} + \omega^{-1} \nabla \Phi) \, d\Omega \,. \tag{63}$$

4.2 ALGORITHM

We evaluate the right hand side of the ODE system

$$\frac{d}{dt} \vec{\mathbf{X}} = \vec{\mathbf{F}}(\vec{\mathbf{X}}) \tag{64}$$

using the sequence

- Given $\vec{\mathbf{X}}$ recover $\vec{U}$ from (57)
- Given $\vec{U}$ calculate $\vec{\Phi}$ from (53)
- Calculate $\vec{\mathbf{F}}(\vec{\mathbf{X}})$ from (61)
- Return

This function may then be passed to a general ODE solver to allow the system (64) to be solved efficiently.

4.3 Example: The Porous Medium Equation

Following [4] and the references therein, for the 1-D porous medium equation (PME)

$$u_t = (u^m u_x)_x \tag{65}$$

the variables t, x, u are invariant under the scalings

$$t \to \lambda t, \qquad x \to \lambda^\beta x, \qquad u \to \lambda^\gamma u \tag{66}$$

provided that

$$\gamma - 1 = (m+1)\gamma - 2\beta\,. \tag{67}$$

If the boundary conditions are such that the total "mass"

$$\int_{x_0(t)}^{x_N(t)} u\,dx \tag{68}$$

is conserved in time, so that $\beta + \gamma = 0$, then

$$\beta = 1/(m+2), \qquad \gamma = -1/(m+2)\,. \tag{69}$$

From these scalings self-similar solutions with a moving boundary can be constructed of the form

$$u(x,t) = \begin{cases} \left(\frac{t_0}{t}\right)^{\frac{1}{m+2}} \left(1 - \left(\frac{x}{Kt^{\frac{1}{m+2}}}\right)^2\right) & x \le Kt^{\frac{1}{m+2}} \\ 0 & x > Kt^{\frac{1}{m+2}} \end{cases}, \tag{70}$$

where t_0, K are constants.

For the multidimensional PME

$$u_t = \nabla \cdot (u^m \nabla u) \tag{71}$$

with $u = 0$ at the moving boundary, the total mass is again invariant.

In the case of the 2-D radially symmetric PME

$$u_t = \frac{1}{r}\frac{\partial}{\partial r}\left(r u^m \frac{\partial u}{\partial r} \right) \tag{72}$$

with $u = 0$ at the boundary, the corresponding self-similar solutions are

$$u(r,t) = \begin{cases} \left(\frac{t_0}{t}\right)^{\frac{1}{m+1}} \left(1 - \left(\frac{r}{Kt^{\frac{1}{2(m+1)}}}\right)^2\right) & x \leq Kt^{\frac{1}{2(m+1)}} \\ 0 & x > Kt^{\frac{1}{2(m+1)}} \end{cases} \tag{73}$$

We shall use this solution to test the 2-D algorithm.

5 Numerical Results

The two-dimensional moving finite element method presented in the previous section is demonstrated here using the Porous Medium Equation (71). Initially we consider the radially symmetric self-similar solutions of the form (73) which can be used to indicate the accuracy of the method. The initial conditions for the radially symmetric numerical experiments are given by (73), in which the parameter K is chosen to give an initial radius of 0.5 and t is initialised to the corresponding value of t_0. The initial solution is taken to be a quadrature-based least-squares fit to the exact solution on the prespecified initial mesh at $t = t_0$, so the nodal solution values are not exact. Note that we take $\omega = 1$ and, initially, $\mathbf{V} = 0$ in (60) (cf. (27)), and $u = 0$ is never explicitly imposed on the boundary.

Figure 1 shows an $l1$ error measure approximated using high order quadrature for the numerical solutions at $T = 1$ (where $T = t - t_0$), for 3 exponents in (71), $m = 1, 2, 4$. The errors are calculated on a series of regular triangular meshes in which the circular domain is divided into 4 quadrants, each of which is triangulated using successive refinement (the finest mesh has 6 levels of refinement: 8321 nodes and 16384 cells). The experiments are conducted keeping $\Delta t/(\Delta x)^2$ constant, dictated by the use of an explicit time-stepping scheme for constant coefficients. The figure clearly shows the errors to be $O((\Delta x)^2)$ for $m = 1$ (given by a line of slope -1). The two other exponents, where the gradient of the exact solution is infinite at the boundary of the domain, give lower accuracy. Importantly, at any mesh point, the scaled variables, $\hat{u} = t^{1/(m+1)}u$ and $\hat{r} = t^{-1/(2(m+1))}r$ remain constant, as predicted by the theory [4].

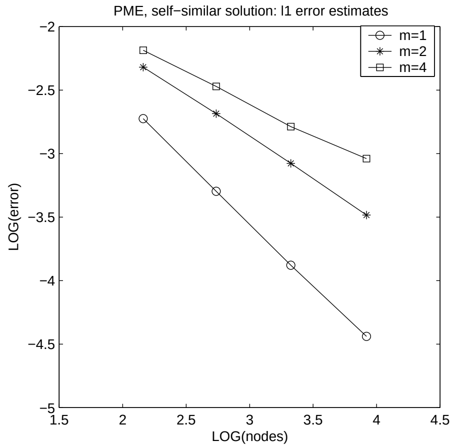

Figure 1: $l1$ error measures for the radially symmetric self-similar solution to the PME for $m = 1, 2, 4$ at $T = 1$.

A second measure of the accuracy of the solution is given by the error in the position of the boundary of the domain. Figure 2 shows how the normalised maximum and minimum radii of the computational domain, $r_{\text{max}}/r_{\text{exact}}$ and $r_{\text{min}}/r_{\text{exact}}$ respectively, vary as the initial mesh is refined (where r_{exact} is the exact radius). In each case, the maximum and minimum radii converge towards each other, and eventually to the exact radius. Note that mass, which is approximated on the initial mesh, is conserved to machine accuracy in every one of the numerical experiments presented here. This is crucial to the domain maintaining its circular shape over long time periods.

Clustering nodes towards the boundary can improve the quality of the solution, particularly in the $m = 2, 4$ cases, though it begins to deteriorate if the clustering is too pronounced, possibly because the simple algorithm used to create the initial meshes starts to pull nodes away from regions where the second derivatives of the solution are high.

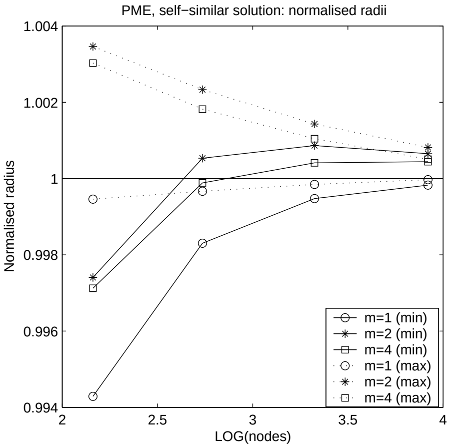

Figure 2: Normalised minimum and maximum radii for the radially symmetric self-similar solution to the PME for $m = 1, 2, 4$ at $T = 1$.

The evolution of the above three cases is illustrated in Figure 3, which contains a series of slices through the centre of the domain, taken through the approximate and exact solutions, alongside three-dimensional views of the numerical solution at $T = 2.0$ (all obtained on a genuinely unstructured, but still uniform, 2349 node, 4539 cell, mesh). The approximations are still accurate, but the figures clearly show that $u = 0$ is not enforced strongly at the boundary.

The new scheme is not restricted to modelling self-similar solutions, and within certain bounds, the evolution of any initial conditions can be predicted. Another good test is to check that they verify the comparison principle, or "sandwiching" property, of the continuous equations [4, 8], *i.e.* given three sets of initial conditions,

$$u_1(x, y, t_0) \leq u_2(x, y, t_0) \leq u_3(x, y, t_0) \quad \forall\,(x, y) \in \Omega\,, \tag{74}$$

then

$$u_1(x, y, t) \leq u_2(x, y, t) \leq u_3(x, y, t) \quad \forall\,(x, y) \in \Omega,\ t \geq t_0\,. \tag{75}$$

This can be seen in Figures 4 and 5, which show two experiments in which the initial conditions are perturbed. The first, Figure 4, applies a random perturbation to the initial solution values and compares its evolution with

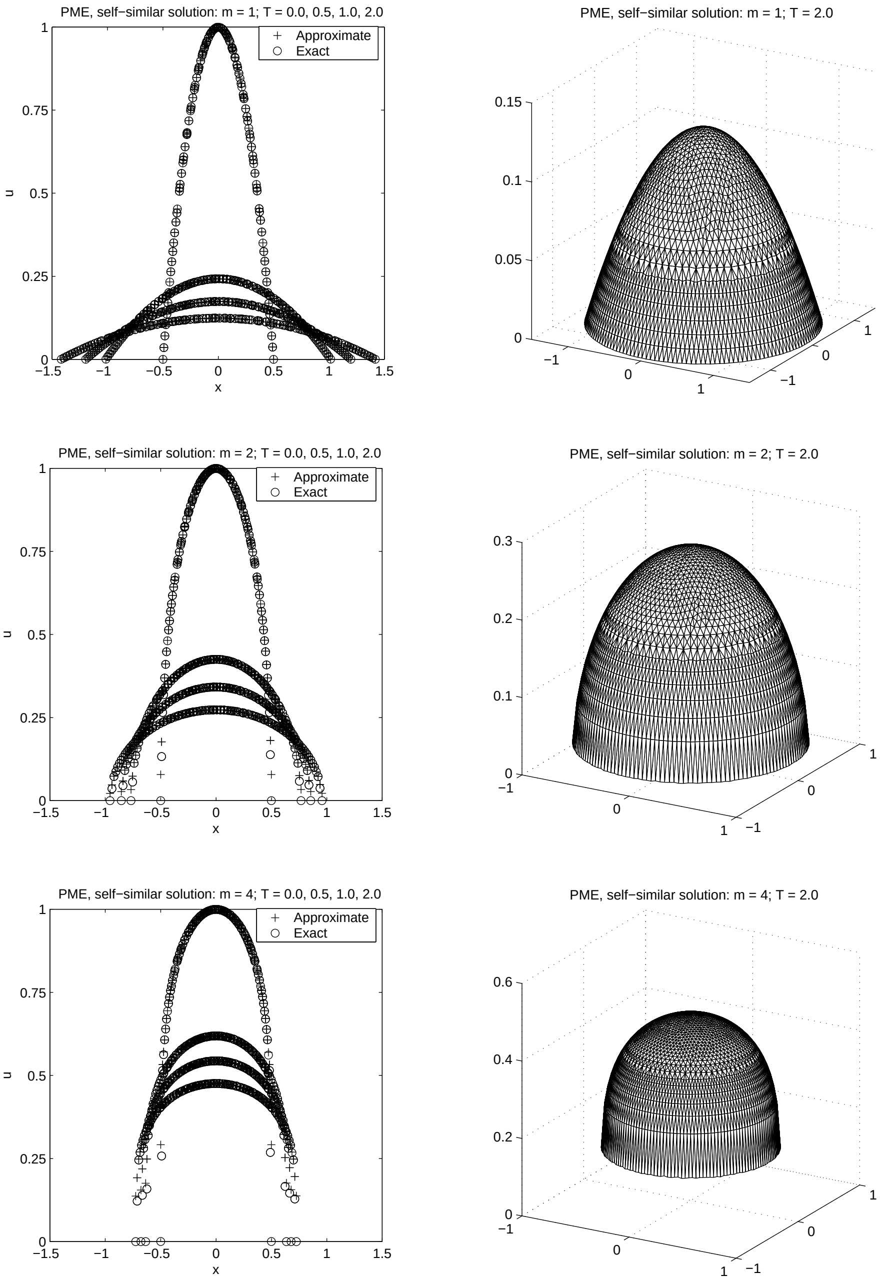

Figure 3: Slices along $y = 0$, taken through the radially symmetric self-similar solutions to the PME at various times, comparing exact with approximate (left), along with the approximation at $T = 2.0$ (right), for $m = 1$ (top), 2 (middle) and 4 (bottom).

solutions scaled according to the minimum and maximum perturbations. The second, Figure 5, applies a sinusoidal perturbation to the initial mesh, which is then sandwiched in a similar manner. Both cases verify this property, though the authors have yet to prove that it will always be satisfied. It is also interesting to note that the random perturbations in the initial conditions are smoothed out very rapidly – this is also true if the mesh (rather than the solution) is randomly perturbed.

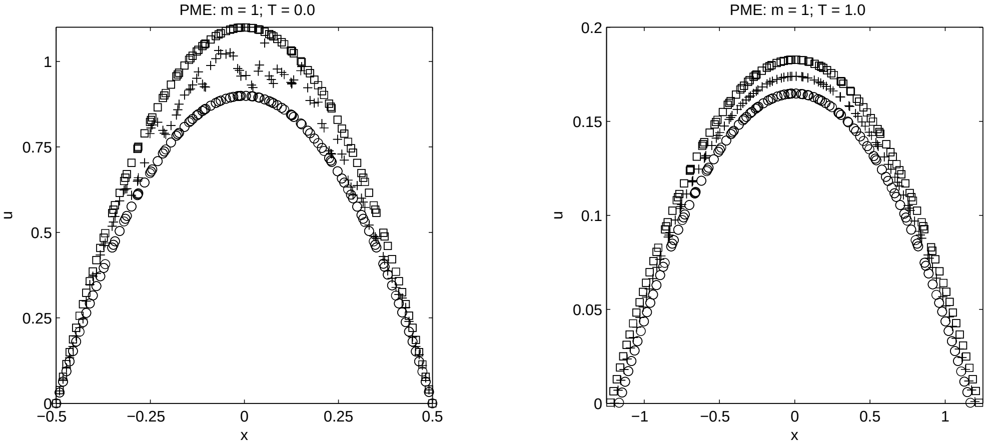

Figure 4: Slices along $y = 0$, taken through the initial conditions (left) and approximate solutions at $T = 1$ (right), illustrating the "sandwiching" of a randomly perturbed solution to the two-dimensional PME with $m = 1$.

It is also possible to impose a background velocity field $\mathbf{v}$ on the mesh movement equations via the extra vorticity term in (60) and (63) (cf. (27)). Figure 6 shows the final mesh and solution for the $m = 1$ self-similar test case described above, with a background mesh velocity of $\mathbf{v} = 2.5r(-y, x)^{\mathrm{T}}$. This gives a rotating mesh with a higher angular velocity at the outer boundary than near the centre. The solution remains close to that obtained on a fixed mesh, until tangling becomes inevitable. When a solid body rotation is imposed there is typically little loss of accuracy: sometimes there is actually an improvement.

Finally, a test case is presented which illustrates the limitations of the current algorithm. The initial profile, shown in Figure 7, is a "horseshoe" shape and its evolution is governed by the PME with $m = 1$. The second

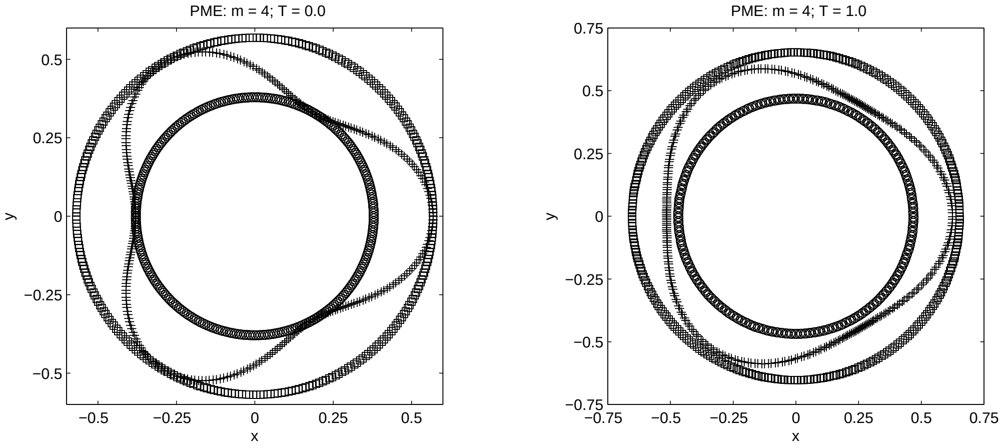

Figure 5: Horizontal slices along $u = 0.1$, taken through the initial conditions (left) and approximate solutions at $T = 1$ (right), illustrating the "sandwiching" of a sinusoidally perturbed mesh for the PME with $m = 4$.

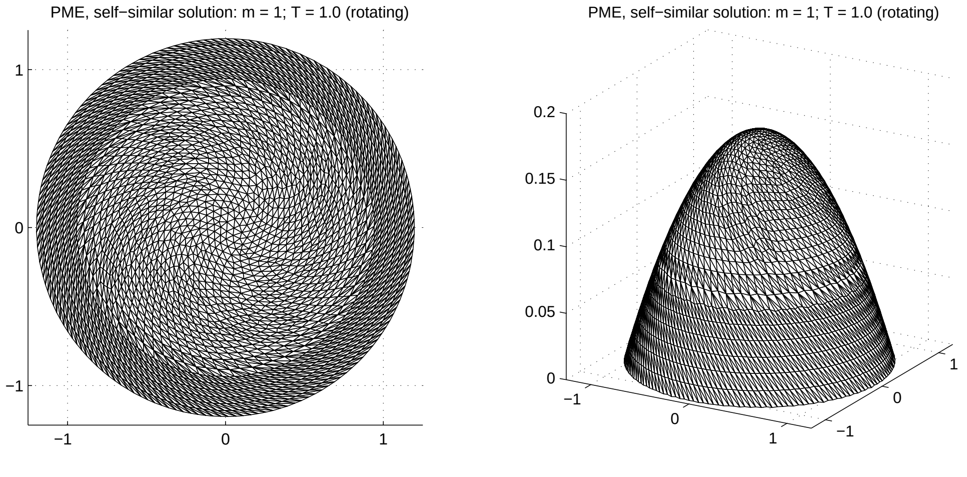

Figure 6: The mesh (left) and approximation at $T = 1$ (right) for the self-similar solution to the PME with $m = 1$ and a rotating background velocity field.

pair of pictures in Figure 7 show the computed solution at $T = 0.55$, when it begins to break down. There are two important aspects to this solution: (i) neither of the two ends of the horseshoe has any knowledge of the other's existence, so they overlap without interacting in any way; (ii) the hole in the centre is about to close up and, in doing so, will cause the cells surrounding it, which are currently long and thin, to tangle and give physically meaningless results. Both of these issues will be addressed in future work.

6 Conclusions

In this report the local invariance property of [4] is generalised to incorporate a weighted form of the invariance equation, resulting in a finite element method with moving nodes. In two dimensions an elliptic problem is derived using the approach in [5] which generates equations for a monitor velocity potential and a stream function. The method is tested against a radial self-similar solution of the 2D PME and shown to be of second order accuracy in space. The numerical results show that the approach is successful in reproducing several aspects of the theory (including local mass conservation and the comparison property of the PME) although not boundary interaction as yet. The comparison principle also appears to hold for the moving boundary.

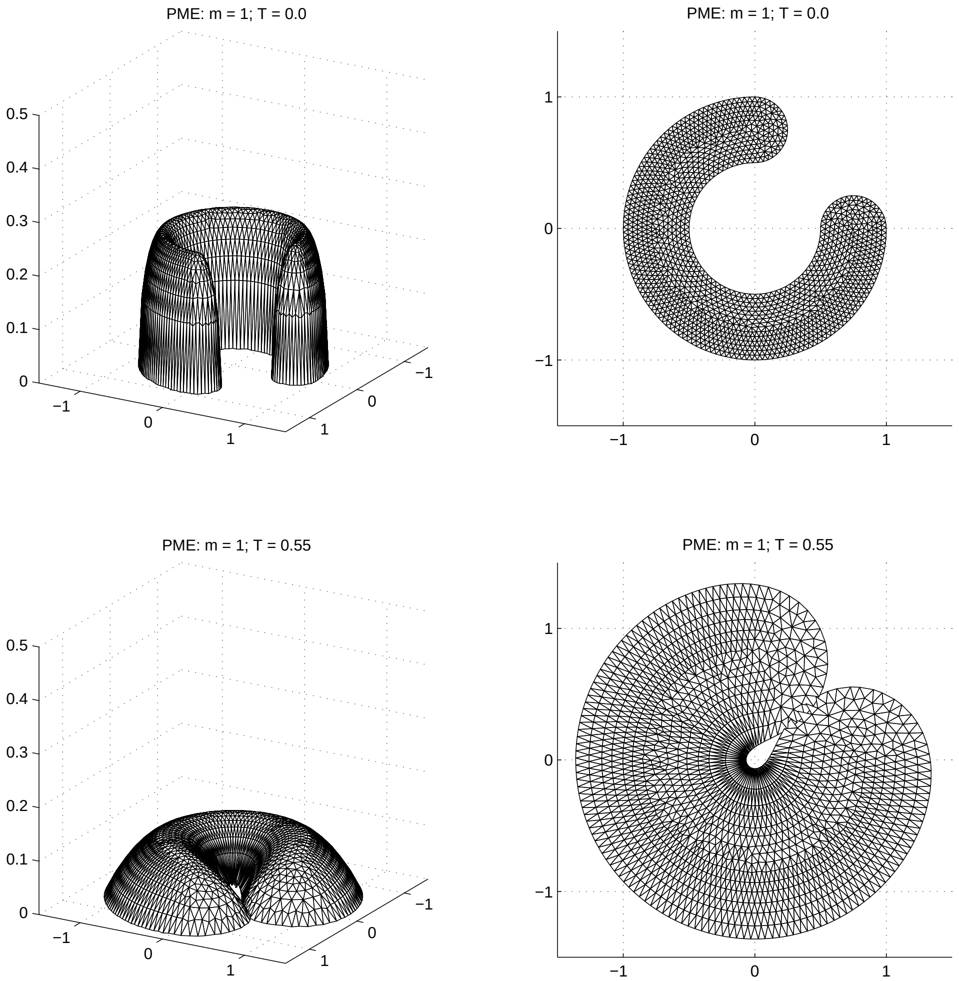

Figure 7: Initial solution and mesh (top), and the approximate solution and mesh at time $T = 0.55$ (bottom) for the “horseshoe” test case.

References

[1] Baines, M.J., Moving Finite Elements, OUP (1994).

[2] Blake, K.W., Moving Mesh Methods for Nonlinear Parabolic Partial Differential Equations, PhD thesis, Dept. of Mathematics, University of Reading, UK (2001)

[3] Blake, K.W. and Baines, M.J., A moving mesh method for non-linear parabolic problems, Numerical Analysis Report 2/02, Dept. of Mathematics, University of Reading, UK (2002)

[4] Budd, C.J. and Piggott, M., Geometric Integration and its Applications, J. Comp. Appl. Math., **128**:399–422 (2001).

[5] Cao, W., Huang, W. and Russell, R.R., A Moving Mesh Method based on the Geometric Conservation Law, SIAM J. Sci. Comp., **24**:118–142 (2002),

[6] Jimack, P.K. and Wathen, A.J., Temporal Derivatives in the Finite Element Method on Continuously Deforming Grids, SIAM J. Num. Anal., **28**:990–1003 (1991).

[7] Miller, K. and Miller, R.N., Moving Finite Elements, SIAM J. Num. Anal., **18**:1019–1057 (1982).

[8] Oleinik, O. et al., Istv. Akad. Nauk. SSR Ser. Mat., **22**:667–704 (1958).

[9] Thomas, P.D. and Lombard, C.K., The Geometric Conservation Law and its Application to Flow Computations on Moving Grids, AIAA Journal, **17**:1030–1037 (1979).

Appendix: Comparison of MFEs

It is interesting to set the present method alongside the earlier Moving Finite Element method of Miller [7, 1]. Transforming to a moving frame $(\mathbf{x}, t)$ the chain rule gives

$$\frac{\partial \hat{u}}{\partial \tau} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{\partial \hat{x}}{\partial \tau} \tag{76}$$

and the PDE (4) becomes

$$\frac{\partial \hat{u}}{\partial \tau} - \frac{\partial u}{\partial x}\frac{\partial \hat{x}}{\partial \tau} = \frac{\partial}{\partial x}\left(u^m \frac{\partial u}{\partial x}\right) . \tag{77}$$

There are two unknowns $\frac{\partial \hat{u}}{\partial \tau}$ and $\frac{\partial \hat{x}}{\partial \tau}$ in (77) which require two equations to determine them. One is the PDE but the other must be obtained from a separate principle.

Both the Moving Finite Element method of Miller and the present method use the weak form

$$\int w \left(\frac{\partial \hat{u}}{\partial \tau} - \frac{\partial u}{\partial x}\frac{\partial \hat{x}}{\partial \tau} - Lu\right) dx = 0 \tag{78}$$

with $\frac{\partial \hat{u}}{\partial \tau}$ and $\frac{\partial \hat{x}}{\partial \tau}$ as linear Finite Element functions.

Miller's method uses two weak forms of the type (78), one with w as the standard linear test function and one with w as $-\frac{\partial u}{\partial x}$ times this function, arising from a least squares minimisation of the residual.

In the present method the weak form (78) is rearranged as

$$\frac{d}{dt}\int wu \, dx - \int w \left(u \frac{\partial \hat{x}}{\partial \tau} + Lu\right) dx = 0 \tag{79}$$

and each term on the left hand side set to zero separately.