THE UNIVERSITY OF READING

Mesh Movement by Mass Conservation

by

K.W. Blake

Numerical Analysis Report 1/2000

The University of Reading
P O Box 220
Reading RG6 6AX
Berkshire

DEPARTMENT OF MATHEMATICS

Mesh Movement by Mass Conservation

K.W.Blake
Department of Mathematics
University of Reading
P O Box 220, Reading, RG6 6AX, UK

May 2000

Abstract

A moving grid method for the solution of the porous media equation in one dimension which involve a mass conservation property is investigated. Following a moving mesh partial differential equation approach, where nodes are moved by equidistributing a measure of the solution, the algorithm here simply conserves mass in each computational cell and moves the nodes appropriately Furthermore a simple grid refinement algorithm near to the moving front is presented to work in conjunction with the outlined method. Numerical results are shown in one-dimension and compared against reference solutions found in insect dispersal models.

1 Introduction

In many areas moving mesh partial differential equations (MMPDE's) have been used to improve the numerical solution of partial differential equations. These MMPDE's prescribe the movement of nodes relative to a computational mesh. In most cases these MMPDE's are derived from an equidistribution principle [2] which controls the grid in such a way so that a predetermined measure of the solution or geometry of the solution is distributed equally over the entire mesh.

Generally the use of such methods involves having two equations to solve, one for the movement of the mesh and one for the solution of the underlying partial differential equation. Presented in this report is a method for the numerical solution of the one-dimensional Porous Media Equation (PME) devised in such a way that the MMPDE and the PME are combined, yielding one system of equations to solve and hence making a computational saving. The PME is written as

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(u^m \frac{\partial u}{\partial x}) \tag{1}$$

where m is real and positive, with initial condistions symmetric about $x = 0$ and Neumann boundary conditions imposed at each outward moving boundary.

Section 2 outlines the method, in particular we see how the technique takes advantages of two major features of the PME, namely its mass conservation property and its singular behaviour at the moving boundary. An approximation to the speed of the moving boundary is derived from the conservation of mass,a simple algorithm for grid refinement is explained and the idea of representing any steep front formations with an appropriate polynomial expression is explored. Section 3 shows numerical results for various values of m. Numerical results are compared against an analytical solution found in Murray [1]. Finally section 4 outlines ideas for further work.

2 Moving Mesh Partial Differential Equation by Mass Conservation

As stated we illustrate the derivation and performance of the method by using the porous media equation (1), two features of which will be of importance when deriving the MMPDE. Firstly the solution has a moving boundary which moves with finite speed. If the boundary's position is denoted by the point $x_{N+1}(t)$ then we have that $u(x_{N+1}(t), t) = 0$ for all time t. Secondly the total mass u is conserved over the region $(0, x_{N+1}(t))$ under consideration.

2.1 Conserving mass and moving the nodes

Generally MMPDE's are derived from an equidistribution principle which moves nodes such that a measure of the solution is spread equally over each computational cell [2]. However, by taking advantage of the conservation of mass property we can move the nodes by ensuring that the mass contained in each cell is kept constant throughout time [3]. In each cell $x_{i+\frac{1}{2}}$ we approximate the mass $C_{i+\frac{1}{2}}$ using a simple trapezium rule integration,

$$C_{i+\frac{1}{2}} = \frac{1}{2}(u_i + u_{i+1})(x_{i+1} - x_i), \tag{2}$$

and fix it for all time.

The rate of change of the integral of the mass between a pair of adjacent moving nodes $x_i(t)$ and $x_{i+1}(t)$ is

$$\frac{\partial}{\partial t}\int_{x_i}^{x_{i+1}} u dx = \int_{x_i(t)}^{x_{i+1}(t)} u_t dx + u_{i+1}\dot{x}_{i+1} - u_i\dot{x}_i$$

If the grid is to be moved whilst conserving mass between adjacent nodes then the left-hand side of this equation will equal zero. Moreover we can substitute the porous media equation (1) into the integral on the right-hand side to get

$$u_i\dot{x}_i - u_{i+1}\dot{x}_{i+1} = \int_{x_i(t)}^{x_{i+1}(t)} (u^m u_x)_x dx.$$

Using upwinding for discretizing the derivative terms on the right-hand side we are left with the discretization

$$u_{i+1}\dot{x}_{i+1} = u_i\dot{x}_i + u^m_{i-\frac{1}{2}}\left(\frac{u_i - u_{i-1}}{x_i - x_{i-1}}\right) - u^m_{i+\frac{1}{2}}\left(\frac{u_{i+1} - u_i}{x_{i+1} - x_i}\right), \tag{3}$$

where $u_{i+\frac{1}{2}}$ is the value of u evaluated at the centre of the $i + \frac{1}{2}$th cell.

Because we are considering a half-section of the problem (the solution is symmetric about $x = 0$) we put $\dot{x}_0 = 0$ since this is the centre of mass and it can be shown that this is constant throughout time [3]. Equation (3) yields a bi-diagonal system of ODE's to solve. By using the fact that $u_{N+1} = 0$ for all time t we can express all values of u_i in terms of the current positions of the grid points $\underline{x}$. Hence from (2) we have that

$$u_i = 2\sum_{k=i+1}^{N+1} C_{k+\frac{1}{2}}\frac{(-1)^{k-i-1}}{x_k - x_{k-1}}. \tag{4}$$

Finally we require a boundary condition for x_{N+1}. Section 2.2 illustrates how an approximation to the wave speed for the porous media equation can be derived from the mass conservation property [3].

2.2 Wave Speed Derived from Mass Conservation

We now present a derivation of a numerical approximation to the speed of the moving front resulting from the non-linear diffusion equation. By conservation of mass we have that

$$\frac{\partial}{\partial t}\int_0^{x_{N+1}(t)} u dx = 0$$

where $x_{N+1}(t)$ is the position of the front at time t.

Expanding the integral we have that

$$\int_0^{x_{N+1}(t)} u_t dx + u(x_{N+1}(t))\dot{x}_{N+1} = 0.$$

Obviously there is no derivative on the node at $x = 0$ since this is the centre of mass, which remains constant with time. Now substituting in the diffusion equation into the integral on the left-hand side we have that

$$\int_0^{x_{N+1}(t)} (u^m u_x)_x dx + u(x_{N+1}(t))\dot{x}_{N+1} = 0$$

giving

$$u^m u_x|_{x=x_{N+1}} + u(x_{N+1}(t))\dot{x}_{N+1} = 0$$

since u_x is zero at $x = 0$ by conservation. Rearranging gives

$$\dot{x}_{N+1} = u^{m-1} u_x|_{x=x_{N+1}}. \tag{5}$$

Although $u = 0$ at $x = x_{N+1}$, u_x is unbounded, yielding a finite, non-zero front speed. Taking the limit as $u \to 0$, i.e.

$$\dot{x}_{N+1} = \lim_{u \to 0}(u^{m-1} u_x),$$

we have the approximation

$$\dot{x}_f \approx u_{N-\frac{1}{2}}^{m-1} \frac{u_{N-1}}{x_N - x_{N-1}}$$

2.3 Mesh Refinement

In contrast to MMPDEs derived from an equidistribution principle, there is no reason why all the computational cells should contain equal amounts of the total mass. By using different distributions of mass we can generate a higher resolution adaptive grid near geometrically complicated features of the solution. In the work presented all the initial grids are generated by equidistributing the mass over the grid [4]. However for higher values of m we can do some further subdivision such that the masses become progressively smaller near the front at x_{N+1}. This can be done very simply by setting a grid tolerance parameter *gtol* and then repeatedly subdividing the last computational cell (using the equidistribution idea) until the mass in the last cell is less than or equal to *gtol*. Currently the mesh has only been subdivided at the initial state of the problem, but there is no reason why this process cannot be introduced at any point during the solution of the problem.

2.4 Inverse Polynomial Approximations

An alternative idea to using a refined mesh near steep fronts is to approximate the solution in the last cell by an appropriate nonlinear function. The motivation for such an approach is that by choosing an appropriate form the solution will satisfy the properties required for a finite wave speed (see equation (5)), requiring the value of $u^{m-1}u_x$ to be finite inside the last cell. We now present two such approximations to u in this region and present results compared with those produced on a refined mesh.

2.4.1 Approximation 1

We approximate u In the region (x_N, x_{N+1}) by

$$u_{c1}(x) = u_N \left(\frac{x_N - x}{x_{N+1} - x_N} \right)^{\frac{1}{m}} \tag{6}$$

where u_N is the approximate value of u at $x = x_N$. Furthermore the linear approximation to the mass in the last cell is replaced by the integral of $u_{c1}(x)$, so that

$$C_{N+\frac{1}{2}} = \int_{x_N}^{x_{N+1}} u_{c1}(x)dx = mu_N \frac{x_{N+1} - x_N}{m+1}$$

Using the expression for $C_{N+\frac{1}{2}}$ we can find the value of u_N which retains this mass under this approximation of the solution. So we have that

$$u_{N-1} = \frac{C_{N+\frac{1}{2}}(m+1)}{m(x_{N+1} - x_N)}$$

The wave speed is then approximated using equations (5) and (6).

2.5 Approximation 2

We can 'upgrade' the polynomial expression (6) above by adding a parameter p which will allow us to use information regarding the derivative of u at $x = x_N$. We now choose u such that

$$u_{c2}(x) = A(x_N - x)^{\frac{1}{m}} + B(x_N - x)^p \tag{7}$$

The constants A and B are chosen such that $u_{c2}(x_N) = u_N$ and $u'_{c2}(x_N) = \frac{u_N - u_{N-1}}{x_N - x_{N-1}}$. As before

$$C_{N+\frac{1}{2}} = \int_{x_N}^{x_{N+1}} u_{c2}(x)dx = \frac{Am(x_{N+1} - x_N)^{\frac{m+1}{m}}}{m+1} + \frac{B(x_{N+1} - x_N)^{p+1}}{p+1}$$

As with the previous approximation the value of u_{N-1}, u_N and hence $\dot{x}_{N+1}$ can be found using the various froms of C. It is worth noting that here $C_{N=\frac{1}{2}}$ is still approximated linearly as before. The details are omitted.

3 Numerical Results

We can compare our numerical results for various values of m against an analytical solution. This solution comes from Murray [1] and represents the behaviour of an insect dispersal model, which predicts how a set quantity of insects Q released at $x = 0$ diffuses out symmetrically about $x = 0$ with time. We take initial conditions at a small arbitrary time *tstart* and then compare the approximate and analytical solutions at various times for values of $x \geq 0$.

In all the results presented here we have used the NAG subroutine D02EJF, which uses a backwards differentiation formula, to solve the stiff system of ODE's (3) with u given by 4). We consider three different values of m.

First, we try the simple case when $m = 1$. Due to the simple power on the diffusion co-efficient, the solution does not form a steep front at the moving boundary, and hence no grid refinement is required. A relatively small number of nodes can be used to get an adequate solution as shown in Figure 1, in this case $N = 20$. The approximate solution values are denoted by the crosses whilst the Murray solution is represented by the solid line. On the left hand side of Figure 6 we can see the convergence of the solution at $x = 0$ at$t = 1$ as N is increased. The graph is plotted as the logarithms of N and the error and from the gradient of the curve we can say that the algorithm has an order of accuracy of 2. The right hand side of the figure shows the trajectories of the nodes, from this you can see the movement is to be as expected since the masses are kept constant and equal (≈ 0.0263) with the nodes moving almost uniformly near $x = 0$. The parameters of the analytical solution in this case are $Q = 1$ and $tstart = 0.01$.

Next we show that by using grid refinement we can accurately resolve a steep front near the moving boundary for a larger value of m. Using the grid refinement algorithm as described in Section 2.3, starting with an initial 15 nodes and adding an extra 10 by setting $gridtol = 10^{-4}$, we solve for $m = 3$. The analytical solution parameters are $Q = 2$ and $tstart = 0.05$. Figure 2 shows the numerical solution, again denoted by crosses against the Murray solution shown with the solid line, for various times.

Finally we present results for when $m = 6$, with $Q = 1$ and $tstart = 0.01$. The mesh was formed initially with 10 nodes and then 9 extra nodes were added by setting $gridtol = 10^{-4}$. This is a much more severe problem and

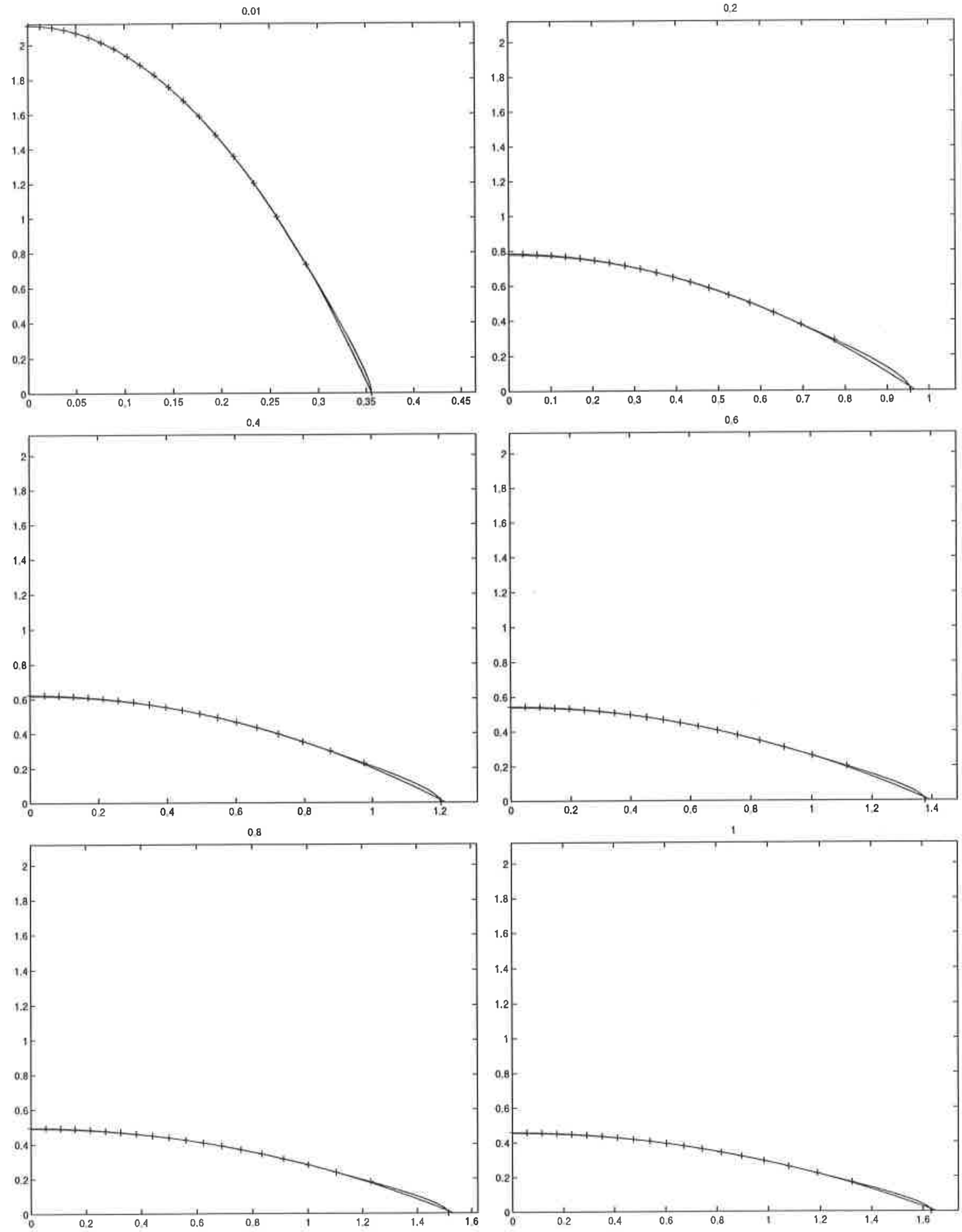

Figure 1: Approximate and reference solutions for $m = 1$

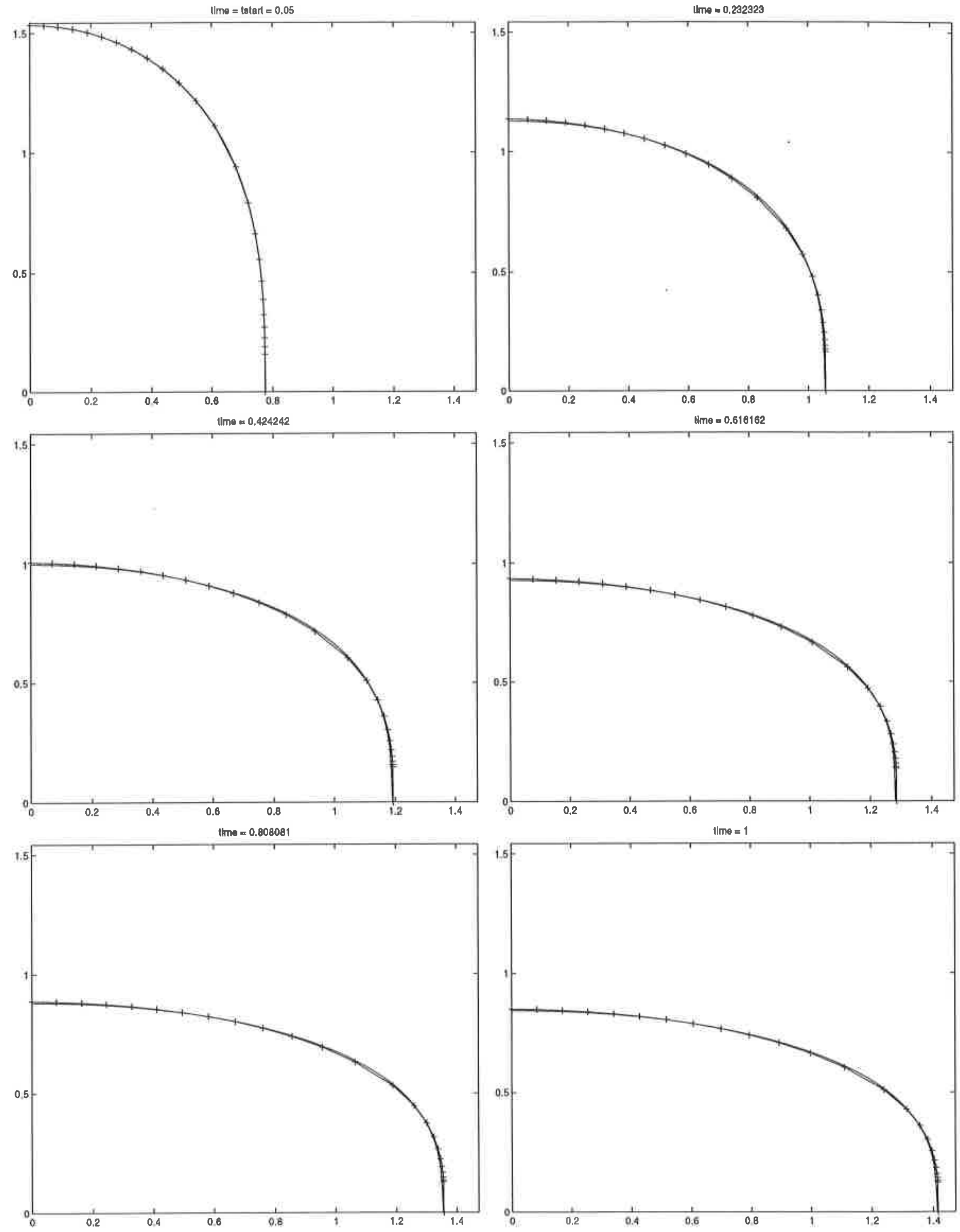

Figure 2: Approximate and reference solutions for $m = 3$

the method seems able to manage the exaggerated steep front quite nicely without using a great number of nodes. Figure 3 shows the results.

Figures (4) and (5) show results using the inverse polynomial approximations outlined in sections 2.4.1 and 2.5 (where $p = \frac{1}{2}$). These are generated using the same solution parameters as used for the results (except that $tstart = 0.01$) shown in figure (2) to give us a direct comparason with the results produced when using a refined mesh. It is obvious that the polynomial approximations do not achieve the desired results. It is apparent that the wave speed has not been accurately approximated since the position of the front is always found trailing behind its exact position. It is worth noting that the inclusion of the parameter p does give a smoother approximation at $x = x_N$ but does little to improve the solution as a whole. Experiments were made changing the value of the parameter p but with little or no significant improvement in the results.

Figure 7 shows the trajectories of the nodes in the cases of $m = 3$ (left hand side) and $m = 6$ (right hand side). Notice that because of the way the nodes are moved, nodes near the front tend to move away from this area as time progresses. It would be expected then that as time progresses the solution would gain accuracy near $x = 0$ and lose accuracy nearer the front. However Figure 8 doesn't agree. Although the relative error at $x = 0$ is significantly greater than at a point measured near the front, the relative error near the front actually decreases with time, whilst the relative error at the centre of mass varies only slightly with time. These results are from the case when $m = 3$.

Finally quantive measures of the error, by means of the $L1$ and $L2$ norms are shown in tables (3-3).

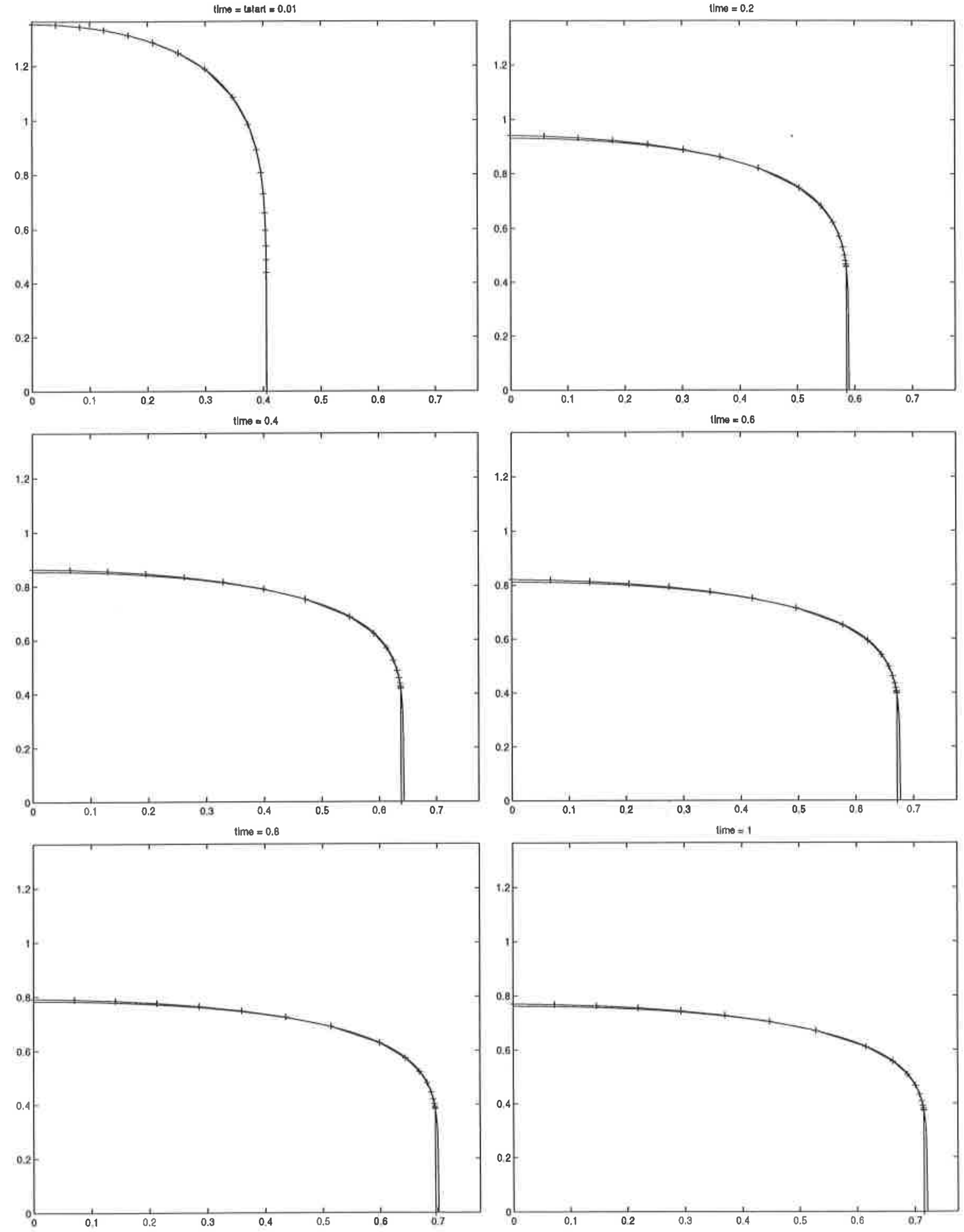

Figure 3: Approximate and reference solutions for $m = 6$

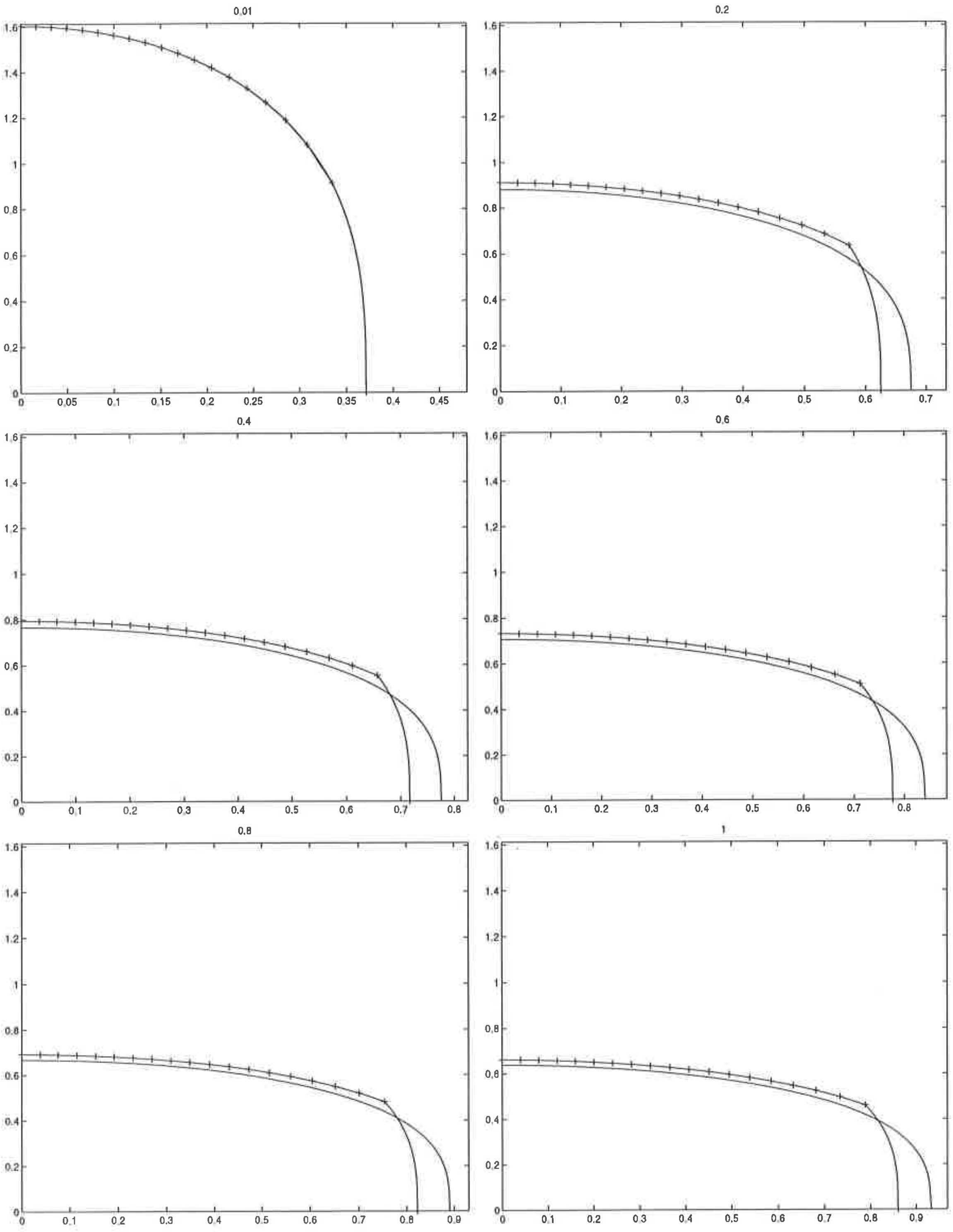

Figure 4: Approximate and reference solutions for $m = 3$ using approximation 1

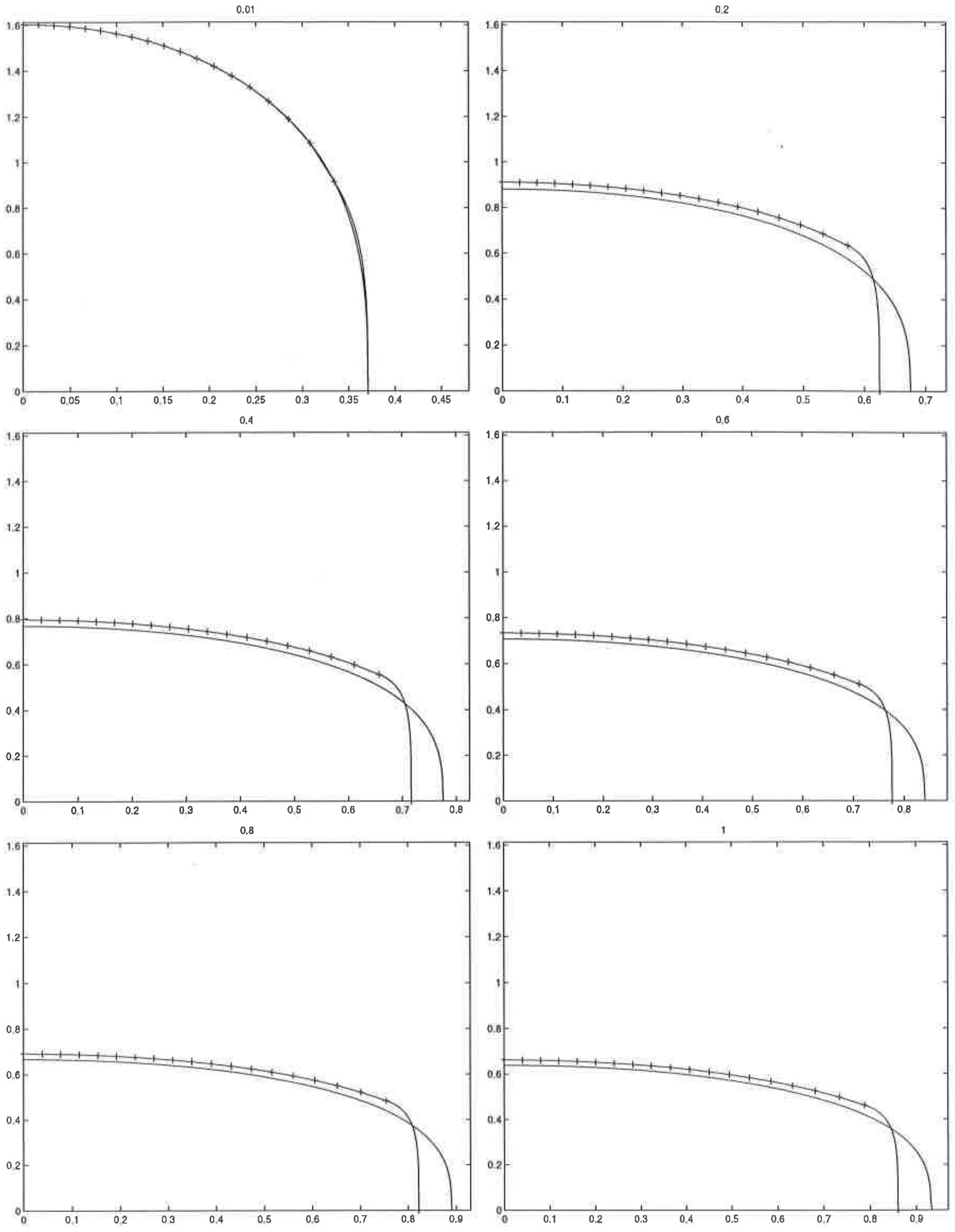

Figure 5: Approximate and reference solutions for $m = 3$ using approximation 2

Time	0.2	0.4	0.6	0.8	1.0
L1 Norm	0.0900	0.0725	0.0636	0.0580	0.0539
L2 Norm	0.0272	0.0219	0.0192	0.0175	0.0163

Table 1: Errors for m=1 (See Fig 1)

Time	0.2323	0.4242	0.6162	0.8081	1.0
L1 Norm	1.0259	0.9075	0.8414	0.7966	0.7631
L2 Norm	0.3939	0.3479	0.3227	0.3056	0.2928

Table 2: Errors for m=3 (See Fig 2)

Time	0.2	0.4	0.6	0.8	1.0
L1 Norm	0.5055	0.4730	0.4528	0.4383	0.4272
L2 Norm	0.4495	0.4139	0.3940	0.3804	0.3701

Table 3: Errors for m=6 (See Fig 3)

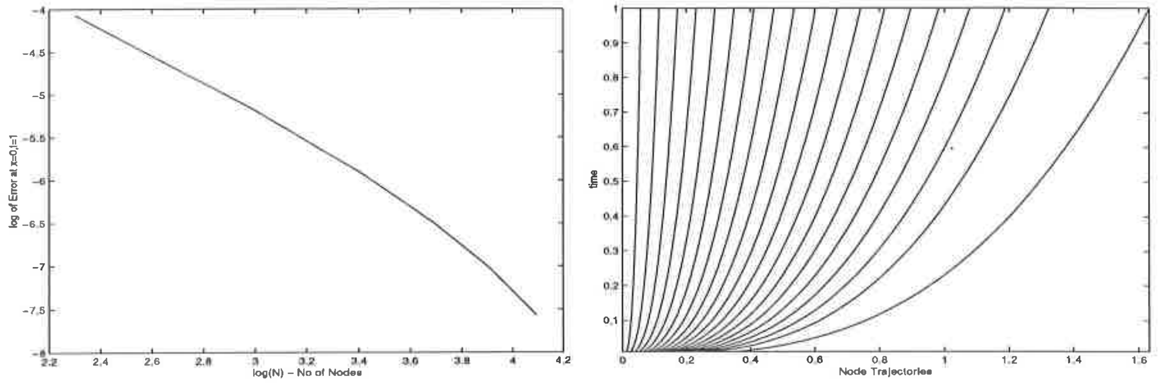

Figure 6: Error convergence and Node trajectories for $m = 1$

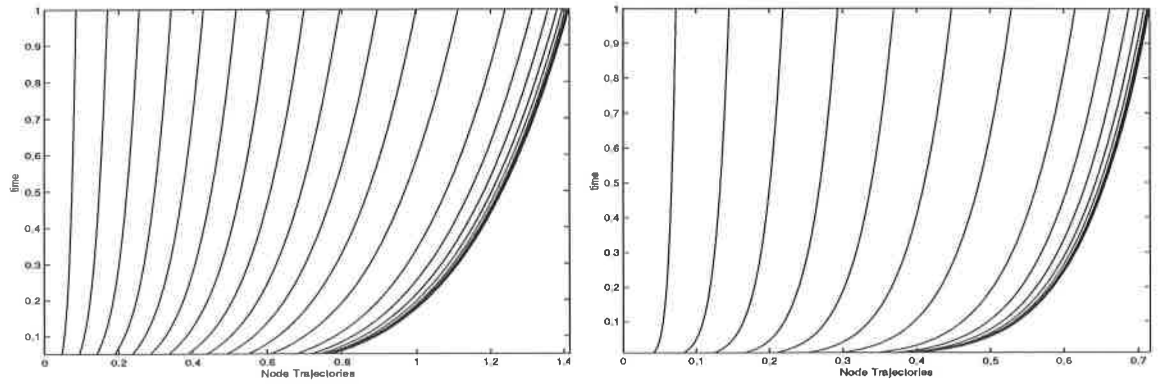

Figure 7: Node trajectories for $m = 3$ and $m = 6$

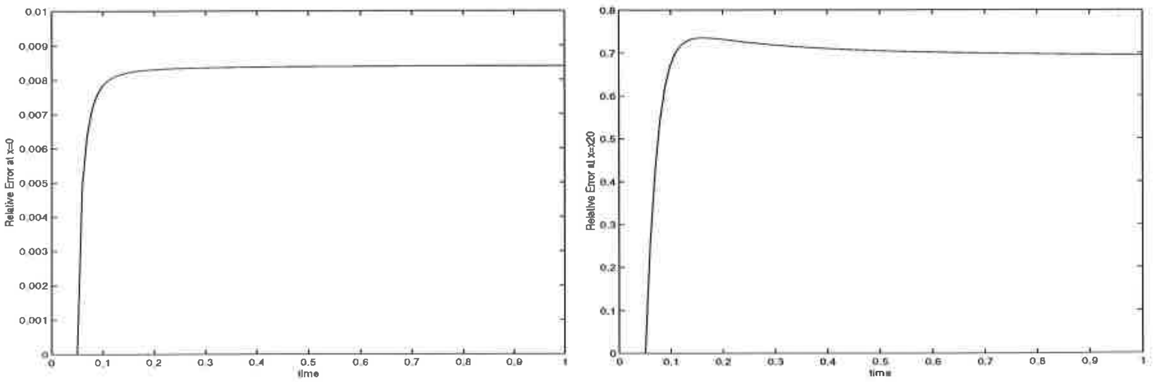

Figure 8: Relative error with time at $x = 0$ and $x = x_{20}(t)$.

4 Conclusions and Further Work

The results presented in Section 3 show that the method can give good quality solutions to the porous media equation for various values of m by solving only one system of bi-diagonal ODE's. It is also shown that the steep front which is a feature of the porous media equation can be resolved by using a simple grid refinement algorithm at the initial state of the solution, although attempts at replacing a refined mesh with a polynomial approximation of u in the last cell did not prove successful.

However, it must be noted that this approach is only possible due to the two important features of the PME noted in Section 2. The conservation of mass means there are no terms involving integral of mass differentiated with respect to time, which could be hard to discretise accurately. Also the fact that $u = 0$ at the foot of the front allows the values of u to be easily expressed in terms of the node positions, so giving only one system of equation to solve.

These two points open interesting avenues for further research. It is easy to see that if either of these two resttictions could be overcome then the method would applicable to several different areas. For example, if an accurate way of expressing the time-derivatives of integrals could be introduced then the method could be applied to reaction-diffusion problems such as the Modified-Fisher-Equation. Alternatively if an algorithm for the expression of u in terms of x was introduced when u is unknown at the boundary, then other conservation laws could possibly also be modelled using ideas from this work.

Other ideas for further work include extending the method to higher dimensions and finding other ways of generating and refining the grid, either initially or as required as the solution evolves.

References

[1] Murray J.D, Mathematical Biology, Springer 1993, p238-241

[2] Weizang Hauang, Yuhe Ren and R.D.Russell. Moving Mesh Partial Differential Equations based on the Equidistribution Principle. SIAM Journal of Numerical Analysis, Vol 31 No. 3 pp. 709-730, June 1994

[3] C.J.Budd, G.J.Collins, W.Z.Huang and R.D.Russell. Self-similar Numerical Solutions of the Porous Medium Equation using Moving Mesh Methods. Proceedings of the Royal Society, 357, pp1047-1077, 1999

[4] Baines M.J. Grid Adaptation via node movement. Applied Numerical Mathematics 26, pp77-96, 1998