A Moving Mesh Finite Element Method for High Order Non-Linear Di□usion Problems

Bonhi Bhattachar a

August 21, 2006

Abstract

In this dissertation a moving mesh method, based on a conservation of massract

Contents

Chapter 1

Introduction

Adaptive mesh techniques play an important role in improving existing finite element methods for the numerical solution of partial differential equations, by concentrating mesh points in areas of interest. Such areas exist when large variations occur in the solution, which include moving boundaries, shocks and blow up.

the possible behaviours that can arise as the boundary moves. We also discuss conjectures and results in the existing literature about the parameters for which these behaviours occur. We then go on in Chapter Eight to provide numerical results of the investigation of the moving front and compare these in the fourth order case to the results obtained by asymptotic analyses. For the sixth order case we investigate whether our results support existing conjectures. Further since analytic error analysis is not possible for moving mesh methods, we investigte whether the results obtained on a moving mesh match results obtained using a fixed mesh method for which error analysis exists. Finally in Chapter Nine, we present our conclusions, and discuss limitations of the model and possible improvements, as areas of possible further work.

Chapter 2

Non-Linear Diffusion

In this chapter we present some applications of nonlinear diffusion. It is hoped that this chapter will illustrate the need for efficient numerical solutions of the nonlinear diffusion equations, by providing examples of the wide

material through which the di□usion is occurring.

2.1 The Porous Medium Equation

The Porous Medium equation, which is characterised by $p = u$ in (2.0.1),

stand pathological conditions related to these materials, one example being tumour growth.

In the first stage of tumour development, tumours are ordinarily avascu-

When $n = 1$, the equation models flow in a Hele-Shaw cell. In a Hele-

Another possible application has been postulated in modelling approaches to the wrinkling process when a compressively strained elastic film is bonded to a viscous layer [19]. A specific example is the wrinkles formed upon annealing of a compressively strained silicon germanium alloy film bonded to a silicon substrate covered with a glass layer. This wrinkling has also been seen in thin metal films on polymers and may have uses in optical devices such as di raction gratings.

Chapter 3

Moving Mesh Methods

In this chapter, we look at moving mesh methods in the context of solving time-dependent partial differential equations, examples of which were given in Chapter Two. These equations have solutions with features which evolve over time, and an adaptive numerical method is needed if these features are to be resolved accurately.

Moving mesh methods require the generation of a mapping from a regular domain in the computational space, Ω_c, to an irregular domain in the physical space, Ω. The physical domain can be covered with a computational mesh by connecting points in the physical space to corresponding discrete points in the computational space. Let x denote the physical co-ordinate in the domain Ω, and ξ, the computational co-ordinate in the domain Ω_c . Then this mapping can be defined as a one-to-one transformation described by

$$x = x(\xi, t) \tag{3.0.1}$$

which maps points in the computational space at time t, onto the physical space.

Many approaches have been developed for generating moving adaptive meshes, and most can be classified as either location based or velocity based methods. Location based methods are so called because they seek to directly

control the location of the mesh points, an example being the variational method, which determines the mapping from the computational to the physical domain by minimizing a variational form, or functional [14].

3.1 Velocity Based Methods

Velocity based methods are considered in greater detail, as the moving mesh method used in this dissertation is an example of this group. Velocity based methods compute a mesh velocity, $v = x_t$, using a Lagrangian like formulation. The mesh point location can be found from this velocity using time integration.

We can consider a classical Lagrangian method, as in fluid dynamics, where the Lagrangian co-ordinates form a co-ordinate system which follows fluid particles. Then if $u(x, t)$ represents the velocity of the fluid, ξ represents the reference co-ordinate of a fluid particle, and $x(\xi, t)$, the position of the particle at time t. The particle then evolves with

$$\frac{\partial x}{\partial t} = u$$

We can then consider a velocity based method in two stages. First a mapping from the computational to the physical domain is generated. Once a suitable mapping has been determined, a motion is induced on the mesh by considering the rate of change in time of this mapping, which generates the moving mesh equations. These equations give each computational node an associated velocity, which can then be used to advance the mesh forward in time.

One of the most commonly used approaches to generate an irregular mapping is the equidistribution principle introduced by De Boor, [11]. Here, mesh points are chosen so that some measure is equally distributed over each computational cell of the mesh. This measure is user defined, known as the

monitor function, and is a positive function of the solution u and/or its derivatives.

In terms of the mapping outlined in equation (3.0.1), the equidistribution principle can be written as

$$\int_{x(\xi_i)}^{x(\xi_{i+1})} M\,dx = \frac{1}{N}\int_0^1 M\,dx \tag{3.1.1}$$

for M, some monitor function, with the form

$$M = M(x, u, u_x, u_{xx} \ldots) \tag{3.1.2}$$

Possibe monitor functions include $M = 1$, which produces a uniform grid, and the popular arc-length monitor, where

$$M = \left(1 + \left(\frac{du}{dx}\right)^2\right)^{1/2} \tag{3.1.3}$$

Other choices exist for the monitor function, which are not discussed here, but their aim is to create a grid with low resolutions where there is low activity in the solution, and increased resolution where there is greater change in the solution. Many moving mesh methods have been derived using a monitor function to control mesh movement and other methods have also been developed which are related in some way to these methods, see for example [14], [15], [20].

Recently, there has been much work centered on the use of moving mesh methods applied to PDEs which possess scale invariant behaviour and self-similar solutions [13]. Here it is proposed that the monitor function used should, in some manner, also be scale invariant when applied to PDEs which are scale invariant. In [5], Baines, Hubbard and Jimack, used the mass monitor function $M = u$, when constructing a moving mesh method to solve the nonlinear diffusion equations, the motivation being that these equations are mass conserving.

In the next chapter, a moving mesh finite element algorithm, as used in [5] is derived for the adaptive solution of nonlinear di□usion equations with moving boundaries in one dimension.

Chapter 4

$$\frac{d}{dt}\int_{x_0(t)}^{x_N(t)} u(t) = \int_{x_0(t)}^{x_N(t)} \frac{\partial u}{\partial t} dx + \frac{d}{dt} x_N(t) \quad dd$$

$d^{(t}$

Using Leibnitz' rule in (4.3), gives

$$\int_{x_{i-1}(t)}^{x_i(t)} \frac{\partial u}{\partial t}dx + \frac{d}{dt}x_i(t)\frac{d}{dx_i(t)}\int_{x_{i-1}(t)}^{x_i(t)} u(t)dx + \frac{d}{dt}x_{i-1}(t)\frac{d}{dx_{i-1}(t)}\int_{x_{i-1}(t)}^{x_i(t)} udx = 0$$

Substitution from (4.1.1) leads to

$$\int u^n \frac{\partial^{2m+1}u}{\partial x^{2m+1}} + \dot{x}u\Big|_{x_0(t)}^{x_i(t)} = 0 \tag{4.1.7}$$

Further, since $u = 0$ at $x = x_0(t)$,

$$u^n \frac{\partial^{2m+1}u}{\partial x^{2m+1}} + \dot{x}u = 0 \tag{4.1.8}$$

at $x = x_i(t)$ i. Hence

$$\dot{x} = -u^{n-1}\frac{\partial^{2m+1}u}{\partial x^{2m+1}} \tag{4.1.9}$$

except when $u = 0$. By continuity, (4.1.9) also holds as $u \to 0$.

This gives the velocity of a general mesh point, from which the new position of the point can be calculated by time integration. The new solution u may then be recovered from the consery.Bf30.90.d[(ma04.13.9390TeΩ8(ol)1(ut8se)-1(rp4(e)]TJ/F38)-31pl

$$\int_{x_{i-1}(t)}^{x_{i+1}(t)} u\frac{\partial \varphi_i}{\partial x}\dot{x}dx = -\int_{x_{i-1}(t)}^{x_{i+1}(t)} u^n \frac{\partial \varphi_i}{\partial x}\frac{\partial^{2m+1}u}{\partial x^{2m+1}}dx$$
$$= -\int_{x_{i-1}(t)}^{x_{i+1}(t)} u^n \frac{\partial \varphi_i}{\partial x}\frac{\partial q}{\partial x}dx \qquad (4.2.5)$$

for all interior nodes, where, for example, in the sixth order case, $2m+1 = 5$

$$q = -\frac{\partial^2 p}{\partial^2 x} \qquad \text{and} \qquad p = -\frac{\partial^2 u}{\partial^2 x} \qquad (4.2.6)$$

For any u, p is obtained from the weak form of the equation

$$p = -\frac{\partial^2 u}{\partial^2 x} \qquad (4.2.7)$$

$$\int_{x_{i-1}}^{x_{i+1}} \varphi_i p dx = -\int_{x_i+1}^{x_i-1} \varphi_i \frac{\partial^2 u}{\partial x^2}$$
$$= -\varphi_i \frac{\partial u}{\partial x}\Big|_{x_{i-1}}^{x_{i+1}} + \int_{x_i+1}^{x_i-1} \frac{\partial \varphi_i}{\partial x}\frac{\partial u}{\partial x}dx \qquad (4.2.8)$$

and using the boundary condition that $\varphi = 0$ at $x_{i+}, xi - 1$,

$$\int_{x_i+1}^{x_i-1} \frac{\partial \varphi_i}{\partial x}\frac{\partial u}{\partial x}dx = \int_{x_{i-1}}^{x_{i+1}} \varphi_i p dx \qquad (4.2.9)$$

Now the finite element approximations are expanded in terms of the basis functions φ_i to give

$$u = \sum_{j=1}^{N} u_j \varphi_j, \qquad p = \sum_{j=1}^{N} p_j \varphi_j, \qquad q = \sum_{j=1}^{N} q_j \varphi_j \qquad (4.2.10)$$

These forms are substituted in to (4.2.6). Then

$$K\underline{u} = M\underline{p} \qquad (4.2.11)$$

where both K, is the standard stiffness matrix with entries of the form

$$K_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} dx \tag{4.2.12}$$

and M is the standard mass matrix with entries of the form

$$M_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \varphi_i(x) \varphi_j(x) dx \tag{4.2.13}$$

Both matrices are tridiagonal, and the system is solved using a direct tridiagonal solver. In a similar fashion, q can be obtained from p by solving the mass matrix system

$$K\underline{p} = M\underline{q} \tag{4.2.14}$$

Once q has been obtained, it is used in equation (4.2.5)

$$\int_{x_{i-1}(t)}^{x_{i+1}(t)} u \frac{\partial \varphi_i}{\partial x} \dot{x} dx = - \int_{x_{i-1}(t)}^{x_{i+1}(t)} u^n \frac{\partial \varphi_i}{\partial x} \frac{\partial q}{\partial x} dx \tag{4.2.15}$$

The matrix form of this equation is difficult to solve in practice because the resulting system creates a non symmetric matrix. It is therefore convenient to introduce a velocity potential ψ, such that

$$\dot{x} = \frac{d\psi}{dx} \tag{4.2.16}$$

where ψ is piecewise linear, and the expanded finite element approximation of ψ is given by

$$\psi = \sum_{j=1}^{N} \psi_j \varphi_j \tag{4.2.17}$$

This results in (4.2.14) becoming the stiffness matrix system

$$K(u)\underline{\dot{x}} = -K(u^n)\underline{q} \tag{4.2.18}$$

where $K(u)$ and $K(u^n)$ are weighted stiffness matrices. $K(u)$ has entries of the form

$$K_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} u \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} dx \tag{4.2.19}$$

and $K(u^n)$ has entries of the form

$$K_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} u^n \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} dx \tag{4.2.20}$$

for interior nodes. The integrals are evaluated within these intervals by numerical integration using Simpson's rule. $\dot{x}$ is then recovered from ψ by constructing a finite element formulation of equation(4.2.15),

$$\int_{x_{i-1}}^{x_{i+1}} \varphi_i \dot{x} dx = \int_{x_{i-1}}^{x_{i+1}} \varphi_i \frac{d\psi}{dx} dx \tag{4.2.21}$$

which results in the mass matrix system

$$M\underline{\dot{x}} = B\underline{\psi} \tag{4.2.22}$$

where B has entries of the form

$$B_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \varphi_i \frac{\partial \varphi_j}{\partial x} dx \tag{4.2.23}$$

Evaluations of these integrals results in an antisymmetric structure for B as ilustrated below.

$$\underline{B} = \begin{bmatrix} 0 & -\frac{1}{2} & 0 & 0 & \dots \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & \dots \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & \ddots \\ \vdots & 0 & \frac{1}{2} & 0 & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

showing how $\frac{d\psi}{dx}$ is constructed from ψ

i

Once the new mesh has been obtained, the new solution on this mesh is found by solving the conservation of mass equation (4.2.1), with the new positions of the nodes. This solution is equivalent to solving the mass matrix system

$$M\underline{u} = \underline{c} \tag{4.2.24}$$

where c is given by

$$c = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \varphi_i u_0 dx \tag{4.2.25}$$

for initial data u_0. The integral is evaluated using a Gaussian Quadrature rule, and remains constant, so preserving the distributed mass in an interval.

However, overwriting the boundary conditions $u = 0$ at $(x_0(t), x_N(t))$ will result in the loss of mass conservation, since the first and last equations of (), will in general no longer be satisfied. To overcome this a mass conserved version of () is used, in which the first equation of () is added to the second, and the last equation to the last but one, prior to overwriting these conditions.

4.2.1 Timestepping

The new node positions can be found using any timestepping algorithm. In practice though, implementation of implicit schemes is a complicated

(6.0.3)). This has posed a problem here for the C++ code written, due to memory restrictions allowing only a certain number of timesteps to be run; such a small timestep means the behaviour of the solution over longer time scales cannot be determined. To overcome this problem, a version of the same method implemented in FORTRAN from the work of Baines, Hubbard, and Jimack is used to run the method for larger final times.

Chapter 5

Self-Similar Solutions

Any numerical method used to solve the nonlinear diffusion problem would need to have the property that the numerical solution will eventually converge to the true solution. A class of true solutions called similarity solutions are used here to verify the numerical results obtained. Once the reults have been evaluated, the method can then be used to investigate initial data for which no analytic solution is known.

5.1 Scale Invariance

A symmetry of the partial differential equation,

$$u_t = f(x, u, u_x, u_{xx}, ...) \tag{5.1.1}$$

is defined to be a transformation of (x, u, t) which leaves the underlying PDE unchanged. This can be considered as a transformation of (x, u, t) to $(\bar{x}, \bar{u}, \bar{t})$ such that

$$\bar{x} = \bar{x}(x, u, t), \bar{u} = \bar{u}(x, u, t), \bar{t} = \bar{t}(x, u, t) \tag{5.1.2}$$

so that the equation satisfied by (x, u, t), is also satisfied by $(\bar{x}, \bar{u}, \bar{t})$. To construct similarity solutions for the PDE, a subclass of symmetries known as

$$(2m + 1)\beta - n\gamma = 1 \tag{5.1.9}$$

giving a class of symmetries

5.2 Self Similar Solutions

The boundary conditions $u = 0$ at $a(t) = b(t)$ for the general equation induce conservation of mass(see (4.1.4)), which becomes

$$\int_{a(t)}^{b(t)} u\,dx = \int_{a(t)}^{b(t)} \lambda^{\gamma}\bar{u}\,d(\lambda^{\beta}\bar{x}) = \text{constant in time} \tag{5.2.1}$$

$$\lambda^{\gamma+\beta} \int_{a(t)}^{b(t)} \bar{u}\,d(\bar{x}) = \text{constant in time} \tag{5.2.2}$$

and scale invariance holds for the nonlinear diffusion equations, provided

$$\gamma + \beta = 0 \tag{5.2.3}$$

Then it can be seen, by solving the simultaneous equations (5.1.9), and (5.2.3), that the only self similar solution for the PDE under these conditions, has

$$\gamma = \frac{-1}{n + (2m + 2)} \qquad \beta = \frac{1}{n + (2m + 2)} \tag{5.2.4}$$

and the behaviour of the solution in the transformed coordinates will possess the conservation property of the solution in the non-transformed space.

A similarity solution of the PDE is defined as a solution of the PDE which is invariant under the action of the scaling transformations described in (5.1.3). From [28], it is known that the nonlinear diffusion equations admit a family of self similar solutions of the form,

$$u(x, t) = t^{\gamma}\varphi\left(\frac{x}{t^{\beta}}\right) \tag{5.2.5}$$

5.2.1 A Fourth Order Self Similar Solution

To construct the similarity solution,for example, for the fourth order case, similarity variables y and v are introduced such that

$$y = \frac{x}{t^\beta} = \frac{\bar{x}}{t^\beta} \qquad v = \frac{u}{t^\gamma} = \frac{\bar{u}}{t^\gamma} \tag{5.2.6}$$

where y and v are independent of λ and are invariant under (5.1.3). By equation (5.2.4), for the fourth order case, β and γ have the values.

$$\gamma = \frac{-1}{n+4} \qquad \beta = \frac{1}{n+4} \tag{5.2.7}$$

A similarity solution is sought of the form $v = f(y)$, by obtaining an ordinary differential equation for v in terms of y.
Transforming the left hand side of (5.1.4) gives

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \frac{\partial}{\partial t}(vt^\gamma) \\
&= t^\gamma \frac{dv}{dt} + v\gamma t^{\gamma-1} \\
&= t^\gamma \frac{dv}{dy}\frac{\partial y}{\partial t} + v\gamma t^{\gamma-1} \\
&= t^\gamma \frac{dv}{dy}\left(\frac{-\beta x}{t^{\beta+1}}\right) + v\gamma t^{\gamma-1} \\
&= -\beta t^{\gamma-1} y \frac{dv}{dy} + v\gamma t^{\gamma-1}
\end{aligned} \tag{5.2.8}$$

To transform the right hand side of (5.1.4) into v and y, first consider

$$\frac{\partial u}{\partial x} = \frac{dy}{dx}\frac{du}{dv}\frac{dv}{dy} \tag{5.2.9}$$

From (5.1.3)

$$\frac{dy}{dx} = t^{-\beta}, \qquad \frac{du}{dv} = t^\gamma \qquad \text{giving} \qquad \frac{\partial u}{\partial x} = t^{\gamma-\beta} v \tag{5.2.10}$$

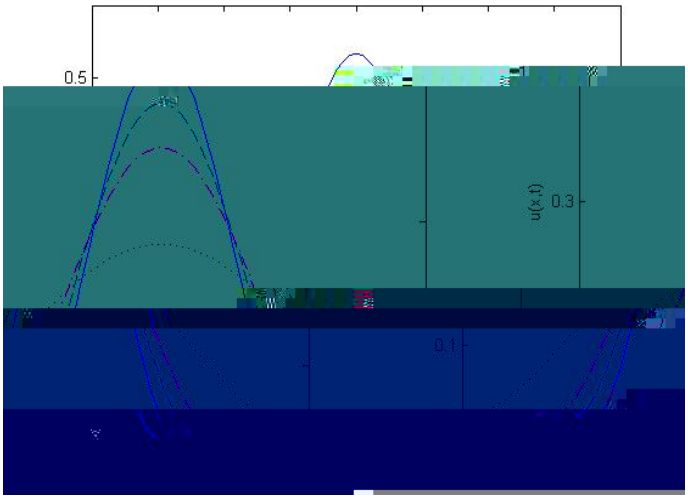
from which,

$$\frac{\partial^2 u}{\partial x^2} = t^{\gamma-2\beta} v \tag{5.2.11}$$

$$\frac{\partial^3 u}{\partial x^3} = t^{\gamma - 3\beta} v \tag{5.2.12}$$

Then

$$\frac{\partial}{\partial x} \square$$


0.5
u(x,t)
0.3
0.1

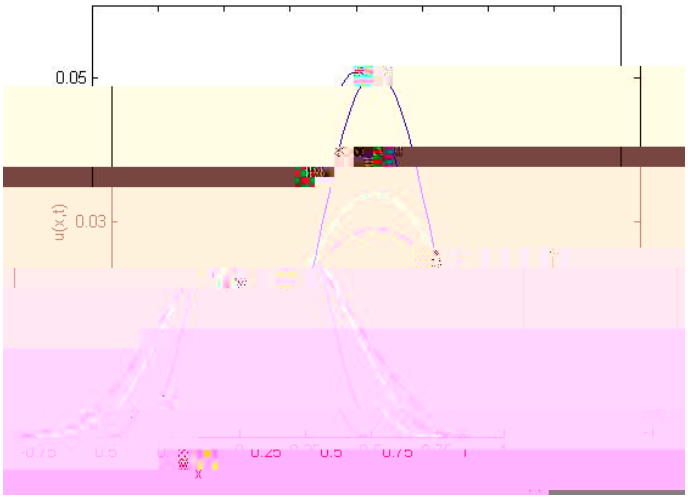


Figure 5.2.2: Distribution of the sixth order self similar solution

The initial data supplied to the program, at time $t = 0$ is given by

$$u(x,0) = \frac{1}{120}\left[4 - 16x^2\right]_+^3 \qquad (5.2.23)$$

5.3 Scale Invariance of the Numerical Method

PDE.

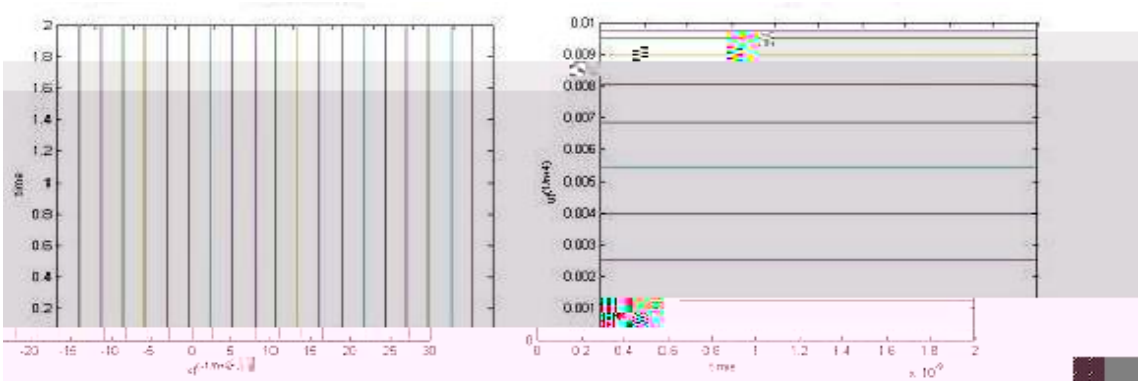


Figure 5.3.1: Invariance of the fourth order solution and mesh for 41 node mesh

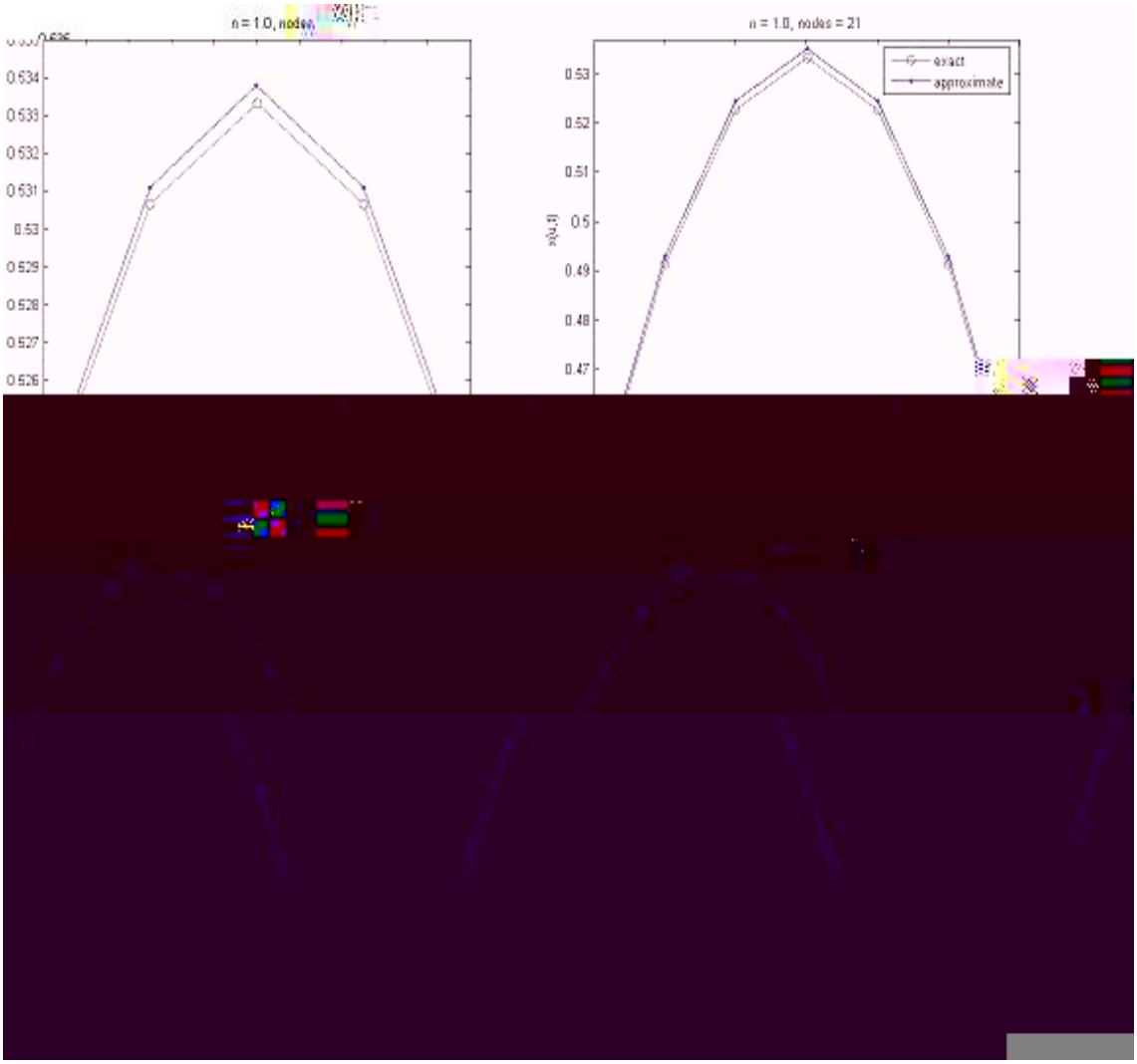


Figure 6.0.1: Convergence of numerical solution to true solution with increasing nodes, t = 0.0005

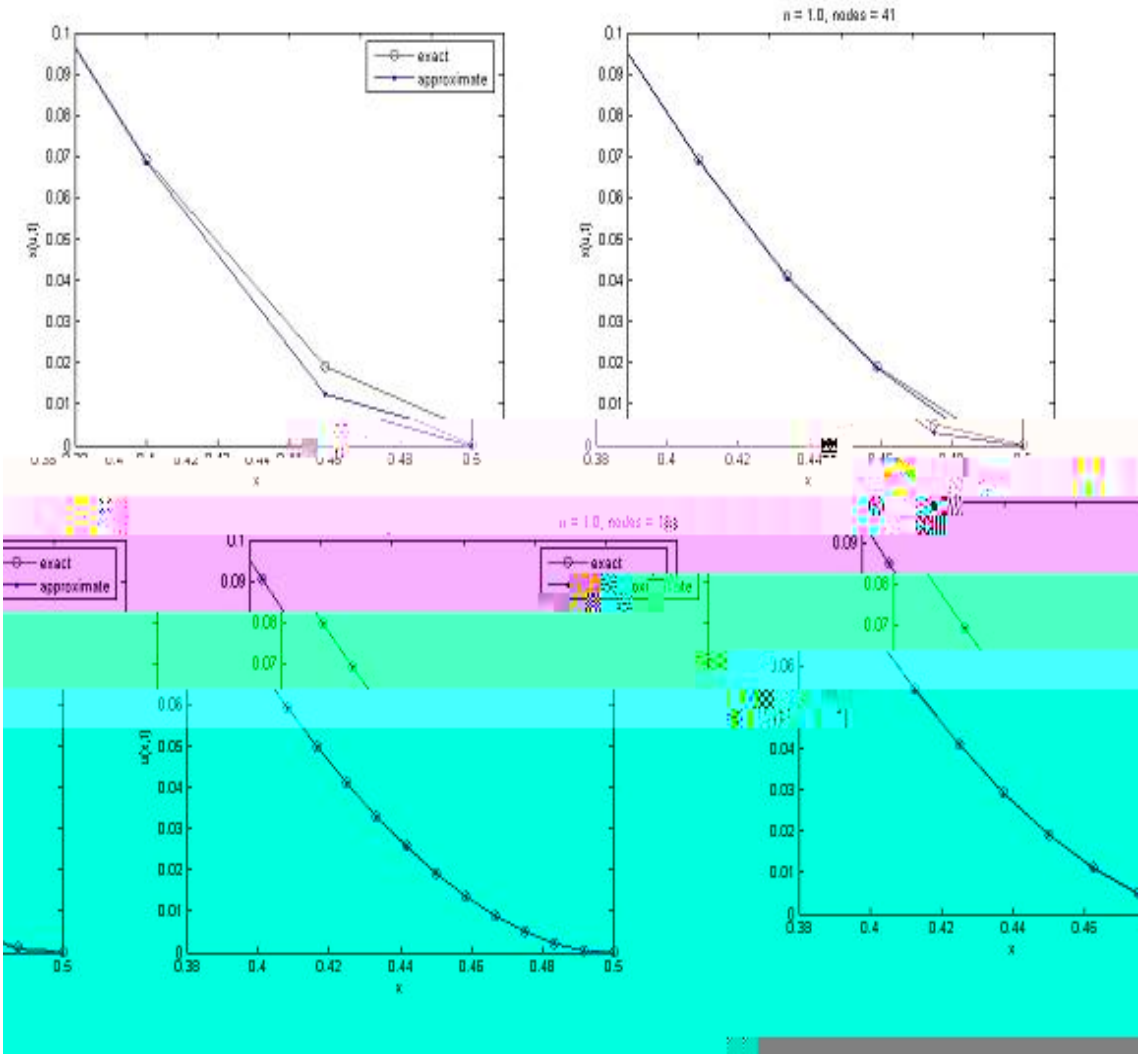


Figure 6.0.2: Resolution of numerical solution at boundary with increasing nodes, t = 0.0005

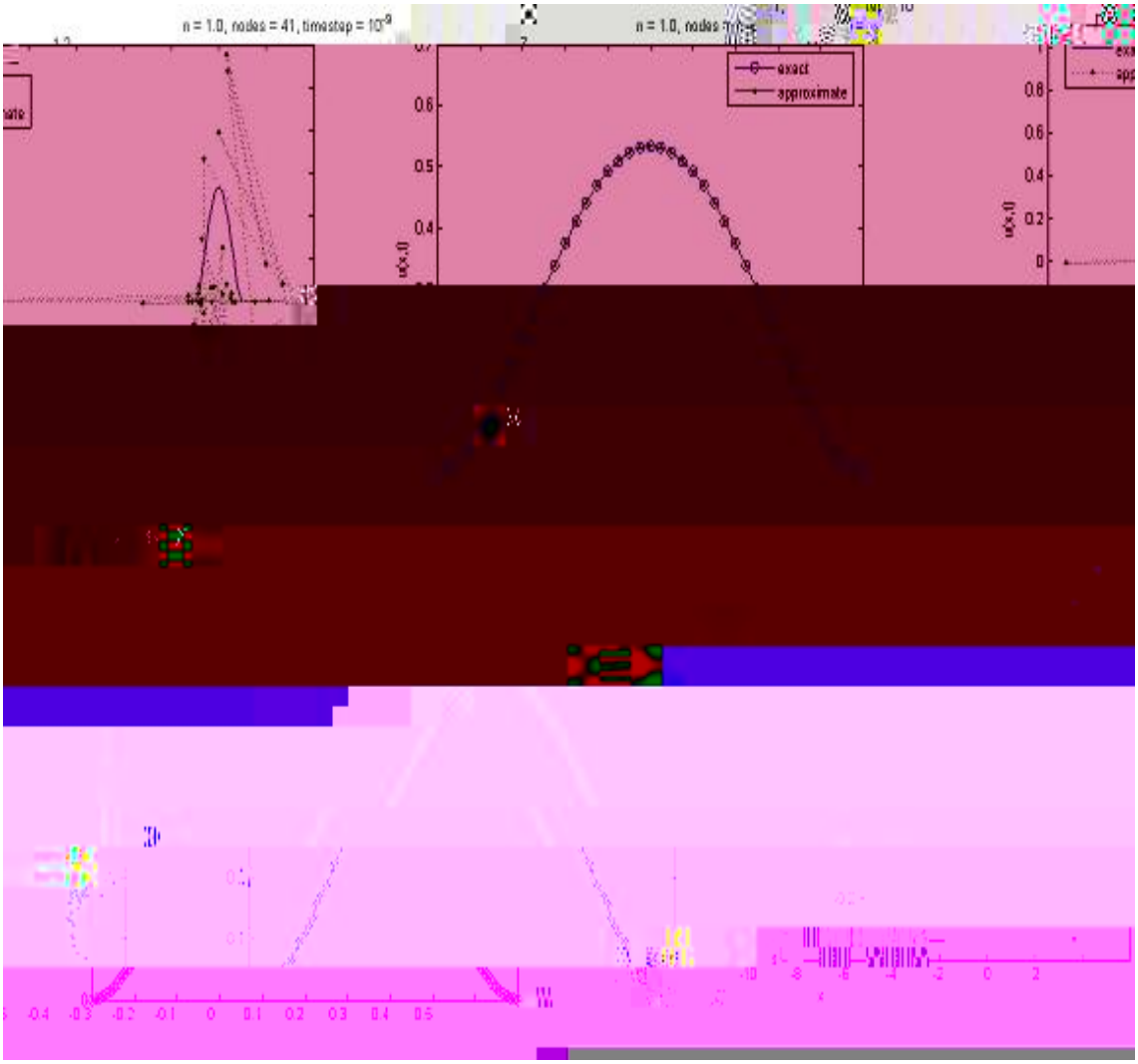


Figure 6.0.3: Timestep dependence on the number of nodes in the mesh

the numerical solution is diffusing to the exact solution with time.

6.1 Fourth Order Results

Dots represent approximate solution, and the line represents the exact solution. Solutions are computed at

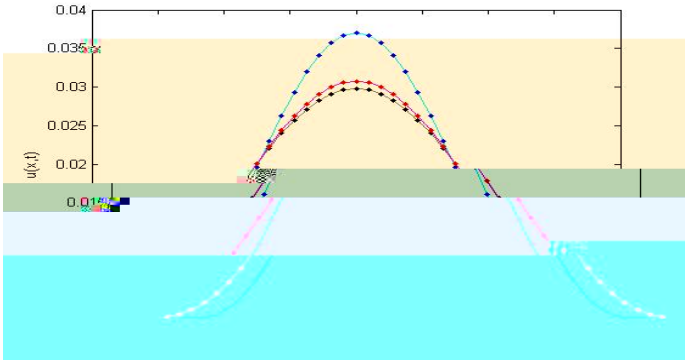


Figure 6.2.1: Exact and approximate solutions for the sixth order problem.

Chapter 7

Behaviour of the Moving Boundary

The occurrence of fronts is an interesting feature of the nonlinear diffusion equations, where the front is the interface between positive values of the solution, and zero values of the solution. This interface can possess three kinds of behaviour. Firstly, the interface can move immediately, and if it does so, it can either retreat or advance. Secondly, a waiting time scenario can occur, where the interface remains stationary for a finite time, and then starts to move. Thirdly the interface can wait forever. Prediction of the behaviour of the moving front can be important in many of the physical applications of the nonlinear diffusion equations outlined in Chapter Two.

These behaviours are dependent on the values of n and α. n is the diffusion coefficient and represents the viscosity of the fluid film. The values of n will influence the speed of the moving front. For example, a large value of n represents a high viscosity of fluid which moves with slower velocity than a low viscosity fluid. α represents the initial contact angle of the interface; the larger the value of α, the shallower the initial contact angle, illustrated in Figure (7.0.1).

Waiting time behaviour occurs when the solution undergoes an initial

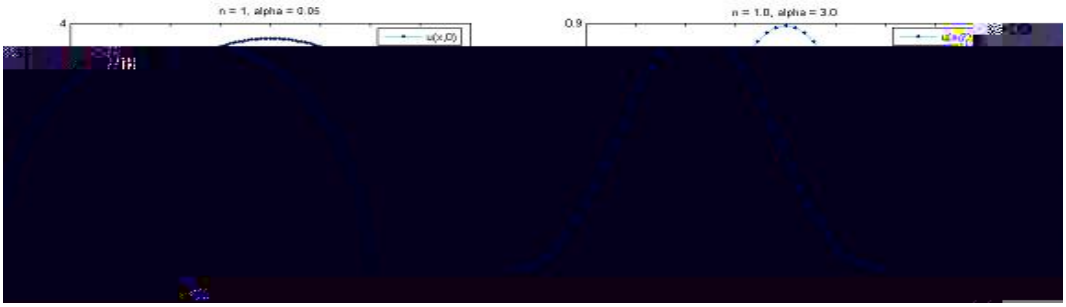


Figure 7.0.1: Change in contact angle for di□erent alpha

redistribution behind the moving front, during which time the contact angle of the boundary readjusts. As it does so the solution waits, until that time when the angle reaches a value for which advancing or retreating behaviour is possible. At this time the solution begins to move suddenly, following the behaviour that the new contact angle dictates.

7.1 Fourth Order Case

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left[u^n \frac{\partial^3 u}{\partial x^3} \right] \tag{7.1.1}$$

with

$$u = \frac{\partial u}{\partial x} = u^n \frac{\partial^3 u}{\partial x^3} = 0 \qquad \text{at} \qquad x = b(t) \tag{7.1.2}$$

and

$$u = u_0(x) \qquad \text{at} \qquad t = 0 \tag{7.1.3}$$

where $x = b(t)$ represents the right hand moving behaviour, whose local behaviour will be considered.

Previous work has shown that for $n \in (0, 3)$, $b(t)$ represents a boundary moving at finite speed. In [10], Blowey et al. consider these values of n in detail, with various values of α, and Figure (1.1) from [10], reproduced here in Figure (7.1.1), illustrates their results.

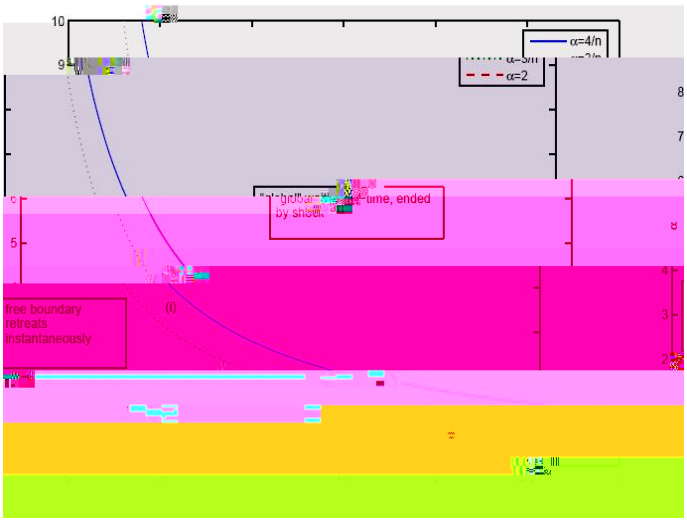


Figure 7.1.1: A summary of the possible small-time behaviours with respect to n and α

7.2 Sixth Order Case

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left[u^n \frac{\partial^5 u}{\partial x^5} \right] \tag{7.2.1}$$

with

$$u = \frac{\partial u}{\partial x} = \frac{\partial^3 u}{\partial x^3} = u^n \frac{\partial^5 u}{\partial x^5} = 0 \qquad \text{at} \qquad x = b(t) \tag{7.2.2}$$

and

$$u = u_0(x) \qquad \text{at} \qquad t = 0 \tag{7.2.3}$$

where $x = b(t)$ represents the right hand moving boundary.

Flitton and King, [17], conducted both asymptotic and numerical studies for the sixth order problem, and presented conjectures, relating to the value of n, which are outlined below.

A moving front regime has been identified for $0 < n < 5/2$, in which the free boundary is expected to move immediately, and the region $5/3 < n < 5/2$ has been identified as one in which the free boundary should advance instantaneously. For $5/2 < n < 6$, the free boundary is not expected to

move, as is the case for $n > 6$.

However, since Flitton and King use different boundary conditions from those used in this dissertation, we base the comparison of our results with those produced by Langdon, on a fixed mesh discretisation, in [22]

Chapter 8

Numerical Results for the Moving Boundary

To investigate the behaviour at the moving boundary, we take the initial data

$$u_0(x) = 5\max\left(\frac{9}{16} - x^2, 0\right)^{\alpha}, \quad \alpha \in \mathbb{R}^+ \tag{8.0.1}$$

8.1 The Fourth Order Case

For the fourth order case, where the small-time behaviour has been determined [10], the C++ program is used to verify the cases in which instantaneous advancing or retreating of the boundary is expected, since this movement will be apparent even over small time scales. For the waiting-time cases, this program was also run to verify that initially no movement of the boundary occurs. The FORTRAN program is then used for these cases, to see that the solution does eventually move after a finite waiting time.

The results presented here are chosen to provide examples of each type of behaviour

8.1.1 Instant Advance

From 7.1, we expect an instant advance of the free boundary for $\alpha < 4/n$, and $2 < n < 3$. Results for $n = 2.5$ and two di□erent values of α are shown in Figure (8.1.

of 1UCTJ+355.620.334(Mo)28(vse)sphavre acucingunts d(ify)-440(t)-3492sxprogsehavt a

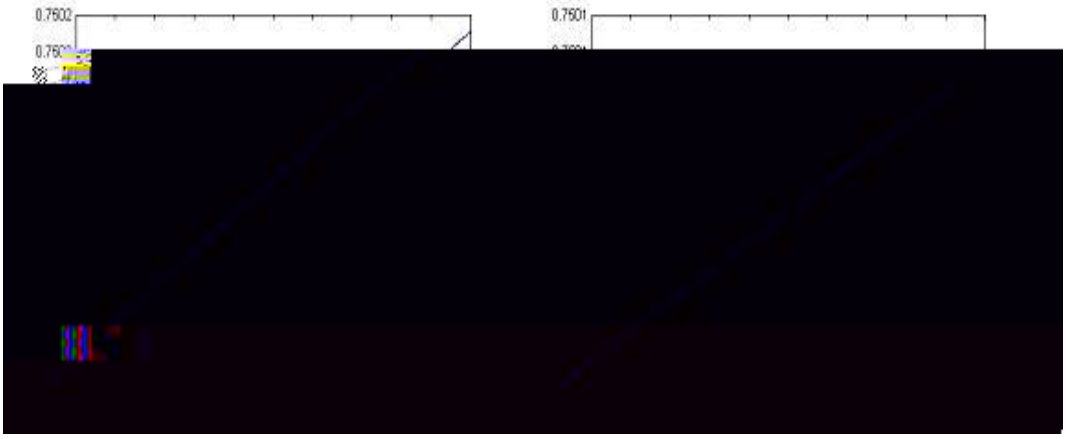


Figure 8.1.1: Advancing moving front with n = 2.5, alpha = 0.5, and n = 2.5, alpha = 1.4

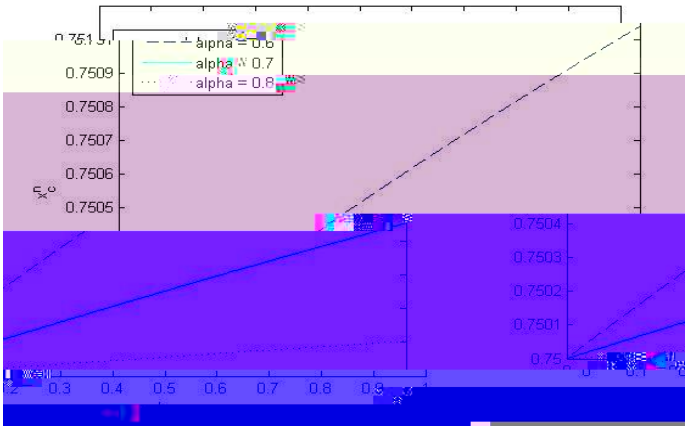


Figure 8.1.2: Advancing moving front for n = 1.0, alpha (0.5, 1.0)

(8.1.3).

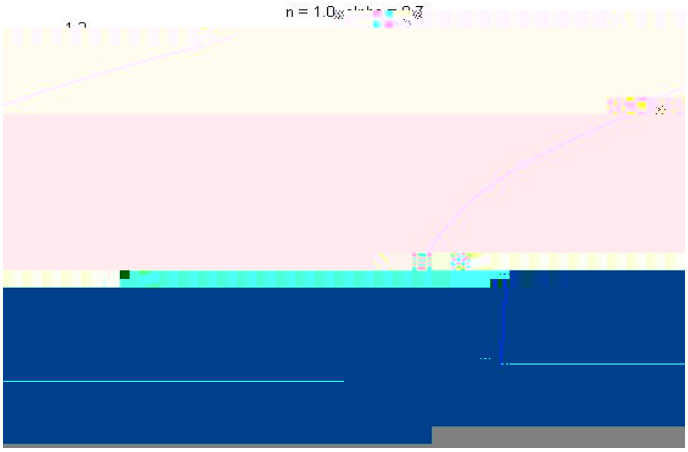


Figure 8.1.3: Advancing moving front for n = 1.0, alpha = 0.7, run to final time 0.002

8.1.2 Instant Retreat

The conjectures in 7.1, suggest an instant retreat of the free boundary for and $2 < \alpha < 3/n$. One of the results obtained supporting this is presented here.

Figure (8.1.4), shows the retreat of the solution for n and α satisfying $2 < \alpha < 3/n$. As for the advancing case, di□erent values of α for the same n were also investigated, and the results in Figure (8.1.5) suggest that the initial velocity of the boundary is again decreasing as α increases.

In Figure (8.1.6), two of the cases shown in Figure (8.1.5), were run with the FORTRAN program, and some interesting behaviour can be seen. For the smaller α value, the free boundary is seen to retreat initially and then advance. This suggests that for the other cases, this too is happening only on a much longer timescale than has been investigated here.

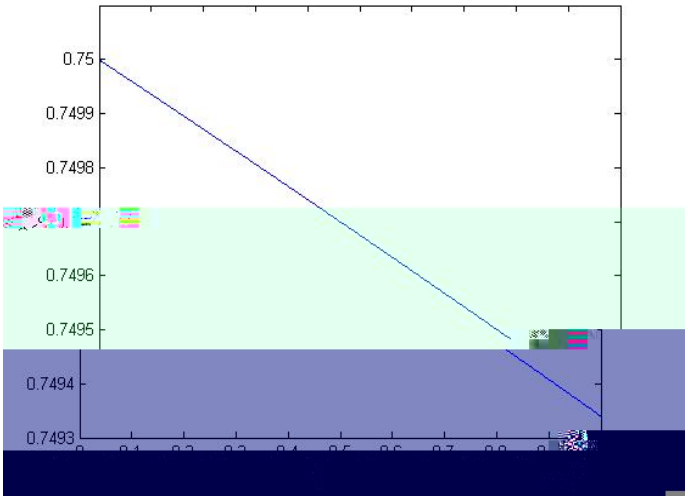


Figure 8.1.4: Retreating moving front for n = 1.0, alpha = 2.5

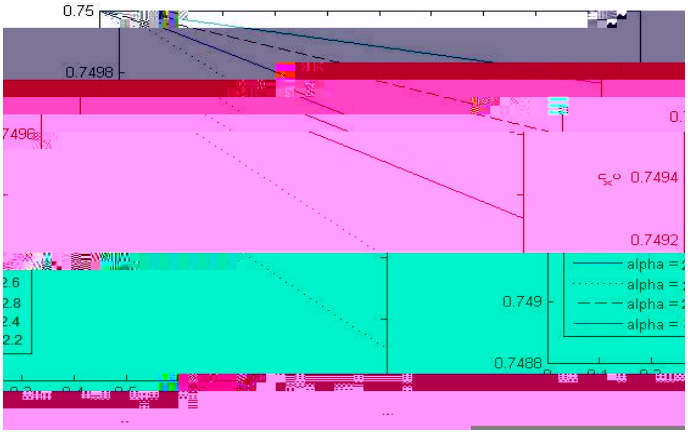


Figure 8.1.5: Retreating moving front for n = 1.0, alpha (2.0, 3.0)

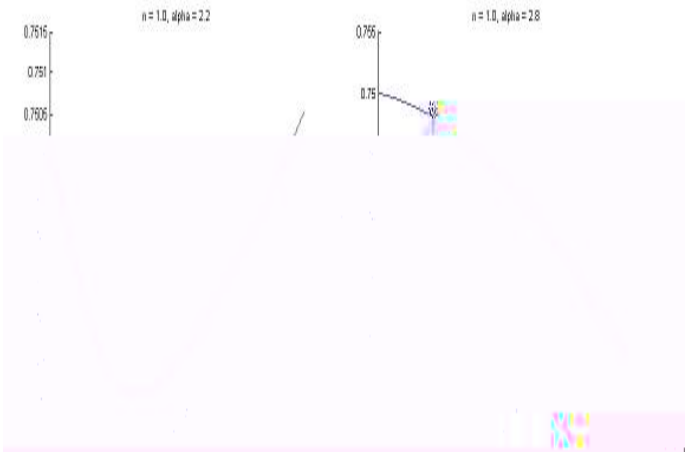


Figure 8.1.6: Retreating moving front for n = 1.0, alpha =2.2, and n = 1.0, alpha = 2.8

8.1.3 Waiting Time Behaviour

and time constraints meant that a comprehensive study of all possible cases could not be performed. As for the fourth order case, we take initial data (8.0.1).

The possible behaviours for the cases $n = 0.5$ to $n = 2.5$ were considered in detail. The findings are summarised in Figure

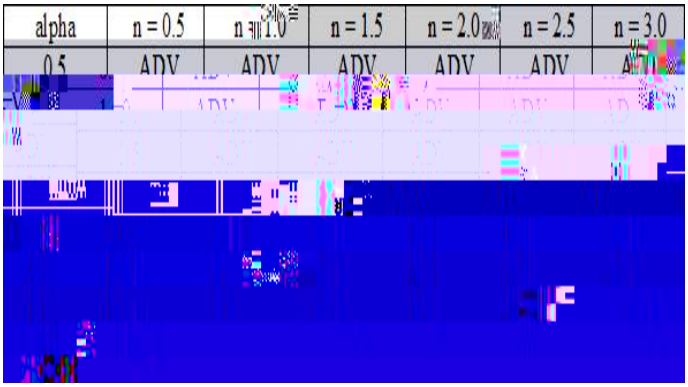


Figure 8.2.1: Table summarising behaviour observed for n (0.5, 2.5) and alpha (0.5, 3.5)

It is clear that the behaviour suggests that a change is occurring for α (2, 2, 5). This was investigated further, and our results are presented for the cases $n = 0.5$, and $n = 2.0$, in Figures (8.2.2), and (8.2.3). The results were run for 500 timetseps with a step size 10^{-12}. The number of timesteps was limited in order to gauge more information about the initial movement of the boundary. The figures suggest that upto a certain value of α, the solution advances instantaneously. Beyond this value, the results show that the solution starts to retreat initially before advancing. As the value of α increases beyond this point, the solution retreats for longer before advancing until, for the length of time the solutions were run, no subsequent advance can be seen. Investigations carried out as to the particular value of α for which the solution first starts to retreat, suggests that this value of alpha decreases as n increases, although for more conclusive results the

solutions should be computed with more variation of step size, and number of nodes in the mesh.

are present for $n = 5$, but not for any other values of n. Further experiments showed that as α was increased the behaviour of the moving front became smoother until no humps could be seen at all for $\alpha > 0.6$. While in the case $n = 5.0$ solutions matched those on the fixed mesh, for n larger, they did not. A possible reason for this is the conjecture that for sufficiently large values of n, the behaviour of the solution is entirely determined by α, and since the numerical method can never resolve α exactly, with more and more time steps as the moving mesh repositions nodes at the boundaries, the approximation to α changes continously. Based on this a further conjecture was proposed that the numerical scheme may not converge i.e. the problem itself could be an ill posed one [23].

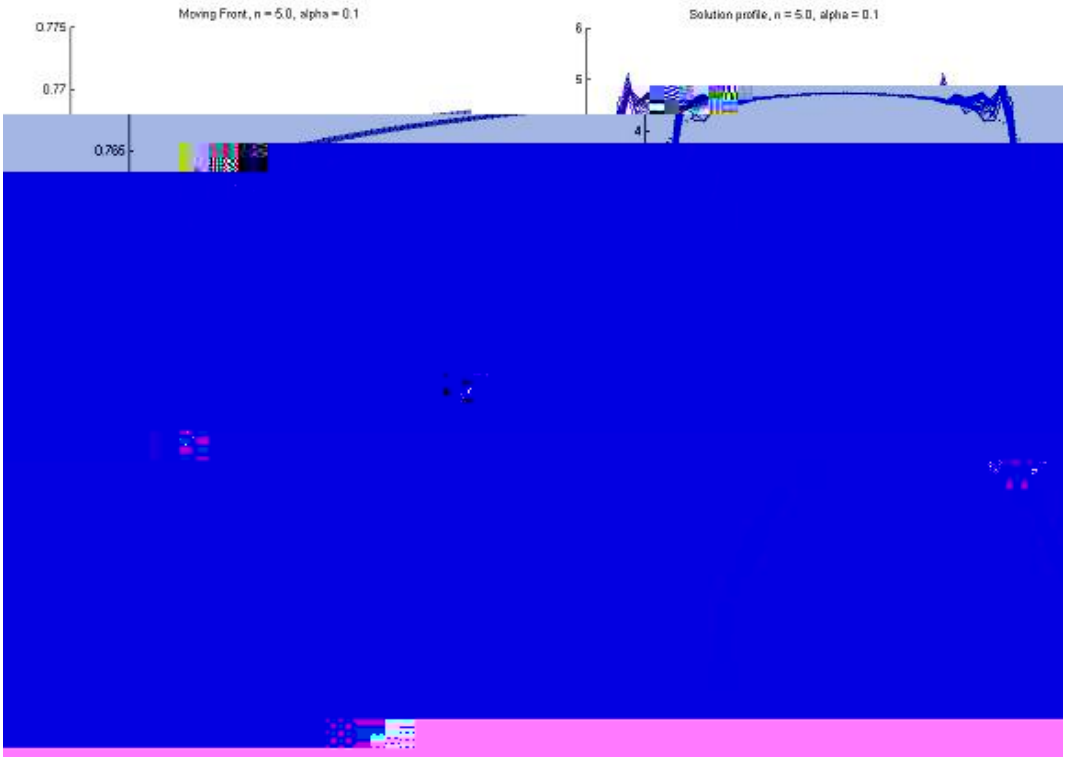


Figure 8.2.4: Behaviours observed in the case n = 5.0

Interestigly, formation of the humps in the solution profile was also seen

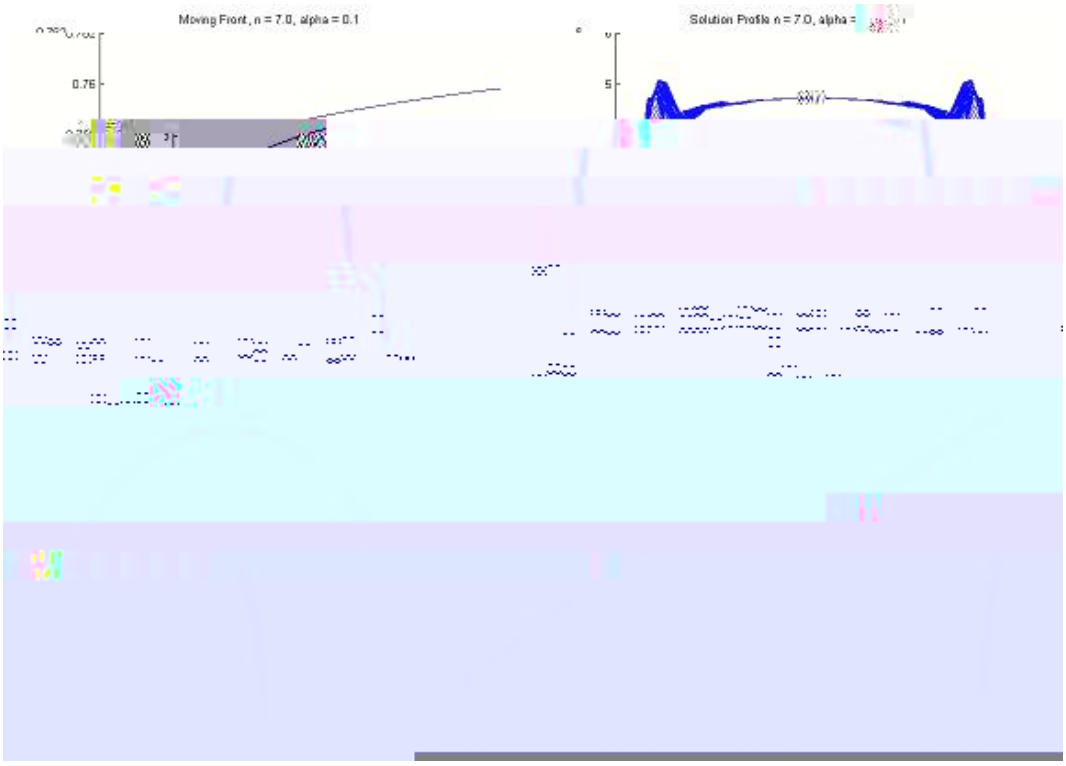


Figure 8.2.5: Behaviours observed in the case n = 7.0

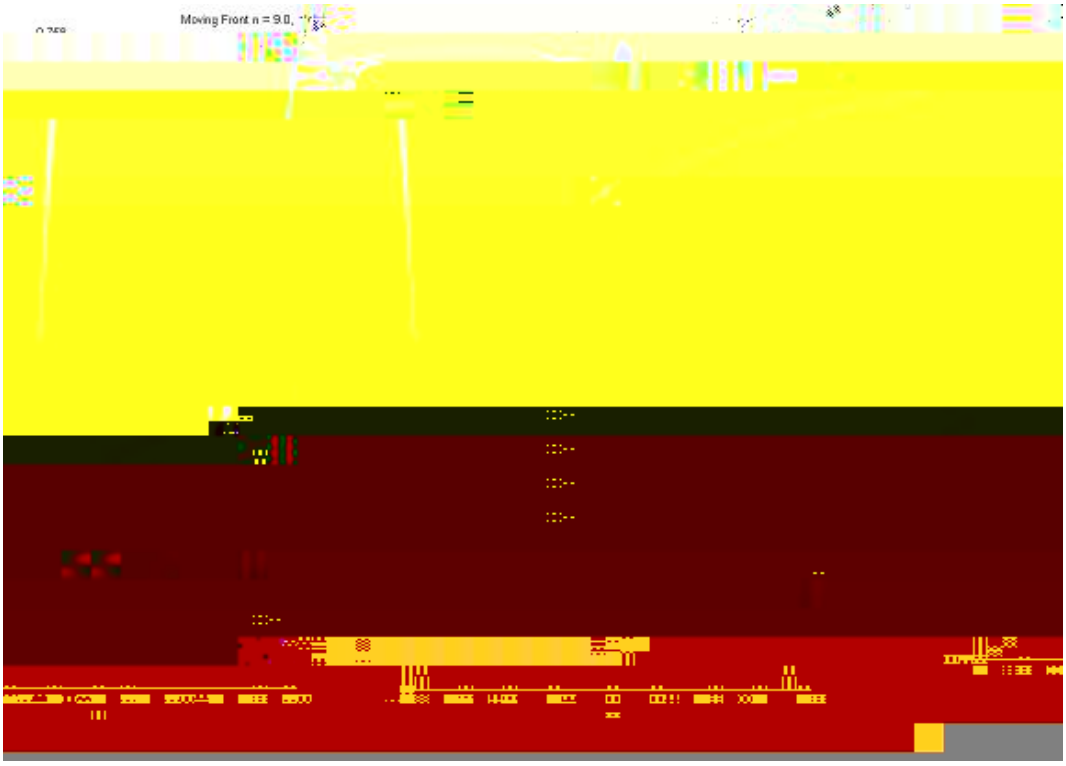


Figure 8.2.6: Behaviours observed in the case n = 9.0

in the cases α = 0.1, and smaller values of n = 1.0, n = 3.0. These results did not concur with those obtained on the fixed mesh. These results warrant further investigation as they may well be spurious.

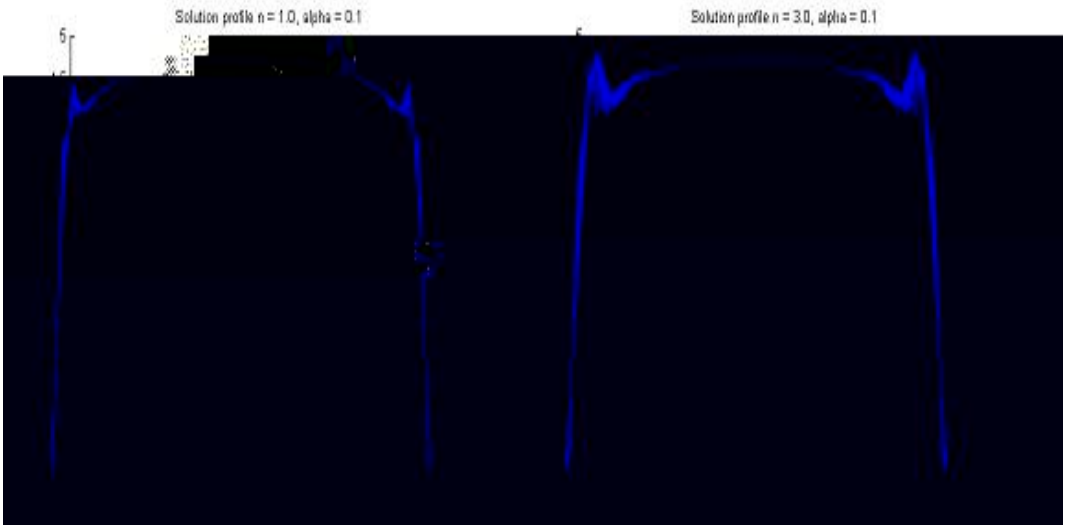


Figure 8.2.7: Humps in the solutions for n = 1.0, and n = 3.0

Chapter 9

equations, and similarity solutions to the nonlinear diffusion equations were constructed. In Chapter Six some numerical results to the self similar solution were presented and matters relating to timestep size and convergence with increasing nodes were investigated. Chapter Seven described possible behaviours of the moving boundary of the solution, and discussed the conjectures surrounding them In Chapter Eight, we applied our numerical method to the fourth and sixth order conjectures to see whether the moving mesh method could accurately resolve the features of the moving boundary.

9.2 Remarks and Further Work

Here we look at aspects not fully covered in this dissertation.

9.2.1 Scale Invariance

In Chapter Five, we remarked that figure (5.3.1), suggested that the moving mesh method possessed the same scale invariance properties as the PDE it was solving, and that this was a desirable property for the numerical scheme to have. For the purposes of this dissertation, the very small timesteps used meant that scale invariance could be assumed. It should be pointed out, however, that the numerical scheme is not strictly scale invariant. The ODE for the new node positions,

$$\dot{X} = F(X) \tag{9.2.1}$$

recovered in Chapter Four, is invariant under the mapping (5.1.3), as is the Forward Euler discretisation of (9.2.1) given by

$$\frac{X_{N+1} - X_N}{t_{N+1} - t_N} = F(X_N) \tag{9.2.2}$$

However this is not true of the local truncation error ,(LTE), of (9.2.1)

$$LTE = \frac{X_{N+1} - X_N}{t_{N+1} - t_N} - F(X_N) \tag{9.2.3}$$

which is not scale invariant under the mapping described in equation (). In [6], a timestepping method with a scale invariant LTE is outlined, by

[5], and investigations of the solution error in the L^1 norm were carried out. In the case of the Fourth Order Diffusion Problem, these illustrated a fourth order accuracy in one dimension, using a uniform initial mesh, with diffusion coefficient $n = 1$. It would be appropriate to consider investigating the accuracy of the solution for the Sixth Order Problem in a similar manner.

9.2.4 Initial Grid Distribution

For the purposes of this dissertation, a uniform initial distribution of the nodes was used, where the distance between each of the nodes in the mesh was equal.

We could investigate how the solution varies, if at all, with the initial mesh used. For example, the equidistribution algorithm outlined by Baines [4], could be used, to start with an initial mesh in which the nodes are placed so the mass is equal in each cell.

It was remarked in Chapter Six, that the use of mass as a monitor function, resulted in the nodes following the moving boundary, but did not necessarily seem to increase the distribution of nodes to these areas. To remedy this, we could start with an initial mesh, in which the nodes were more clustered around the boundary, for example, by requiring that smaller amounts of mass are placed in this region. An extension to the method used in this dissertation, is then to investigate the moving front, in the sixth order case, where, as noted in Chapter Eight, worsening resolution at the boundaries for increasing values of the diffusion coefficient could mean that the numerical schemes do not converge for these situations. If we start with an initial mesh with improved resolution of the moving front, this could result in improved resolution of the contact angle α, and may improve the solutions obtained in Chapter Eight.

9.2.5 Extension to Further Applications

Further work could consider the application of this moving mesh method to problems in two dimensions, in order to better apply them to examples considered in Chapter two. One such possibilty would be to consider the problem of oxygen di□usion in tumour growth, and apply the method considered here to a two dimensional radial proble, which would better model the distribution of a tumour.

Bibliography

[1] D.G.Aronson. Some Problems in Nonlinear Di□fusion, Editors: A.Fasano, M.Primiceno, *Lecture Notes in Mathematics*,1224 Springer-Verlag, New York (1986)

[2] D.G.Aronson, H.F.Weinberger. Nonlinear Di□usion in Population Genetics and Nerve Pulse Propagation, Partial Di□erential Equations and Related Topics, *Lecture Notes in Mathematics*,446 Springer-Verlag, New York (1975), 5-49

[3] J.P.Armitage, F.S.Garduno, P.K.Maini, R.A.Satnoianu. Travelling Waves in a Nonlinear Degenerate Di□usion Model for Bacterial Pattern Formation, *Discrete and Continuous Dynamical Systems - Series B*,1 (2001), 339-362

[4] M.J.Baines. Grid Adaptation via Node Movement, *Applied Numerical Mathematics*,26 (1998), 77-96

[5] M.J.Baines, M.E.Hubbard, P.K.Jimack. A Moving Mesh Finite Element Algorithm for the Adaptive Solution of Time-Dependent Partial Di□erential Equations with Moving Boundaries, *Applied Numerical Mathematics*,54 (2005), 450-469

[6] M.J.Baines, M.E.Hubbard, P.K.Jimack, A.C.Jones. Scale-invariant Moving Finite Elements for Nonlinear Partial Di□erential Equations in Two Dimensions. *Applied Numerical Mathematics*,56 (2006), 230-252

[16] J.Crank, R.S.Gupta. A Moving Boundary Problem Arising from the

[26] T.G.Myers. Thin Films with High Surface Tension, *SIAM Review*,40 (1988), 441-462

[27] T.G.Myers. Surface Tension Driven Thin Film Flows, *Mechanics of Thin Film Coatings*,Wiley, 1996

[28] N.F.Smyth and J.M.Hill. High Order Nonlinear Di□usion *IMA Journal of Applied Mathematics*,40 (1988), 73-86