

A Moving Mesh Finite Element Method for High
Order Non-Linear Diffusion Problems

Bonhi Bhattacharya

August 21, 2006

Acknowledgments

Firstly I would like to thank my supervisors, Steve Langdon for his much appreciated help and enthusiasm during this dissertation, and Mike Baines for the same, as well his endless patience in answering all my questions! Secondly, thanks to all the friends I've made in the maths department at Reading, for making this year so much fun, as well as for all the help and support you have given me through the hard times. Good luck for the future. Special thanks also to Pete Sweby for being so understanding during this year. Last but not least, I would like to thank my Mum, Dad and Mils for all their encouragement, support and love. xb

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Bonhi Bhattacharya.

Abstract

In this dissertation a moving mesh method, based on a conservation of mass principle, is used to produce numerical approximations to the solutions of the fourth order and sixth order nonlinear diffusion equations. The behaviour of the moving boundary of these solutions is investigated in detail, and is compared to existing asymptotic results for the fourth order case. For the sixth order case, conjectures as to the behaviour of the moving boundaries are investigated.

Contents

1	Introduction	1
2	Non-Linear Diffusion	4
2.1	The Porous Medium Equation	5
2.2	The Fourth Order Thin Film Equation	6
2.3	Sixth Order Non-Linear Diffusion	7
3	Moving Mesh Methods	9
3.1	Velocity Based Methods	10
4	A Moving Mesh Method	13
4.1	A Conservation Principle	13
4.2	Finite Element Formulation	15
4.2.1	Timestepping	20
5	Self-Similar Solutions	22
5.1	Scale Invariance	22
5.2	Self Similar Solutions	24
5.2.1	A Fourth Order Self Similar Solution	25
5.2.2	A Sixth Order Self Similar Solution	27
5.3	Scale Invariance of the Numerical Method	28
6	Numerical Results for the Self Similar Solution	30

6.1	Fourth Order Results	34
6.2	Sixth Order Results	34
7	Behaviour of the Moving Boundary	36
7.1	Fourth Order Case	37
7.2	Sixth Order Case	38
8	Numerical Results for the Moving Boundary	40
8.1	The Fourth Order Case	41
8.1.1	Instant Advance	41
8.1.2	Instant Retreat	43
8.1.3	Waiting Time Behaviour	45
8.2	The Sixth Order Case	45
9	Conclusions and Further Work	54
9.1	Summary	54
9.2	Remarks and Further Work	55
9.2.1	Scale Invariance	55
9.2.2	Timestepping	56
9.2.3	Errors	56
9.2.4	Initial Grid Distribution	57
9.2.5	Extension to Further Applications	57

List of Figures

4.2.1 Linear Finite Element Basis Function	16
5.2.1 Distribution of the fourth order self similar solution	27
5.2.2 Distribution of the sixth order self similar solution	28
5.3.1 Invariance of the fourth order solution and mesh for 41 node mesh	29
6.0.1 Convergence of numerical solution to true solution with in- creasing nodes, $t = 0.0005$	31
6.0.2 Resolution of numerical solution at boundary with increasing nodes, $t = 0.0005$	32
6.0.3 Timestep dependence on the number of nodes in the mesh . .	33
6.1.1 Exact and approximate solutions for the fourth order problem.	34
6.2.1 Exact and approximate solutions for the sixth order problem.	35
7.0.1 Change in contact angle for different alpha	37
7.1.1 A summary of the possible small-time behaviours with respect to n and α	38
8.1.1 Advancing moving front with $n = 2.5$, $\alpha = 0.5$, and $n =$ 2.5 , $\alpha = 1.4$	42
8.1.2 Advancing moving front for $n = 1.0$, $\alpha \in (0.5, 1.0)$	42
8.1.3 Advancing moving front for $n = 1.0$, $\alpha = 0.7$, run to final time 0.002	43

8.1.4 Retreating moving front for $n = 1.0$, $\alpha = 2.5$	44
8.1.5 Retreating moving front for $n = 1.0$, $\alpha \in (2.0, 3.0)$	44
8.1.6 Retreating moving front for $n = 1.0$, $\alpha = 2.2$, and $n = 1.0$, $\alpha = 2.8$	45
8.1.7 Apparent waiting time behaviour, for different n and α	46
8.1.8 Actual waiting time behaviour, for different n and α	46
8.2.1 Table summarising behaviour observed for $n \in (0.5, 2.5)$ and $\alpha \in (0.5, 3.5)$	47
8.2.2 Moving front results for $n = 0.5$	48
8.2.3 Moving front results for $n = 2.0$	49
8.2.4 Behaviours observed in the case $n = 5.0$	50
8.2.5 Behaviours observed in the case $n = 7.0$	51
8.2.6 Behaviours observed in the case $n = 9.0$	52
8.2.7 Humps in the solutions for $n = 1.0$, and $n = 3.0$	53

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.

An adaptive mesh scheme becomes preferable to a fixed mesh scheme when these areas of interest represent only a fraction of the domain being investigated. Increasing resolution selectively at these regions is computationally less expensive than refinement of the mesh over the entire grid.

There are three main types of grid adaptation, the most common type being h -refinement, which adds or removes nodes to or from the existing mesh, resulting in local refinement or coarsening of the mesh. Another is p -refinement, which changes the order of the polynomial used in the finite element approximation according to the smoothness of the solutions. The least common type is r -refinement, known as the *moving mesh method*, which relocates mesh points to concentrate them where needed. Such moving meshes are attracting increasing interest, especially in the numerical approximation of time dependent problems since the continuous movement of the mesh allows easier inclusion of time integrators.

This dissertation will investigate the application of a particular moving mesh method, based on a conservation of monitor function principle. This method is applied, in one dimension, to a class of evolutionary, degenerate partial differential equations (PDEs), the non-linear diffusion equations with moving boundaries.

In Chapter Two, we provide examples of some of the applications of nonlinear diffusion, in order to motivate the numerical solutions of these equations. We consider the second order Porous Medium Equation, which has been extensively investigated, and then go on to consider applications of the fourth and sixth which are not so well known. Both physical and biological applications are considered. In Chapter Three, we go on to consider moving mesh techniques in general and give a brief account of how they are constructed. We also introduce the principles on which the moving mesh method we use here is based. Chapter Four then goes on to develop this moving mesh method by constructing mesh equations based on a conservation of mass principle, and then outlining a finite element numerical solution to the problem. Limitations of the model, in terms of programming are also considered. Chapter Five introduces scale invariance to illustrate the construction of self-similar solutions of the nonlinear diffusion equations. These solutions are a class of true solutions which can be used to verify the numerical solutions we obtain, since analytic solutions are not known. The chapter concludes by looking at the scale invariant properties of the numerical model. In Chapter Six, we present the numerical results obtained by using the similarity solutions evaluated at time $t = 0$, as initial data for the method. This enables us to see whether the numerical method produces diffusive results as we hope that it would. We also consider the convergence of the numerical solution to the true solution as the number of nodes in the mesh increases, and look at issues relating to the size of the timestep. Chapter Seven considers the moving boundary of the solution and discusses

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 investigate 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 variety of physical and biological situations in which they arise.

Nonlinear diffusion equations are considered, with general form

$$\frac{\partial u}{\partial t} = (-1)^m \frac{\partial}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} \right) \quad (2.0.1)$$

Equations of this type describe many physical processes such as heat transfer in ionized gases, unconfined groundwater flow, electric transmission in cables with resistive coatings, and many other phenomena.

The equations derive from the generalised Reynolds equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u^n \frac{\partial p}{\partial x} \right) \quad (2.0.2)$$

where p represents a driving force which determines the order of the equation [17].

n represents the diffusion coefficient which represents the viscosity of the

material through which the diffusion is occurring.

2.1 The Porous Medium Equation

The Porous Medium equation, which is characterised by $p = u$ in (2.0.1), representing a gravity driven flow, is an example of a second order non-linear diffusion equation.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u^n \frac{\partial u}{\partial x} \right) \quad (2.1.1)$$

The second order case has been widely investigated and forms the basis for many physical models. It is used to model the percolation of gas through a porous medium and the spreading of thin viscous liquid films spreading under gravity, [1]. The modelling of diffusion of oxygen into a medium which is simultaneously consuming oxygen also uses second order nonlinear diffusion [16].

Further applications of the second order case arise in biological modelling. Nonlinear degenerate diffusion is used to model bacterial density, in models describing spatiotemporal evolution of bacterial colonies on agar plates, [3], where

$$\frac{\partial b}{\partial t} = D_b \frac{\partial}{\partial x} \left(nb \frac{\partial b}{\partial x} \right) + nb \quad (2.1.2)$$

for nutrient concentration, n , bacterial cell density, b , and D_b , the diffusion coefficient of the bacteria.

The Porous Medium equation also occurs in medical modelling. Most tissues of the human body, for example, bone cartilage, muscle, are porous media, and their functioning depends on the flow of fluids such as blood through them. Porous Medium models have been applied to better under-

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

In the first stage of tumour development, tumours are ordinarily avascular, and they gain nutrients and oxygen for growth by diffusion from already existing vasculature surrounding them. Thus the size of the tumour is initially limited by diffusion through a porous medium. Non-linear diffusion models are used to represent this stage of tumour growth, [12].

The second order non linear diffusion equation is used in electrophysiology to describe propagation of nerve action potentials, [2], [25]. In a nerve axon, dissipation of the energy of a moving impulse occurs as a result of ohmic losses of internal and external ionic current flows through the axon, which occur due to changes in membrane permeability as the impulse travels along the axon, and is modelled by

$$\frac{\partial v}{\partial t} = \left(\frac{1}{rc} \right) \frac{\partial^2 v}{\partial x^2} - j_{ion} \quad (2.1.3)$$

where $\frac{1}{rc}$ represents the diffusivity of the material in the axon.

2.2 The Fourth Order Thin Film Equation

The fourth order non-linear diffusion equation

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x} \left(u^n \frac{\partial^3 u}{\partial x^3} \right) \quad (2.2.1)$$

in the case $p = -\frac{\partial^2 u}{\partial x^2}$ in (2.0.1), occurs when surface tension drives a thin film flow, where u represents the height of the film.

The most common occurrence of the thin film equation is as a lubrication approximation of the Navier Stokes equations for thin film viscous flows [26], and models numerous physical processes. In the case $n = 3$, the equation is used to model situations involving the spreading of a liquid film along a solid surface, from coating flows, for example, rain running down a window pane, to the evolution of drying paint layers [27].

When $n = 1$, the equation models flow in a Hele-Shaw cell. In a Hele-Shaw cell, a fluid is placed between two closely spaced parallel plates, and the fluid moves in response to pressure gradients arising from surface tension, and externally imposed forces, [21].

The fourth order equation has also been used to model the motion of a contact lens on a tear film [24]. The equation for the film height is given by

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x} \left(C \frac{u^3}{3} u_{xxx} - v(t)u \right) \quad (2.2.2)$$

where $v(t)$ represents the eyeball velocity.

2.3 Sixth Order Non-Linear Diffusion

This sixth order case,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u^n \frac{\partial^5 u}{\partial x^5} \right) \quad (2.3.1)$$

also forms the basis for physical phenomena, although this case has not been investigated in as much detail as the fourth and second order cases.

In semiconductor fabrication, the equation is used to model a viscous flow of silicon oxide occurring in a long thin region, under an elastic nitride cap [18].

The slow viscous flow of silicon is modelled using the second order incompressible Reynold Equation, and the elastic nitride cap is modelled by the fourth order light beam equation. Coupling of these two equations results in a nonlinear sixth order equation,

$$\frac{\partial u}{\partial t} = \frac{\lambda}{12} \frac{\partial}{\partial x} \left(u^3 \frac{\partial^5 u}{\partial x^5} \right) \quad (2.3.2)$$

where u is the film thickness, x represents the distance under the cap, and λ , represents the flexural rigidity.

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 diffraction 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

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

for M , some monitor function, with the form

$$M = M(x, u, u_x, u_{xx} \dots) \quad (3.1.2)$$

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

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

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 diffusion equations with moving boundaries in one dimension.

Chapter 4

A Moving Mesh Method

In this chapter a method is developed, based on conservation of the proportion within each computational cell, of the total integral of the mass u , over the domain being considered. The method is constructed to return the velocities of the mesh nodes, which move so that the conservation principle is satisfied within each interval. The velocities are obtained by differentiating the conservation principle with respect to time, to determine how the nodes move so mass is conserved. The new mesh positions are generated from the velocities via a time stepping algorithm.

4.1 A Conservation Principle

Consider the general non linear diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} \right) \quad (4.1.1)$$

with $u = 0$ at the boundaries $(x_0(t), x_N(t))$, of the domain being considered.

Integrating equation (4.1.1) from $x_0(t)$ to $x_N(t)$ gives

$$\begin{aligned}
 \frac{d}{dt} \int_{x_0(t)}^{x_N(t)} u(t) dx &= \int_{x_0(t)}^{x_N(t)} \frac{\partial u}{\partial t} dx + \frac{d}{dt} x_N(t) \frac{d}{dx_N(t)} \int_{x_0(t)}^{x_N(t)} u(t) dx \\
 &\quad + \frac{d}{dt} x_0(t) \frac{d}{dx_0(t)} \int_{x_0(t)}^{x_N(t)} u dx \\
 &= \int_{x_0(t)}^{x_N(t)} \frac{\partial u}{\partial t} dx + \dot{x}_N(t) u(x_N(t)) - \dot{x}_0(t) u(x_0(t))
 \end{aligned}$$

Then substituting the value of $\frac{\partial u}{\partial t}$ from (4.1.1) gives

$$\begin{aligned}
 \frac{d}{dt} \int_{x_0(t)}^{x_N(t)} u(t) dx &= \int_{x_0(t)}^{x_N(t)} \frac{\partial}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} \right) dx + [\dot{x}u]_{x_0(t)}^{x_N(t)} \\
 &= \left[u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} + \dot{x}u \right]_{x_0(t)}^{x_N(t)} \\
 &= 0
 \end{aligned} \tag{4.1.2}$$

using the conditions $(x_0(t) = 0$ and $x_N(t) = 0$ at the boundaries. Hence

$$\frac{d}{dt} \int_{x_0(t)}^{x_N(t)} u(t) dx = 0 \tag{4.1.3}$$

and so

$$\int_{x_0(t)}^{x_N(t)} u(t) dx = 0 \tag{4.1.4}$$

demonstrating conservation of mass over the domain.

A local conservation principle consistent with mass conservation is now introduced such that for any interior points $x_{i-1}(t)$ and $x_i(t)$,

$$\int_{x_{i-1}(t)}^{x_i(t)} u(t) dx = \text{constant in time} \tag{4.1.5}$$

$$\frac{d}{dt} \int_{x_{i-1}(t)}^{x_i(t)} u(t) dx = 0 \tag{4.1.6}$$

Using Leibnitz' rule in (4.3), gives

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

Substitution from (4.1.1) leads to

$$\left[u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} + \dot{x} u \right]_{x_0(t)}^{x_i(t)} = 0 \quad (4.1.7)$$

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

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

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

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

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

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 conservation of mass principle (4.1.4), using the new values of x_i and x_{i+1} .

4.2 Finite Element Formulation

In practice a numerical method is used to solve these equations, and here this is obtained by using finite element discretisations. Before introducing finite elements, a weak form of the problem is constructed. A weak form of the conservation of mass principle (4.1.4) is

$$\int_{x_{i-1}(t)}^{x_i(t)} w_i u(t) dx = \text{constant in time} \quad (4.2.1)$$

where the w_i are continuous and once differentiable test functions, which form a partition of unity.

$$\sum_{j=1}^N w_j = 1 \quad (4.2.2)$$

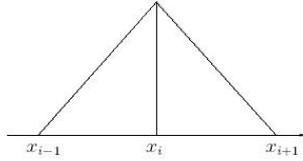


Figure 4.2.1: Linear Finite Element Basis Function

Differentiating with respect to time gives a weak form of equation (4.1.2) for the nonlinear diffusion equation,

$$\int_{x_N(t)}^{x_0(t)} w \frac{\partial}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} + \dot{x}u \right) dx = 0 \quad (4.2.3)$$

The test functions used take the form of linear hat functions, ϕ_i , as in figure (4.1). After integration by parts, (4.2.3) gives

$$\phi_i \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} + \dot{x}u \right) - \int_{x_{i-1}(t)}^{x_{i+1}(t)} \frac{\partial \phi_i}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} + \dot{x}u \right) dx = 0 \quad (4.2.4)$$

The first term disappears for interior nodes as the basis function ϕ_i equals 0 at $x_i(t)$ and $x_{i+1}(t)$, and also disappears at the boundary nodes, since $u = 0$ at these nodes. Applying the boundary conditions, equation (4.2.4) becomes

$$\begin{aligned} \int_{x_{i-1}(t)}^{x_{i+1}(t)} u \frac{\partial \phi_i}{\partial x} \dot{x} dx &= - \int_{x_{i-1}(t)}^{x_{i+1}(t)} u^n \frac{\partial \phi_i}{\partial x} \frac{\partial^{2m+1} u}{\partial x^{2m+1}} dx \\ &= - \int_{x_{i-1}(t)}^{x_{i+1}(t)} u^n \frac{\partial \phi_i}{\partial x} \frac{\partial q}{\partial x} dx \end{aligned} \quad (4.2.5)$$

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

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

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

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

$$\begin{aligned} \int_{x_{i-1}}^{x_{i+1}} \phi_i p dx &= - \int_{x_{i+1}}^{x_{i-1}} \phi_i \frac{\partial^2 u}{\partial x^2} \\ &= \left[-\phi_i \frac{\partial u}{\partial x} \right]_{x_{i-1}}^{x_{i+1}} + \int_{x_{i+1}}^{x_{i-1}} \frac{\partial \phi_i}{\partial x} \frac{\partial u}{\partial x} dx \end{aligned} \quad (4.2.8)$$

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

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

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

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

These forms are substituted in to (4.2.6). Then

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

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

$$K_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \frac{\partial \phi_i}{\partial x} \frac{\partial \phi_j}{\partial x} dx \quad (4.2.12)$$

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

$$M_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \phi_i(x) \phi_j(x) dx \quad (4.2.13)$$

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

$$K \underline{p} = M \underline{q} \quad (4.2.14)$$

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

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

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

$$\dot{x} = \frac{d\psi}{dx} \quad (4.2.16)$$

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

$$\psi = \sum_{j=1}^N \psi_j \phi_j \quad (4.2.17)$$

This results in (4.2.14) becoming the stiffness matrix system

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

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

$$K_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} u \frac{\partial \phi_i}{\partial x} \frac{\partial \phi_j}{\partial x} dx \quad (4.2.19)$$

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

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

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),

$$\int_{x_{i-1}}^{x_{i+1}} \phi_i \dot{x} dx = \int_{x_{i-1}}^{x_{i+1}} \phi_i \frac{d\psi}{dx} dx \quad (4.2.21)$$

which results in the mass matrix system

$$M \underline{\dot{x}} = B \underline{\psi} \quad (4.2.22)$$

where B has entries of the form

$$B_{ij} = \int_{x_{i-1}(t)}^{x_{i+1}(t)} \phi_i \frac{\partial \phi_j}{\partial x} dx \quad (4.2.23)$$

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

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

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

Once the velocities of the nodes have been found, the new position of these nodes is found by integrating the mesh forward using any timestepping algorithm , for example, the Forward Euler scheme.

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

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

where c is given by

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

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 procedure. However, although implementation of explicit schemes is much easier, the associated restrictions on the size of the timestep for stability, can cause a problem.

The method used in this dissertation is the explicit Forward Euler discretisation

$$\dot{x} = \frac{x^{n+1} - x^n}{\Delta t} \tag{4.2.26}$$

This scheme, although convergent, is only conditionally stable, and requires very small timesteps to ensure stability and avoidance of node overtaking, which results in the mesh nodes becoming tangled (see for example, Figure

(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 results 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,

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

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

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

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

scaling transformations is considered. For any nonlinear partial differential equation, satisfied by the set (x, u, t) , a scaling transformation is described by a mapping to $(\bar{x}, \bar{u}, \bar{t})$, given by the transformation

$$x = \lambda^\beta \bar{x}, \quad u = \lambda^\gamma \bar{u}, \quad t = \lambda \bar{t}. \quad (5.1.3)$$

for some arbitrary positive quantity λ . If the PDE is unchanged by the transformation, then the system is said to be scale invariant. This is illustrated below for the general nonlinear diffusion problem.

Consider the equation

$$\frac{\partial u}{\partial t} = (-1)^m \frac{\partial}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} \right) \quad (5.1.4)$$

under the transformation (5.1.3). The left hand side of this equation becomes

$$\frac{\partial u}{\partial t} = \frac{\partial(\lambda^\gamma \bar{u})}{\partial(\lambda \bar{t})} = \lambda^{\gamma-1} \frac{\partial \bar{u}}{\partial \bar{t}} \quad (5.1.5)$$

and the right hand side becomes

$$\begin{aligned} \frac{\partial}{\partial x} \left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}} \right) &= \frac{\partial}{\partial(\lambda^\beta \bar{x})} \left(\lambda^{n\gamma} \bar{u}^n \frac{\partial^{2m+1}(\lambda^\gamma \bar{u})}{\partial(\lambda^\beta \bar{x})^{2m+1}} \right) \\ &= \lambda^{(\gamma(n+1)-(2m+2)\beta)} \frac{\partial}{\partial \bar{x}} \left(\bar{u}^n \frac{\partial^{2m+1} \bar{u}}{\partial \bar{x}^{2m+1}} \right) \end{aligned} \quad (5.1.6)$$

Equating equations (5.1.5) and (5.1.6) gives the transformed PDE,

$$\lambda^{\gamma-1} \frac{\partial \bar{u}}{\partial \bar{t}} = \lambda^{(\gamma(n+1)-(2m+2)\beta)} \frac{\partial}{\partial \bar{x}} \left(\bar{u}^n \frac{\partial^{2m+1} \bar{u}}{\partial \bar{x}^{2m+1}} \right) \quad (5.1.7)$$

For the original equation (5.1.4) to be scale invariant under the transformation $(x, u, t) \Rightarrow (\bar{x}, \bar{u}, \bar{t})$ requires

$$\gamma - 1 = (n + 1)\gamma - (2m + 1)\beta \quad (5.1.8)$$

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

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

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

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,

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

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

$$y = \frac{x}{t^\beta} = \frac{\bar{x}}{\bar{t}^\beta} \quad v = \frac{u}{t^\gamma} = \frac{\bar{u}}{\bar{t}^\gamma} \quad (5.2.6)$$

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.

$$\gamma = \frac{-1}{n+4} \quad \beta = \frac{1}{n+4} \quad (5.2.7)$$

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

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

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

$$\frac{\partial u}{\partial x} = \frac{dy}{dx} \frac{du}{dv} \frac{dv}{dy} \quad (5.2.9)$$

From (5.1.3)

$$\frac{dy}{dx} = t^{-\beta}, \quad \frac{du}{dv} = t^\gamma \quad \text{giving} \quad \frac{\partial u}{\partial x} = t^{\gamma-\beta} v' \quad (5.2.10)$$

from which,

$$\frac{\partial^2 u}{\partial x^2} = t^{\gamma-2\beta} v'' \quad (5.2.11)$$

$$\frac{\partial^3 u}{\partial x^3} = t^{\gamma-3\beta} v''' \quad (5.2.12)$$

Then

$$\begin{aligned} \frac{\partial}{\partial x} \left(u^n \frac{\partial^3 u}{\partial x^3} \right) &= \frac{\partial}{\partial x} [(v^n t^{(\gamma n)} t^{\gamma-3\beta} v''')] \\ &= t^{\gamma(n+1)-3\beta} \frac{\partial}{\partial x} (v^n v''') \\ &= t^{\gamma(n+1)-(3+n)\beta} (v' v''')' \end{aligned} \quad (5.2.13)$$

In [28], it is noted that the general nonlinear diffusion equation admits a simple exact similarity solution for $n = 1$, and substituting this value of n into (5.2.13), then equating with (5.2.8) results in the following ordinary differential equation, cancelling powers of t ,

$$-\beta y v' + \gamma v = (v' v''')' \quad (5.2.14)$$

which is independent of time. Taking $\bar{t} = 1$, the transformed solution can be mapped back to the original co-ordinates using,

$$\bar{x} = x t^{-\frac{1}{n+4}}, \bar{u} = u t^{\frac{1}{n+4}} \quad (5.2.15)$$

In the case $n = 1$, (5.2.14) is satisfied by a solution of the form

$$u(x, t) = t^{-\frac{1}{5}} \phi \left(\frac{x}{t^{\frac{1}{5}}} \right) \quad (5.2.16)$$

for values of β and γ given in (5.2.4).

For the purposes of evaluating the numerical results obtained in this dissertation, a source type similarity solution given in [8], is used,

$$u(x, t) = \frac{1}{120(t + \tau)^{\frac{1}{5}}} \left[\omega^2 - \frac{x^2}{(t + \tau)^{2/5}} \right]_+^2 \quad (5.2.17)$$

where τ and ω , are arbitrary positive constants. $\tau = 4^{-5}$, and $\omega = 2$ were the values used for these constants and the distribution of this solution at different times is shown in figure (5.2.1).

The results presented in the next section show the approximations to this self similar solution over time, using initial data, at $t = 0$

$$u(x, 0) = \frac{1}{30} [4 - 16x^2]_+^2 \quad (5.2.18)$$

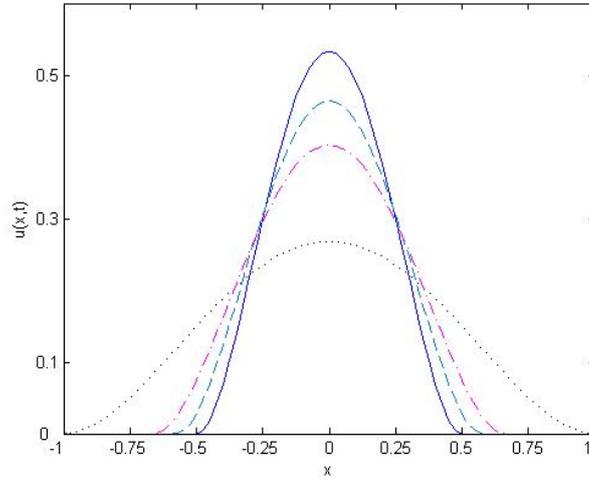


Figure 5.2.1: Distribution of the fourth order self similar solution

5.2.2 A Sixth Order Self Similar Solution

Constructing the sixth order self similar solution, in the same manner as the fourth order results in the fifth order ODE

$$-\beta yv' + \gamma v = (v'v''''')' \quad (5.2.19)$$

where, from (),

$$\gamma = \frac{-1}{n+6} \quad \beta = \frac{1}{n+7} \quad (5.2.20)$$

and again, from [28], for $n = 1$, this ODE has a solution of the form

$$u(x,t) = t^{-\frac{1}{7}} \phi\left(\frac{x}{t^{\frac{1}{7}}}\right) \quad (5.2.21)$$

For evaluation of numerical results, the sixth order similarity solution

$$u(x,t) = \frac{1}{50400(t+\tau)^{\frac{1}{77}}} \left[\omega^2 - \frac{x^2}{(t+\tau)^{2/7}} \right]_+^3 \quad (5.2.22)$$

from [9] is used, with $\tau = 4^{-7}$, and $\omega = 2$. The distribution of this solution can be seen in figure (5.2.2).

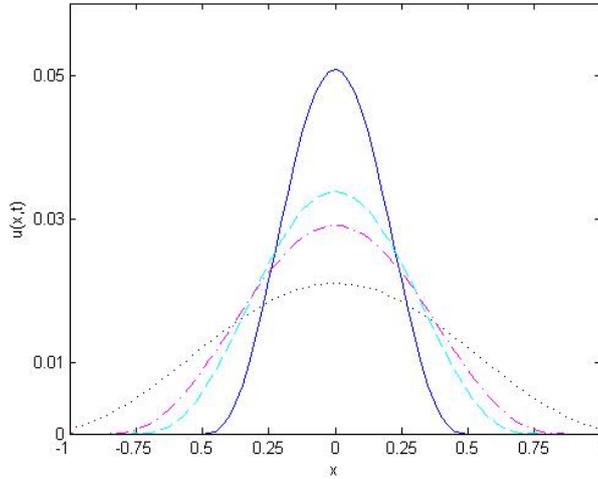


Figure 5.2.2: Distribution of the sixth order self similar solution

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

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

5.3 Scale Invariance of the Numerical Method

In [13] Budd and Piggott propose that a numerical method to discretise a scale invariant PDE should reflect the scale invariance properties of the PDE, i.e. the method should possess discrete self similar solutions which are also scale invariant. From equation(), $xt^{-\frac{1}{n+4}}$ and $ut^{\frac{1}{n+4}}$ are invariant, so it would be hoped that these variables were also invariant under the numerical method approximating this solution. To investigate the behaviour of the variables, plots were constructed of the variable with time. If these variables are indeed invariant, we would expect there to be no change in their values over time. Figure (5.3.1) shows the invariant behaviour of these quantities, visible in the straight line solutions, in the fourth order case, which suggests that this moving mesh method has inherited the scaling properties of the

PDE.

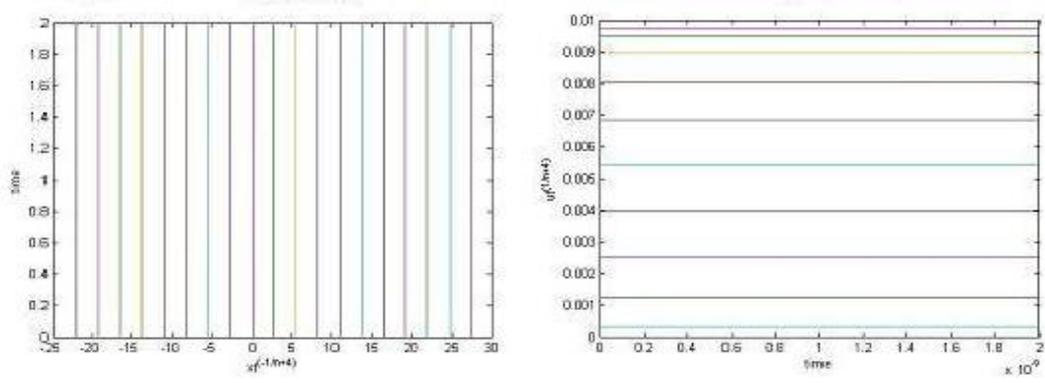


Figure 5.3.1: Invariance of the fourth order solution and mesh for 41 node mesh

Chapter 6

Numerical Results for the Self Similar Solution

The numerical results presented here are compared with the similarity solutions presented in chapter 5, in order to evaluate the accuracy of the moving mesh method, in approximating nonlinear diffusion. Before the program was run for the solutions, some initial investigations were carried out. These were done in order to see if the numerical solution was converging to the true solution, as the number of mesh nodes increased. Further investigations were used to find the largest possible timestep that could be used before tangling of the mesh started to occur.

Figures (6.0.1) and (6.0.2) clearly show the convergence of the numerical solution to the true solution, as the number of nodes in the mesh increased. However, investigation of the timestep showed that increasing the number of nodes in the mesh, resulted in node overtaking unless the timestep was decreased, suggesting that the timestep is proportional to the spatial scales in the mesh. This is clearly seen in Figure (6.0.3), where for all four cases the program was run for 100000 timesteps.

In the next two sections, results are presented for the self-similar solutions described in Chapter Five, for a diffusion coefficient $n = 1$, both show

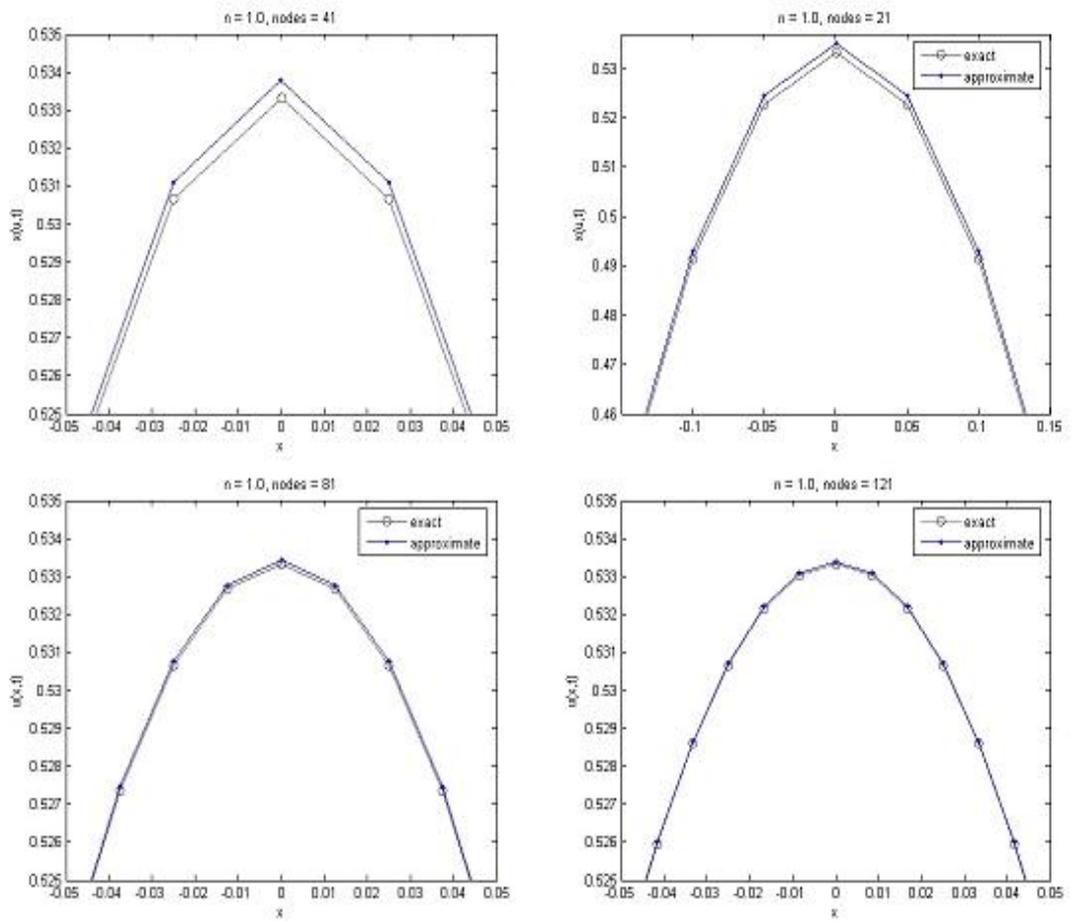


Figure 6.0.1: Convergence of numerical solution to true solution with increasing nodes, $t = 0.0005$

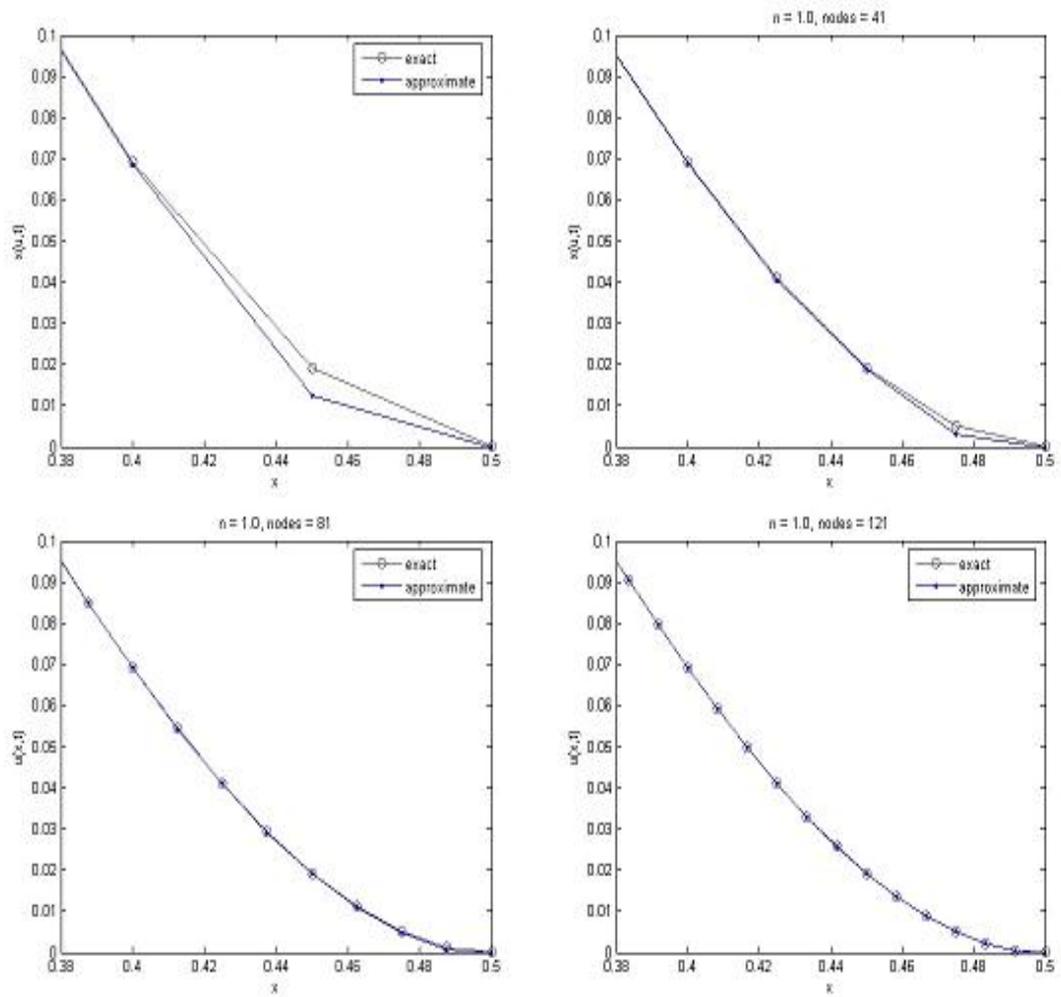


Figure 6.0.2: Resolution of numerical solution at boundary with increasing nodes, $t = 0.0005$

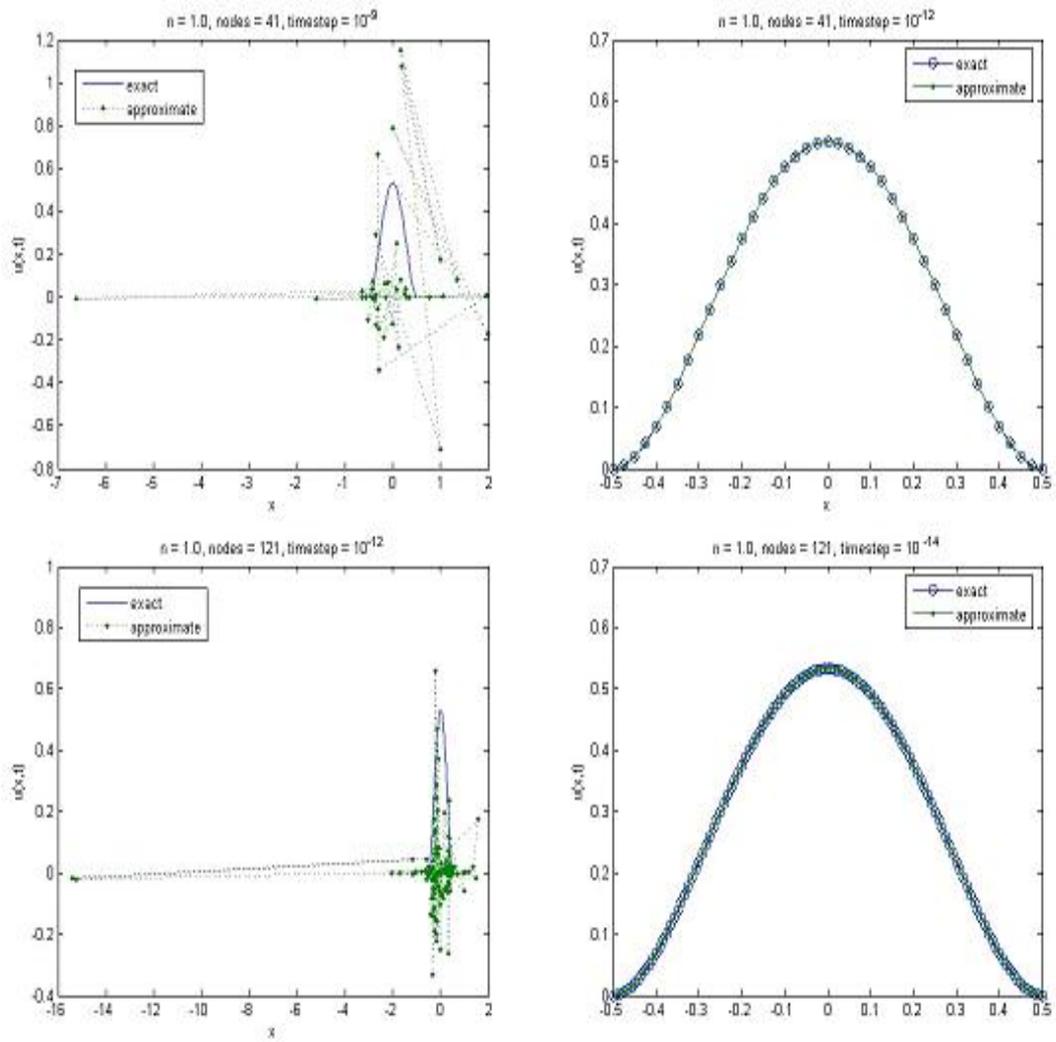


Figure 6.0.3: Timestep dependence on the number of nodes in the mesh

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 $t = 0.0005$, $t = 0.0025$, $t = 0.001$, for $n = 1.0$, on a 41 node mesh

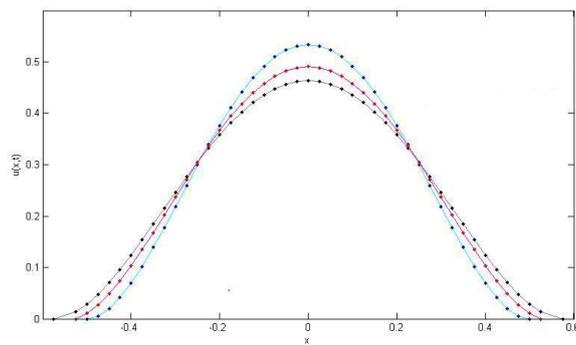


Figure 6.1.1: Exact and approximate solutions for the fourth order problem.

6.2 Sixth Order Results

Dots represent approximate solution, and the line represents the exact solution. Solutions are computed at $t = 0.0005$, $t = 0.0002$, $t = 0.0001$, for $n = 1.0$, on a 41 node mesh

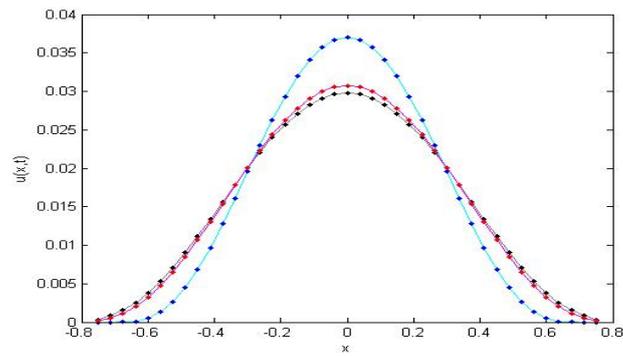


Figure 6.2.1: Exact and approximate solutions for the sixth order problem.

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

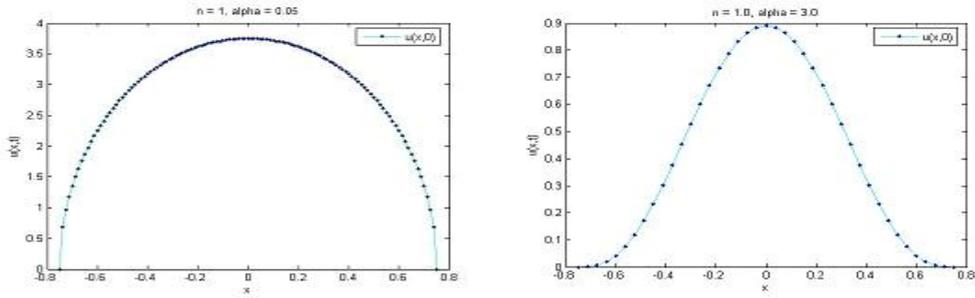


Figure 7.0.1: Change in contact angle for different alpha

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

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x} \left(u^n \frac{\partial^3 u}{\partial x^3} \right) \quad (7.1.1)$$

with

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

and

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

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.

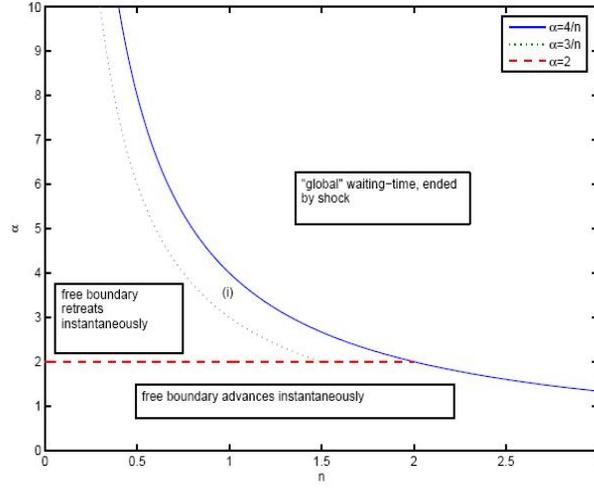


Figure 7.1.1: A summary of the possible small-time behaviours with respect to n and α

7.2 Sixth Order Case

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x} \left(u^n \frac{\partial^5 u}{\partial x^5} \right) \quad (7.2.1)$$

with

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

and

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

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

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

Although no analytic solution is known for this data, a fixed mesh finite element method, [8], exists, for the fourth order case, for which convergence of the approximation to a weak solution, in 1D has been proved. Solutions obtained with this scheme have been used to demonstrate the convergence of the moving mesh solutions for the initial data (8.0.1), in [7].

For the purposes of this dissertation, the investigations are concerned with the behaviour of the moving boundary $b(t)$. The moving mesh method was applied to (8.0.1), for various combinations of n and α , to test the behaviour of the free boundary, in relation to the conjectures discussed in 7.1 and 7.2. Results are presented for certain cases.

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 different values of α are shown in Figure (8.1.1). The stepping seen in the plots is a result of the fact that the numerical approximation to the boundary is calculated on a discrete mesh.

In both cases the moving boundary advances, and the results suggest that for larger α , the speed of the advance is slower. This is better seen in Figure (8.1.2), where the change in the speed of advance for increasing α is clearly seen.

For the advancing case, results in [10] suggest that the free boundary will advance with an unbounded initial velocity, the slow over time. This has not been seen in the results presented here, which show a linear behaviour of the initial velocity. This difference in results may be due to the extremely small timescale over which it has been possible to run this program, so the initial velocities appear linear, whereas their long term behaviour is not. These cases were run using the FORTRAN program to see if this initial unbounded velocity did occur, and this is shown to be the case in Figure

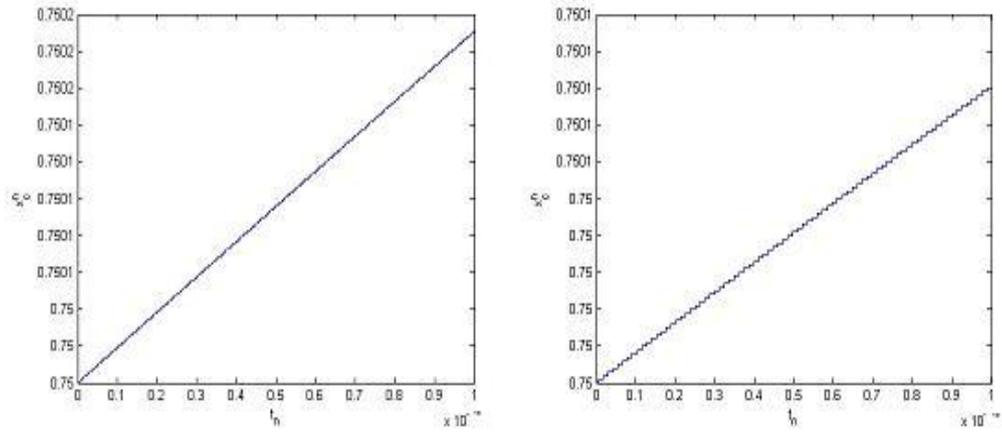


Figure 8.1.1: Advancing moving front with $n = 2.5$, $\alpha = 0.5$, and $n = 2.5$, $\alpha = 1.4$

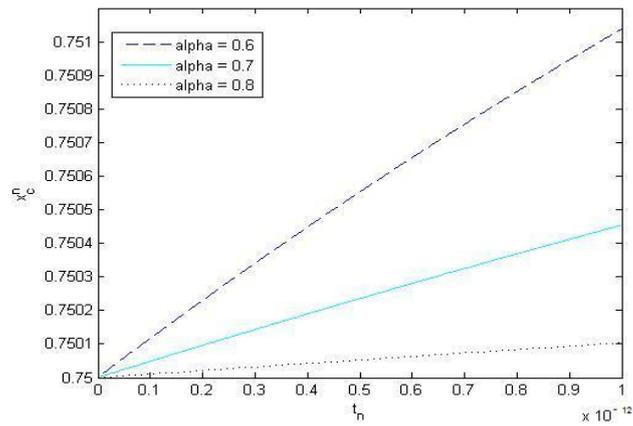


Figure 8.1.2: Advancing moving front for $n = 1.0$, $\alpha \in (0.5, 1.0)$

(8.1.3).

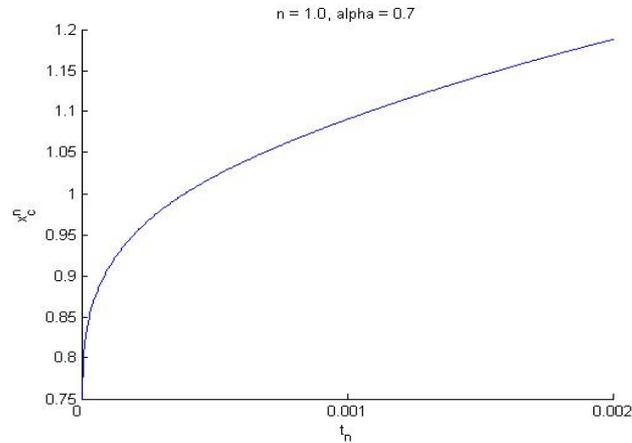


Figure 8.1.3: Advancing moving front for $n = 1.0$, $\alpha = 0.7$, run to final time 0.002

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, different 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.

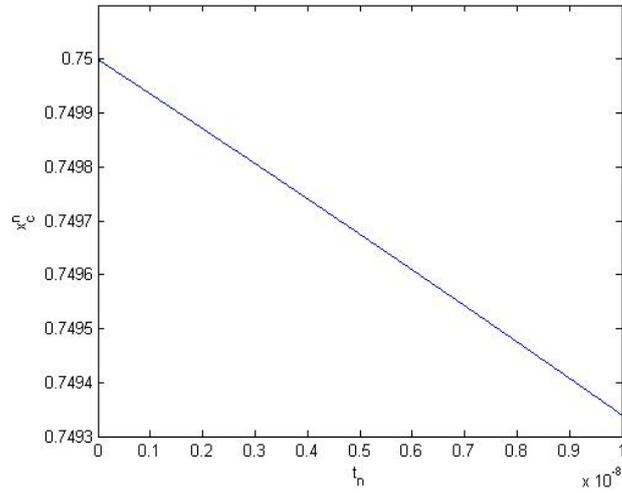


Figure 8.1.4: Retreating moving front for $n = 1.0$, $\alpha = 2.5$

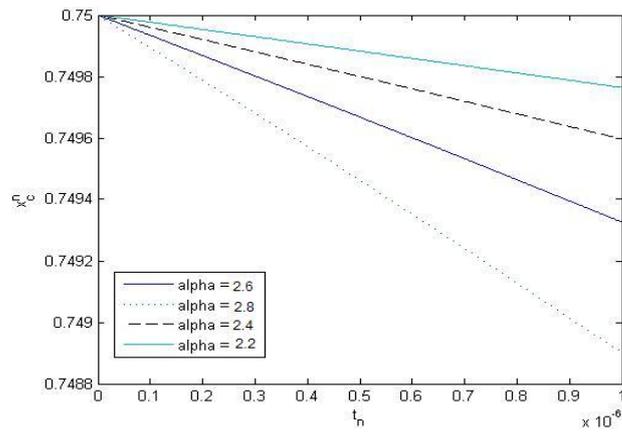


Figure 8.1.5: Retreating moving front for $n = 1.0$, $\alpha \in (2.0, 3.0)$

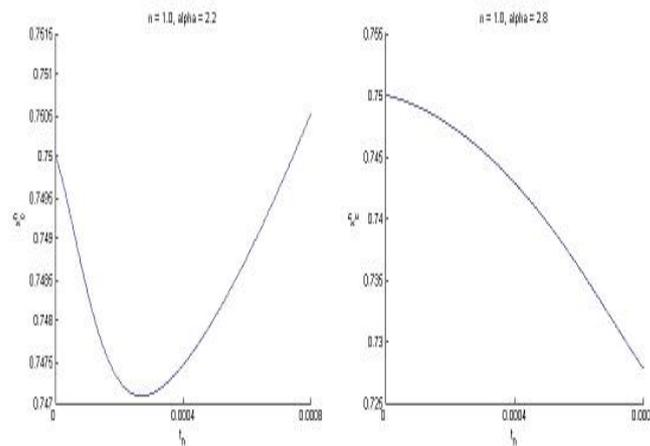


Figure 8.1.6: Retreating moving front for $n = 1.0$, $\alpha = 2.2$, and $n = 1.0$, $\alpha = 2.8$

8.1.3 Waiting Time Behaviour

Due to the limitations on the final time the program can be run to, the results in Figure (8.1.7) are not conclusive as to the waiting time properties for these combinations of n and α . The results seem to indicate that the solution is waiting, but taking the very small timescale over which the program is run, it could be that the moving boundary is advancing or retreating very slowly. These cases were run using the FORTRAN code to elicit more information of their behaviour for longer t . The results can be seen in Figure (8.1.8).

In Figure (8.1.8), the real waiting time behaviour for the given values of n and α can be seen. They show that the solution waits for a time, before the moving boundary suddenly starts to advance, as predicted in the conjectures .

8.2 The Sixth Order Case

For the sixth order case, as previously mentioned, we compare our results to those in [22]. Only a selection of the results obtained are reproduced here,

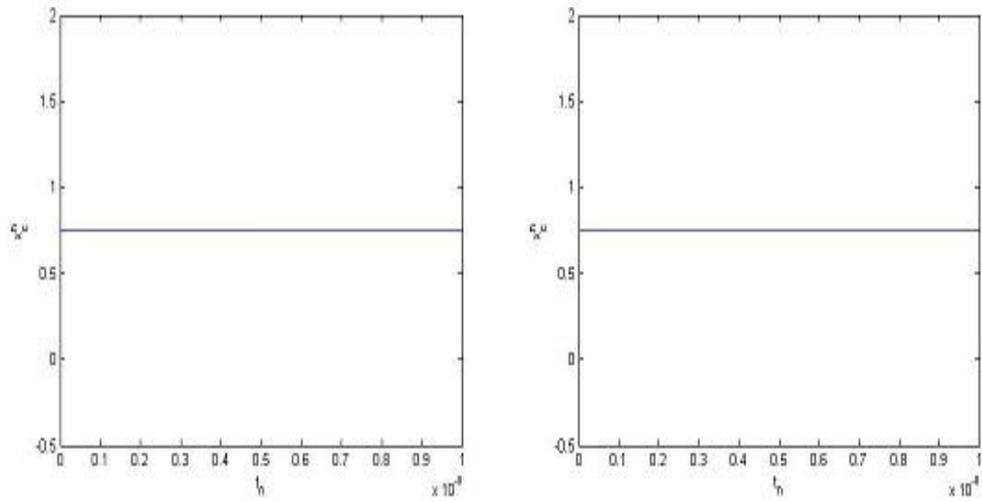


Figure 8.1.7: Apparent waiting time behaviour, for different n and α

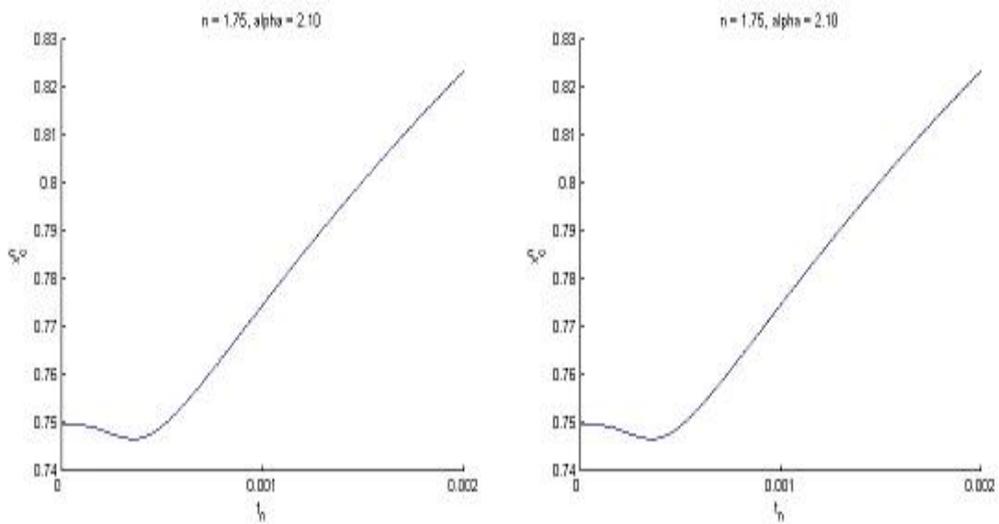


Figure 8.1.8: Actual waiting time behaviour, for different n and α

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

alpha	n=0.5	n=1.0	n=1.5	n=2.0	n=2.5	n=3.0
0.5	ADV	ADV	ADV	ADV	ADV	ADV
1.0	ADV	ADV	ADV	ADV	ADV	ADV
1.5	ADV	ADV	ADV	ADV	ADV	ADV
2.0	ADV	ADV	ADV	ADV	ADV	ADV
2.5	RET	RET	RET	RET	RET	RET
3.0	RET	RET	RET	RET	RET	RET
3.5	RET	RET	RET	RET	RET	RET

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

It is clear that the behaviour suggests that a change is occurring for $\alpha \in (2, 2.5)$. 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 timesteps 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 α 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.

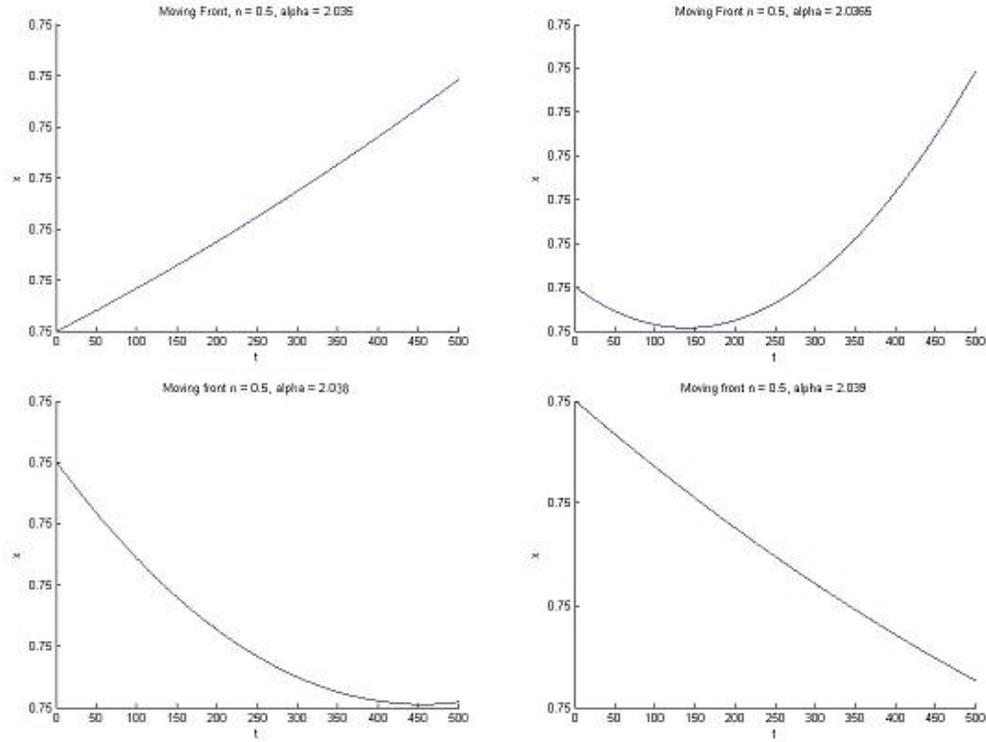


Figure 8.2.2: Moving front results for $n = 0.5$

Following these investigations, further experiments were carried out, this time looking at large values of n . Solutions obtained in [22] for large n are characterised by the formation of humps immediately in the vicinity of the free boundary, and once they have appeared they progress slowly away from the free boundary. Figures (8.2.4), (8.2.5), and (8.2.6) reproduce the results obtained on the moving mesh for $n = 5, 7, 9$, respectively. Figure (8.2.4) clearly shows the presence in the moving front of these humps, for $n = 5, \alpha = 0.1$. In both other cases of n , for $\alpha = 0.1$, these humps are not visible as we would expect. However, in plots of the profile of the solution, we can see the humps clearly. This is not the case when $\alpha = 0.5$, where some small humps

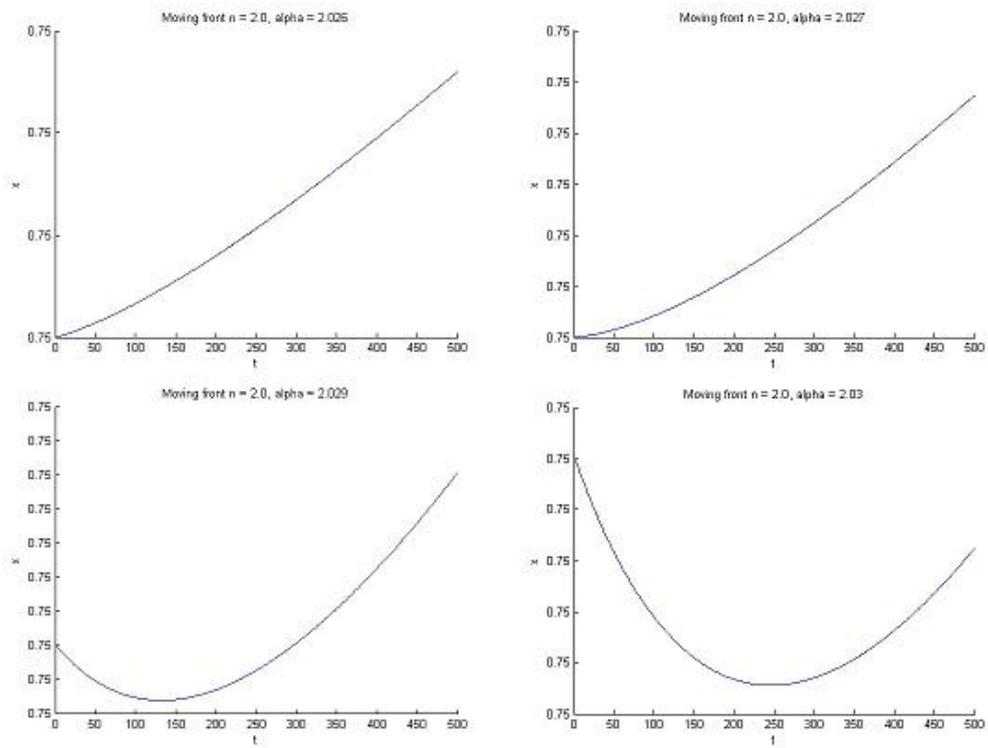


Figure 8.2.3: Moving front results for $n = 2.0$

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 continuously. 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].

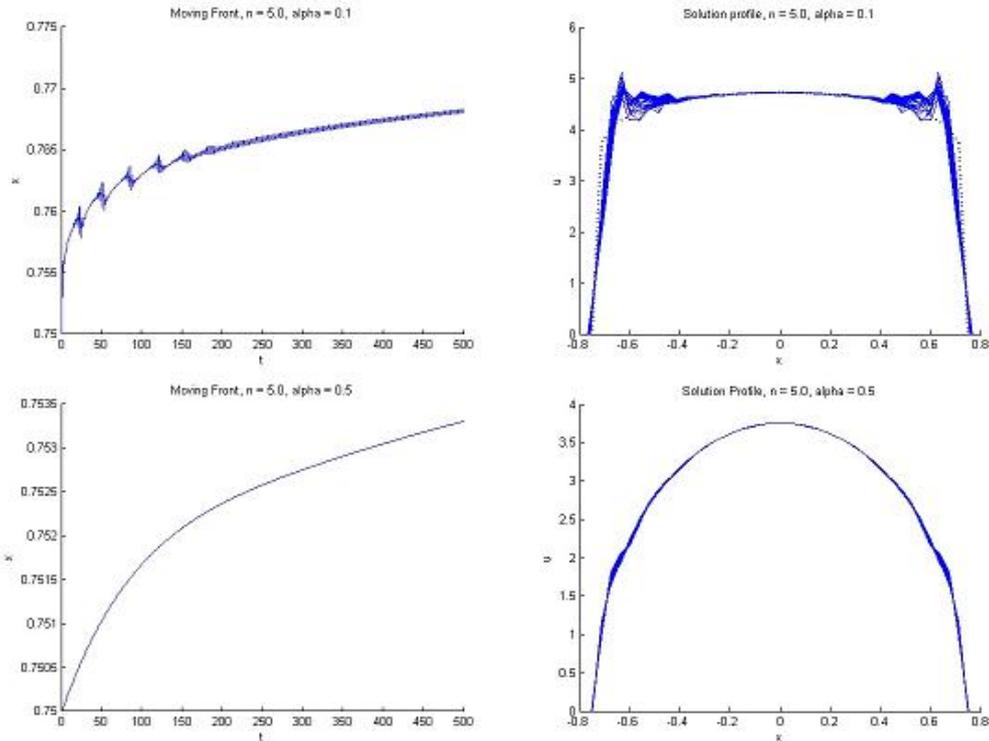


Figure 8.2.4: Behaviours observed in the case $n = 5.0$

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

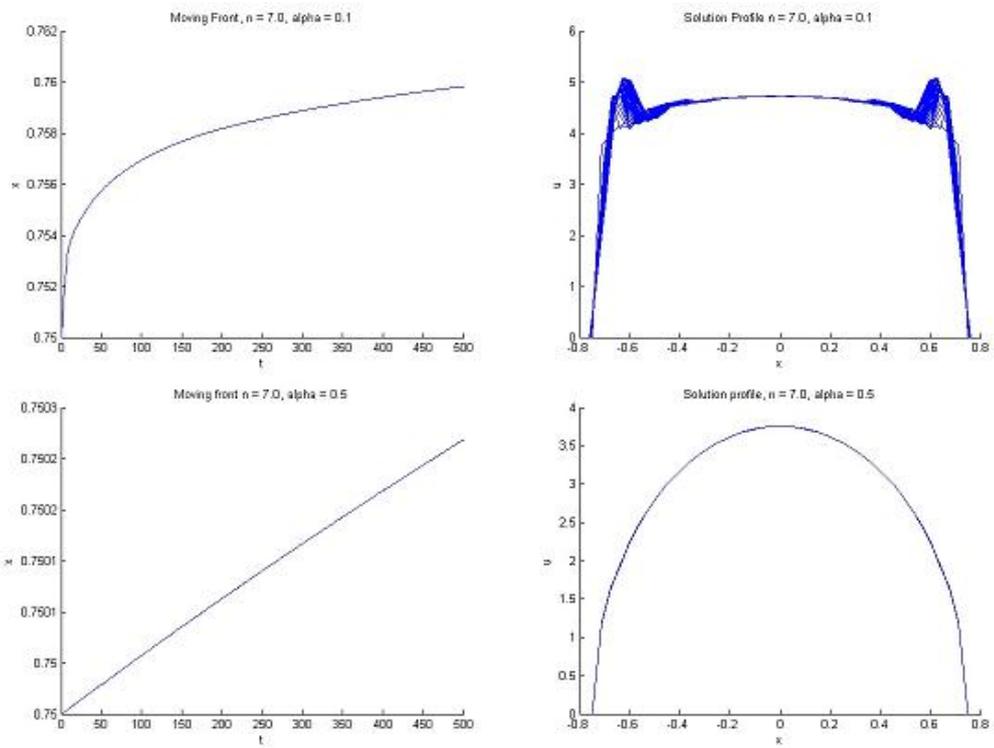


Figure 8.2.5: Behaviours observed in the case $n = 7.0$

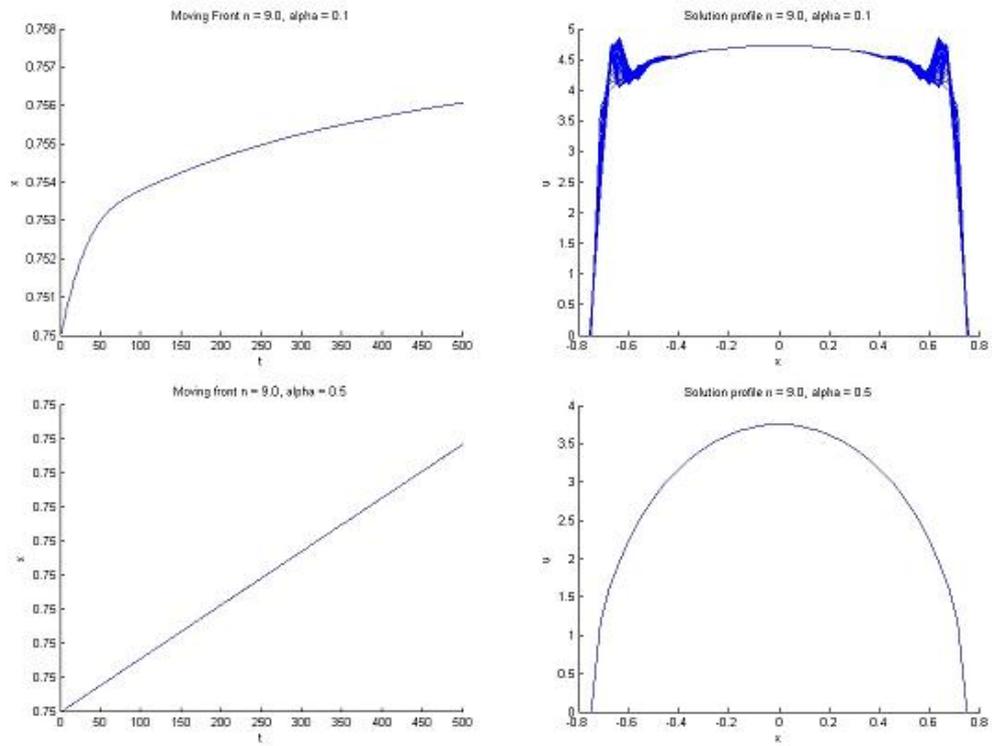


Figure 8.2.6: Behaviours observed in the case $n = 9.0$

in the cases $\alpha = 0.1$, 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.

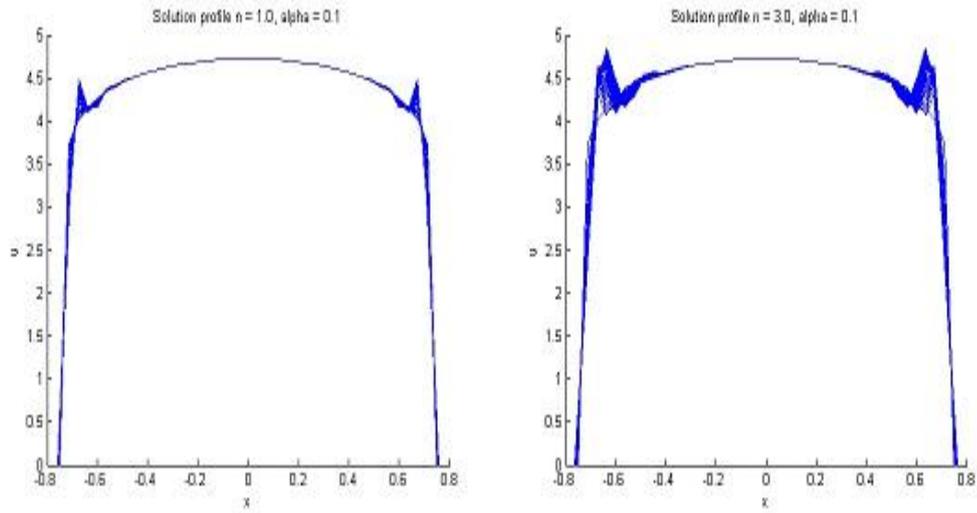


Figure 8.2.7: Humps in the solutions for $n = 1.0$, and $n = 3.0$

Chapter 9

Conclusions and Further Work

9.1 Summary

In this chapter, the work carried out in this dissertation is summarised. We then discuss some of the results and findings made, and suggest possible improvements to this work. We then discuss possible avenues for further research.

In this dissertation, a moving mesh method based on a conservation of mass principle has been implemented in order to generate numerical solutions to the fourth and sixth order nonlinear diffusion equations.

In Chapter Two some of the applications of nonlinear diffusion were considered. In Chapter Three, the principles of velocity based moving mesh methods were described in terms of generating a mapping from a computational domain to a physical domain, and then differentiating this mapping with respect to time, to induce a motion on the mesh. In Chapter Four a moving mesh method based on a conservation principle was derived, and a finite element formulation constructed for its solution. Chapter Five discussed the property of scale invariance in relation to the nonlinear diffusion

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 introducing a new variable, $\sigma = t^\beta$, and transforming (9.2.1), under this gives a scale invariant time-stepping scheme

$$X_{n+1} = X_n + \beta^{-1}((t_{n+1})^\beta - (t_n)^\beta)t_n^{1-\beta}F(X_n) \quad (9.2.4)$$

Future work could look at implementing this timestepping scheme, in order to make the numerical scheme truly invariant.

9.2.2 Timestepping

When considering the size of the timesteps used, it was seen in Chapter Seven, that the timestep associated with the moving mesh, was proportional to the spatial mesh size. A larger number of nodes required a smaller timestep in order to avoid node overtaking. In regions where the solution of the underlying PDE has steep solution gradients, it would be expected that the moving mesh method would generate smaller mesh sizes in space. A possible avenue for further work could be to implement some form of locally adaptive timestepping in these regions.

Another possibility is to consider the implementation of a different timestepping scheme, for example a Runge-Kutta, or Backward Differentiation scheme, both of which possess better stability intervals than the Forward Euler Scheme. This may allow larger timesteps to be taken.

9.2.3 Errors

The results obtained in this dissertation, have been evaluated only on a qualitative basis. It is important to point out that owing to time constraints, no error measures were calculated for the self-similar results obtained, so strictly speaking the accuracy of the solutions presented here cannot be guaranteed. However, the same moving mesh method was implemented in

[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 possibility would be to consider the problem of oxygen diffusion in tumour growth, and apply the method considered here to a two dimensional radial problem, which would better model the distribution of a tumour.

Bibliography

- [1] D.G.Aronson. Some Problems in Nonlinear Diffusion, Editors: A.Fasano, M.Primiceno, *Lecture Notes in Mathematics*,1224 Springer-Verlag, New York (1986)
- [2] D.G.Aronson, H.F.Weinberger. Nonlinear Diffusion in Population Genetics and Nerve Pulse Propagation, Partial Differential 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 Diffusion 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 Differential 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 Differential Equations in Two Dimensions. *Applied Numerical Mathematics*,56 (2006), 230-252

-
- [7] M.J.Baines, S.Langdon. Fixed Versus Moving Finite Element Methods for Fourth Order Nonlinear Diffusion, *In Preparation*,
- [8] J.W.Barrett, J.F.Blowey, H.Garcke. Finite Element Approximation of a Fourth Order Nonlinear Degenerate Parabolic Equation, *Numerische Mathematik*,80 (1998), 525-556
- [9] J.W.Barrett, S.Langdon, R.Nurnberg. Finite Element Approximation of a Sixth Order Nonlinear Degenerate Parabolic Equation, *Numerische Mathematik*,
- [10] J.F.Blowey, J.R.King, S.Langdon. Small and Waiting Time Behaviour of the Thin Film Equation, *Reading University Numerical Analysis Report 9/06*
- [11] C.De Boor. Good Approximation by Splines with Variable Knots II, *Springer Lecture Notes Series 363*, 1973
- [12] C.J.W.Beward, H.M.Byrne, and C.E.Lewis. Modelling the Interactions Between Tumour Cells and a Blood Vessel in a Microenvironment within a Vascular Tumour, *European Journal of Applied Mathematics*,12 (2001), 529-556
- [13] C.J.Budd, M.D.Piggott. The Geometric Iteration of Scale Invariant Ordinary and Partial Differential Equations, *Journal of Computational and Applied Mathematics*,128 (2001), 399-422
- [14] W.Cao, W.Huang, and R.Russell. Approaches for Generating Moving Adaptive Meshes : Location versus Velocity, *Applied Numerical Mathematics*,47 (2003), 121-138
- [15] W.Cao, W.Huang, and R.Russell. A Moving Mesh Method Based on the Geometric Conservation Law, *SIAM Journal of Scientific Computation*,24 (2002), 118-142

-
- [16] J.Crank, R.S.Gupta. A Moving Boundary Problem Arising from the Diffusion of Oxygen in Absorbing Tissue, *Journal of Institute of Mathematical Applications*,10 (1972), 19-33
- [17] J.C.Flittton and J.R. King. Moving Boundary and Fixed Domain Problems for a Sixth-Order Thin Film Equation, *European Journal of Applied Mathematics*,15 (2004), 713-754
- [18] S.D.Howison, D.F.Mayers and W.R.Smith. Numerical and Asymptotic Solution of a Sixth Order Nonlinear Diffusion Equation and Related Coupled Systems, *IMA Journal of Applied Mathematics*,57 (1996), 79-98
- [19] R.Huang, Z.Suo. Wrinkling of a Compressed Elastic Film on a Viscous layer, *Journal of Applied Physics*,91 (2002), 1135-1142
- [20] W.Huang, Y.Ren and R.Russell. Moving Mesh Method Partial Differential Equations Based on the Equidistribution Principle, *SIAM Journal of Numerical Analysis*,31 (1994), 709-730
- [21] J.Hulshof. Some Aspects of the Thin Film Equation, *Proceeding of the European Congress in Mathematics, Volume II*,202, Birkhauser-Verlag, (2001), 291-301
- [22] S.Langdon. Numerical Results for The Sixth Order Problem, *In Preparation*
- [23] S.Langdon. Private Communication,
- [24] J.A.Moriarty, E.L.Terrill. Mathematical Modelling of the Motion of Hard Contact Lenses *European Journal of Applied Mathematics*,7 (1996), 575-594
- [25] J.D.Murray. Mathematical Biology II: Spatial Models and Biomedical Applications, Third Edition, Springer-Verlag, Berlin, 2002

-
- [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 Diffusion *IMA Journal of Applied Mathematics*,40 (1988), 73-86