UNIVERSITY OF READING

DEPdbHDf7NTREADING

Abstract

In this thesis we study the use of a velocity-based moving-mesh numerical method, driven by conservation, for obtaining approximate solutions to nonlinear di usion equations of second, fourth and sixth-order in one dimension. These problems often have moving boundaries and possess many properties that we are able to conserve using the method, such as scale-invariance and conservation of mass.

A key feature of the method is that it possesses the ability to propagate similarity solutions forward in time to within rounding error (termed the S Property). This property, occurring when a scale-invariant time stepping scheme is used and when the

Declaration

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

Nicholas Bird

Acknowledgements

Firstly, my sincerest thanks go to my supervisors for their excellent guidance and support throughout this project. Thank you Dr. Steve Langdon for treating every piece of writing as part of my thesis and honing my precision as a result. It de nitely helps in the long run. Thanks also to Prof. Mike Baines for the near-unlimited ideas and enthusiasm for the work that I have done. Together you have kept me positive when things were not going as well as they could and encouraged me at all other times, which has made the project one I will never forget.

I would also like to acknowledge the funding of both the University of Reading and the Engineering and Physical Sciences Research Council (ESPRC) for enabling this journey to take place.

Particular thanks go to the Mathematics department at Reading for providing a great environment to work in over the past few years. All the various o ce-mates I have had made the process much more interesting, as have all the other PhD students and lecturers that I have been able to talk with (even if just over a cup of tea or two).

Thanks also to my friends and family for being there, asking me how my work was going even if you had no idea what I was talking about most of the time. I hope I've made you proud.

Finally, but most importantly, I need to thank my wife Suzanne for her love and support. Thank you for keeping me sane and putting up with me through the good times and the not-so-good times. We got to the other side of it together and without her I would not even be here. Gratitudes!

Contents

1 Introduction 1

1.1 Motivation 1

2.6 Numerical Methods in Higher Dimensions 32
2.6.1 Meshless Methods 33
2.6.2 MMPDE Methods in 2D 33
2.6.3 The Parabolic Monge-Ampère Method 34
2.7 Summary of this Chapter 35

3 Velocity-Based Moving Mesh Methods 36
3.1 Aims of this Chapter 36
3.2 Velocity Based Methods 36
3.2.1 Moving Finite Element Method 37
3.2.2 The GCL Method 38
3.2.3 The Deformation Map Method 40
3.2.4 A Conservation-Based Method 40
3.2.5 Local Conservation of Mass 41

3.5.4 Mesh Movement and Solution Recovery 53
3.6 The S Property 54
3.7 Summary of this Chapter 54

4 An Implementation of the MPCM Possessing the S Property 55
4.1 Aims of this Chapter 55
4.2 The Fourth-Order Problem with $n = 1$ 55
4.3 The S Property in the MPCM for the Fourth-Order Problem 56
4.3.1 Approximating $q(x;s)$ 57
4.3.2 The Nodal Velocity 60
4.3.3 Updating the Nodal Positions 61
4.3.4 Updating the Solution Values 62
4.3.5 Implications 63
4.3.6 Numerical Results 64
4.4 Propagation of Rounding Error 70
4.4.1 Explaining the Additional Multiple Time Step Error 70
4.4.2 Examining the Time Stepping Scheme 73
4.4.3 Bounding the Error 74
4.5 The S Property in the MPCM for the Second-Order Problem 76
4.5.1 Numerical Results 77
4.6 The S Property in the MPCM for the Sixth-Order Problem 83
4.6.1 Approximating $p(x;s)$ 84
4.6.2 Approximating $q(x;s)$ 87
4.6.3 Mesh Movement 88
4.6.4 Solution Recovery 88
4.7 Summary of this Chapter 88

5 A Finite Element Implementation of the MPCM 90
5.1 Aims of this Chapter 90
5.2 Comparison between the MPCM and FEMPCM 91

5.3 Weak Forms 92
5.3.1 Weighted Conservation of Mass 92
5.3.2 Additional Weak Forms 94
5.4 An Implementation of the FEMPCM for the Fourth-Order Problem . . 95
5.4.1 Finite Dimensional Subspaces 95
5.4.2 The Finite Element Approximation 95
5.4.3 Updating the Nodal Positions 96
5.4.4 Recovering the Solution $U(x;t)$ 96
5.5 The FEMPCM with Piecewise Linear Basis Functions 96
5.5.1 Basis Functions 97

5.8.2 An Implementation of the FEMPCM for the Sixth-Order Problem with $n = 1$ 130
5.9 Summary of this Chapter 135

6 The MPCM in the Fourth-Order Problem with $n > 1$ 137
6.1 Aims of this Chapter 137
6.2 The 4th Order Problem with $n > 1$ 138
6.3 Boundary Velocities when $n > 1$ 139
6.4 Initial Boundary Velocities 140
6.4.1 $n = 3$ 141
6.4.2 $n > 3$ 141
6.4.3 $n < 3$ 141
6.5 Small-Time Behaviours of the Boundary 142
6.5.1 Initial Behaviour 142
6.5.2 Evolution of the Solution Close to the Boundary 143
6.6 Initial Boundary Velocities using the MPCM 143
6.6.1 Numerical Veri cation of Boundary Velocity Issues 144
6.6.2 Initial Boundary Values of $q(x;t)$ 148
6.6.3 Initial Value of $\frac{@q}{@x}$ in the Vicinity of the Boundary 149
6.6.4 Consequences for the MPCM 150
6.7 The FEMPCM when $n > 1$ 150
6.7.1 Finite Advance 151
6.7.2 Finite Retreat 153
6.7.3 Waiting-Time 154
6.7.4 Initial Boundary Velocities 155
6.7.5 Critique of the FEMPCM with $n > 1$ 158
6.8 Mesh Point Distribution 159
6.8.1 Uniform Initial Mesh 159
6.8.2 Equidistribution of Mass 161
6.8.3 Equidistribution of Arc Length 162

6.9 A Hybrid Numerical Method 162
6.9.1 The Fixed-Mesh Method 163
6.9.2 De ning the Numerical Interface 165
6.9.3 Convergence of the Fixed-Mesh Solution 166
6.9.4 Use of the MPCM 167
6.9.5 Critique of the Hybrid Method 167
6.10 An Alternative Expression for the Velocity 168
6.10.1 Approximating the Alternative Velocity 172
6.10.2 Provisional Results 173
6.10.3 The Alternate Velocity near the Boundary as N increases 177
6.10.4 The Alternative Velocity when n = 1 177
6.11 Summary of this Chapter 178

7 Conclusions and Further Work 180
7.1 Summary 180
7.2 Conclusions 184
7.3 Future Work 187

List of Notation

Notation related to analytical functions

$_T$ Domain upon which solutions are sought, both spatial and time dependent.

$\Omega(t)$ Spatial domain, typically a bounded subset of $\mathbb{R}$.

$\hat{\Omega}(t)$ Subregion of $\Omega(t)$ with boundary $\partial\hat{\Omega}(t)$.

$a(t)$ Left-hand moving boundary of $\Omega(t)$.

$b(t)$ Right-hand moving boundary of $\Omega(t)$.

$u(x;t)$ Solution of a nonlinear PDE.

$q(x;t)$ Auxiliary function, related to $u(x;t)$ depending upon order of PDE considered.

$p(x;t)$ Auxiliary function, related to $u(x;t)$ and $q(x;t)$ depending upon order of PDE considered.

$v(x;t)$ Deformation velocity of $\Omega(t)$.

v_b Boundary velocity, typically at $x = b(t)$.

$z(x;t)$ Velocity potential.

t Time variable.

x Spatial variable.

$\hat{x}(t)$ Arbitrary point in $(\quad t)$.

s Scaled time variable, related to t.

m Constant determining the order of a PDE.

n Di usion coe cient.

f Scaled variable, where f denotes a generic variable.

Scaling parameter.

$;\ ;$ Scaling constants.

$;\ ;$ Similarity Variables.

$f^S(x;t)$ Similarity function, where f denotes a generic function. In particular, $u^S(x;t)$ denotes the similarity solution to a nonlinear PDE.

$;\ !$ Constants in similarity solution.

Mass of solution.

$c(\hat{x})$ Partial mass such that $c(\hat{x})$.

Constant used in initial conditions in chapter 6.

Notation related to numerical solutions

$F(x;t)$ Approximation to the analytical function $f(x;t)$.

t^k Discrete time point in the discretisation of time domain.

t Constant time step related to t.

s^k Discrete time point in the discretisation of scaled time domain.

s Constant time step related to s.

X_i^k Mesh point in spatial discretisation of $(\ t)$, at time t^k.

X^k Set of mesh points at time t^k.

N Number of nodes in mesh X^k.

K Total number of time steps.

F_i^k Approximation to function $f(x;t)$ at node i and time point t^k.

F^k Set of points F_i^k.

$_i$ Partial mass of solution centred at node i of mesh.

Set of partial masses.

V^k Approximation to $v(x;s)$ at s^k in terms of scaled time variables.

$w(x;t)$ Test function in weak formulation or finite element method.

$_i^k$ Truncation error of finite difference scheme at node i, time t_k (also $\tilde{}_i^k$, $\hat{}_i^k$).

$(x;t)$ Sextic Lagrange polynomial basis function.

$(x;t)$ Quartic Lagrange polynomial basis function.

$'(x;t)$ Quadratic Lagrange polynomial basis function.

$(x;t)$ Linear Lagrange polynomial basis function.

E_N^F l^1 error of F^k on a mesh of N nodes.

E_N^B Relative error in boundary node position on a mesh of N nodes (typically right-hand boundary $x = b(t)$).

List of Figures

2.1 Image highlighting the difference between fixed and moving domains Ω_T. The left-hand image shows a fixed-mesh domain, with interfaces given by the black asterisk. The solution outside the interfaces exists but is equal to the value at the interface. The right-hand plot shows a moving domain, where the black asterisk denotes the boundary of the Ω_T. The solution is not defined outside of the boundary. 28

2.2 Example of the type of initial condition considered in Giacomelli et al. (2008) with $s_0 = 2$. 32

4.1 Absolute error in the MPCM over a single time step. The top left window contains the absolute error in U^1 ($O(10^{-17})$), the top right window the error in Q^0 ($O(10^{-15})$), bottom left the error in V^0 ($O(10^{-14})$) and bottom right the error in the mesh positions ($O(10^{-16})$). 66

4.2 l^1 error in the approximation U^k from the MPCM over the time window $s \in [1; 2.5]$. Scale of y-axis is $O(10^{-14})$. 67

4.3 l^1 error in Q^k (top) and V^k (bottom) over the time window $s \in [1; 2.5]$. Vertical scale of the top plot is $O(10^{-13})$, while the vertical scale for the bottom plot is $O(10^{-12})$ 68

4.4 Boundary node error E^B_{21} over the time window $s \in [1; 2.5]$. Vertical scale of plot is $O(10^{-11})$. 69

4.5 l^1 error in the solution when Q, V and X in the method are set equal to the exact value, but the solution is recovered approximately. Scale of y-axis is 10^{-17}. 71
4.6 l^1 error in the solution when Q and V in the method are set equal to the exact value, but the nodes are updated using a scale-invariant time stepping scheme and the solution is recovered approximately. Scale of y-axis is 10^{-13}. 72

5.3 Nodal values of $U(x; 1.25)$ on meshes of $N = 21$ (blue xs), 41 (green os), 81 (red +s) and 161 (cyan s) nodes. 106
5.4 Nodal values of $Q(x; 1.25)$ on meshes of $N = 21$ (blue xs), 41 (green os), 81 (red +s) and 161 (cyan s) nodes. 107
5.5 Nodal values of $V(x; 1.25)$ on meshes of $N = 21$ (blue xs), 41 (green os), 81 (red +s) and 161 (cyan s) nodes. 107
5.6 Position of the right-hand boundary node on meshes of $N = 21$ (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes. 108
5.7 Plot of the various choices of Lagrange polynomial used in this implementation of the Finite Element Method, plotted over a single element. . . . 111
5.8 Absolute error in the FEMPCM over a single time step. The top left window contains the absolute error in U^1 ($O(10^{-15})$), the top right window the error in Q^0 ($O(10^{-13})$), bottom left the error in V^0 ($O(10^{-13})$) and bottom right the error in the mesh positions ($O(10^{-16})$). 124
5.9 Evolution of the solution for the FEMPCM described in §5.6 for $s \in [1; 1.25]$ using a mesh of $N = 21$ nodes. 125
5.10 L^2 error in the solution for the quartic implementation of the FEMPCM, plotting over the time window $s \in [1; 1.25]$ on a mesh of 21 Nodes. Scale of y-axis on plot is 10^{-13}. 126
5.11 L^2 error in $Q(x; s)$ (top plot) and $V(x; s)$ (bottom plot) for the quartic implementation of the FEMPCM, plotting over the time window $s \in [1; 1.25]$ on a mesh of 21 Nodes. Scale of y-axis on both plots is 10^{-13}. . 127

6.1 The computed velocity V_i^0 plotted over the final 20 mesh points for the $n = 6.5$, $= 5.2$ finite retreat case. 148
6.2 Boundary trajectories for the finite advance experiment. Shown here are trajectories for meshes of $N = 21$ (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes. 152

6.3 Boundary trajectories for the nite retreat experiment. Shown here are trajectories for meshes of N = 21 (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes. 154
6.4 Boundary trajectories for the waiting-time experiment. Shown here are trajectories for meshes of N = 21 (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes. 156
6.5 Computed velocity in the rst time step for the nite advance experiment with N

List of Tables

6.1 Initial boundary velocities for the nite advance experiments in x6.6.1. The expected boundary velocity is given in the nal row for each case. . 146

6.2 Initial boundary velocities for the nite retreat experiments in x6.6.1. The expected boundary velocity is given in the nal row for each case. 146

6.3 Initial boundary velocities for the waiting-time experiments in x6.6.1. The expected boundary velocity is given in the nal row for each case. 147

6.4 Absolute di erence of mesh points and solution values for the nite advance experiment at coinciding points 153

6.5 Absolute di erence of mesh points and solution values for the nite retreat experiment at coinciding points 155

6.6 Absolute di erence of mesh points and solution values for the waiting-time experiment at coinciding points 156

6.7 Initial Boundary Velocities for the experiments in x6.7.1{6.7.3. The expected boundary velocity is given in the nal row for each case. 157

6.8 Initial boundary velocities for the nite advance experiments using the alternative velocity. The expected boundary velocity is given in the nal row for each case. 174

6.9 Initial boundary velocities for the nite retreat experiments using the alternative velocity. The expected boundary velocity is given in the nal row for each case. 174

6.10 Initial boundary velocities for the waiting-time experiments using the alternative velocity. The expected boundary velocity is given in the nal row for each case. 175

Chapter 1

Introduction

1.1 Motivation

There are a number of physical processes which correspond to nonlinear di usion problems when modelled mathematically (see, e.g., Barrett et al. (2004)). These problems can be of various orders, but those of particular interest in this thesis are those of high order (fourth and sixth-order), although we also consider the second-order Porous Medium Equation (PME). Mathematically, problems of this type consist of a partial di erential equation (PDE), together with an initial condition and suitable boundary conditions.

The high-order nonlinear di usion problems considered in this thesis have a number of interesting features, the most important of which is that the boundaries of the solution can evolve over time. These moving boundaries add an extra complexity to the problems, since (from a numerical viewpoint) some mechanism is required which can track the position of the boundary.

There exists a large body of literature describing the behaviour of solutions to high-order nonlinear di usion problems. In particular, the degree of nonlinearity of the problem can have an important e ect on the behaviour of the solution close to the moving boundary (see, e.g., Beretta et al. (1995), Bowen and King (2001)). The preservation of nonnegative solutions has also been proven for a range of problems relevant to the work

recovery can be achieved essentially to within rounding error using a nite di erence scheme, in the case where the initial condition coincides with a similarity solution and when a scale invariant time stepping scheme is used.

Both the BHJ method and that of Parker are velocity-based moving-mesh schemes built upon the same conservation strategy. We shall refer to these as conservation-based schemes (or simply the conservation method). This thesis generalises the conservation method and provides an implementation in both nite di erences and nite elements which is able to propagate similarity solutions forward in time to essentially within rounding error. A signi cant feature of the implementations in this thesis is the use of invariant time stepping schemes which preserve scale invariance.

Also of interest is whether such a method is able to accurately model the small-time behaviours of solutions which do not initially correspond to a similarity solution. Particular attention is given to (and comparison made with) the work in Blowey et al.

Produce an implementation of the BHJ method using higher order basis functions and scale-invariant time stepping in order to propagate similarity solutions forward in time to within rounding error. This nite element implementation should be able to approximate solutions to second, fourth and sixth-order problems.

Use these implementations to provide a comparison with the work in Blowey et al. (2007) in order to verify whether the moving-mesh method is able to replicate the small-time behaviours of solutions (computed in that paper using a xed-mesh method). It is hoped that the method can not only model such behaviour, but do so at a reduced computational expense.

The key results of this thesis are as follows:

The results of Parker (2010) are successfully extended to a more general class of second-order nonlinear di usion problems using a conservation-based nite di erence method. The method (termed the Moving Point Conservation Method, or MPCM) can be shown to be possess the S Property, with approximate solutions matching the exact solutions essentially to within rounding error.

The MPCM is extended to approximate fourth-order (written as two second-order equations) and sixth-order (written as three second-order equations) nonlinear di usion problems. By careful construction of the numerical schemes used in the method, the MPCM is shown to possess the S Property for the fourth and sixth-order problems.

Numerical results verifying the possession of the S Property reveal a slow build up of global error in the method (shown to be bounded) when run over multiple time steps.

The BHJ method is implemented using piecewise linear nite elements, with some minor di erences with regards to the solution recovery step.

By selection of appropriate basis functions, a nite element implementation of the MPCM (termed the FEMPCM) for second, fourth and sixth-order problems is

also shown to possess the S Property, with numerical results verifying this for the fourth-order problem.

Study of the fourth-order problem in cases where an explicit similarity solution does not exist highlights the issues which the MPCM can face related to the boundary velocity of the domain (when the fourth-order PDE is written as a pair of second-order equations). The presence of singularities at the moving boundary causes the method to be unable to model a large range of cases where the boundary experiences a nite or zero velocity due to certain functions becoming unbounded at the boundary. The method is also unable to model cases where the boundary initially moves with an instantaneously unbounded velocity.

An investigation into whether the FEMPCM is able to alleviate some of the issues experienced by the nite di erence implementation (for the fourth-order PDE when written as two second-order equations). It is shown that the boundary velocity issues are molli ed in the nite element case, but not eliminated completely.

An alternative expression for the velocity of the domain is proposed (in which the fourth-order PDE is not written as a pair of coupled equations) which alleviates the issues experienced in the instances where the boundary experiences a nite or zero velocity. This is achieved by rewriting the velocity in terms which are smooth and nite at the boundary. Numerical results show that this alternative expression produces approximations to the velocity which are accurate, albeit with some oscillations in the velocity still appearing.

1.3 Thesis Outline

In Chapter 2 we introduce the nonlinear di usion PDEs which will be studied in this thesis. Coupled with appropriate boundary conditions and an initial condition, these PDEs make up the fourth, second and sixth-order problems, of which the fourth-order problem (written as a pair of coupled equations) will form the main focus of the thesis.

A discussion of the existing literature surrounding these problems is presented, detailing known properties and behaviours of the nonlinear PDEs. The properties which are of the most relevance for this thesis are then expanded upon, including mass conservation and similarity solutions.

The chapter ends with a description of existing numerical methods for solving the problems outlined at the beginning of the chapter, drawing upon the large body of literature which exists around this topic. As all work in this thesis in concerned with one spatial dimension we focus on this, but a short description of methods in higher dimensions is provided.

Chapter 3 explores a particular type of moving-mesh numerical method, which forms the basis of the numerical method developed for this thesis. A velocity-based moving-mesh method based on conservation seeks to obtain the velocity of the discretised domain along with the approximate solution to the PDE. This mesh velocity is then used to update the domain over time. Of the di erent velocity-based methods which exist the conservation method is emphasised in the thesis.

The numerical method used for the work in the thesis, the MPCM, is then described, with nite di erence implementations for the various nonlinear di usion problems outlined. An important property which the MPCM may possess is then de ned, the S Property, with scale-invariant time stepping an essential ingredient in the MPCM for this property to be present. If an implementation of the MPCM possesses the S Property, then we are able to obtain extremely accurate solution approximations under certain conditions.

An implementation of the MPCM is described in Chapter 4 for the fourth-order problem. This implementation is shown to possess the S Property upon some modi cation to the basic MPCM. Details on how the MPCM implementation can be modi ed to enable the method to possess the S Property for second and sixth-order problems are also given.

Chapter 5 sees the introduction of the FEMPCM (building upon the BHJ method), starting with the necessary weak forms. It is shown that by carefully selecting the nite-

dimensional subspaces and the basis functions which lie in those spaces, the FEMPCM can possess the S Property.

The cases where there is no explicit similarity solution available (so that the MPCM implementations do not possess the S Property) are explored in Chapter 6. In particular, we show that there are multiple possible initial behaviours of the boundary of the domain. We demonstrate these different behaviours using both the MPCM and FEMPCM and highlight issues stemming from singularities at the boundary which can cause the MPCM and FEMPCM to fail. We also investigate convergence of the method through tracking of the boundary position of the domain and solution comparisons.

Two attempts to alleviate the issues with boundary singularities for the MPCM are proposed. The first of these is a hybrid numerical method making use of both fixed-mesh and moving-mesh methods which allows for modelling of an initially unbounded velocity. The hybrid numerical method is explored in a descriptive sense only without validation.

The second proposal is to introduce an alternative expression for the velocity, bypassing writing the fourth-order problem as a pair of coupled equations. Promising numerical results are given to validate the alternative velocity method.

Finally, in Chapter 7 we summarise the main conclusions of the thesis and outline possible areas of future work.

Chapter 2

Nonlinear Di usion

2.1 Aims of this Chapter

In this chapter we introduce the various nonlinear di usion equations which shall be studied over the course of this thesis, with speci c examples of second, fourth and sixth-order equations considered. We also describe the boundary conditions used in the work, as well as a discussion on alternative boundary conditions that have been used in the literature.

We shall also outline various properties which these equations possess such as scale invariance and conservation of mass. The preservation of these properties is a key aim in the numerical methods to be discussed in this and later chapters.

We then provide a discussion on numerical methods for nding approximate solutions to nonlinear di usion problems and give examples of their use in the literature.

2.2 Nonlinear Diffusion Equations

2.2.1 General Form

The equations considered in this thesis are one-dimensional (1D) nonlinear diffusion PDEs which can be written in the general form: Find $u = u(x,t)$ such that

$$\frac{\partial u}{\partial t} = (-1)^m \frac{\partial}{\partial x}\left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}}\right), \quad \text{in } \Omega_T := \Omega(t) \times (t^0, T), \tag{2.1}$$

where $\Omega(t) \equiv (a(t), b(t)) \subset \mathbb{R}$ is a moving bounded domain, n is a positive constant and the integer $m = 0, 1, 2, \ldots$ determines the order of the equation. We supplement (2.1) with an initial condition $u(x, t^0) = u^0(x)$ and suitable boundary conditions (see §2.2.2–§2.2.4 for a description of the boundary conditions to be used in the work in this thesis).

We can rewrite the general form (2.1) as

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}\left(u^n \frac{\partial q}{\partial x}\right), && \text{in } \Omega_T, \\ q &= (-1)^m \frac{\partial^{2m} u}{\partial x^{2m}}, && \text{in } \Omega_T, \end{aligned} \tag{2.2}$$

where $q(u)$ is dependent on the order of the problem under consideration. Equation (2.2) is the generalised Reynolds equation (Flitton and King (2004)) and the problems considered in this thesis will be derived from (2.1) or (2.2) for second, fourth and sixth-order problems with different $q(u)$ for each case.

The majority of the work covered in this thesis pertains to the fourth-order problem, so this will form the main focus of this chapter.

2.2.2 A Fourth-Order PDE

A fourth-order PDE is obtained by taking $m = 1$ in (2.1) or by setting $q = \frac{\partial^2 u}{\partial x^2}$ in (2.2)

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u^n \frac{\partial q}{\partial x}\right), \quad \text{in } \Omega_T; \tag{2.3a}$$

$$q = \frac{\partial^2 u}{\partial x^2}, \quad \text{in } \Omega_T. \tag{2.3b}$$

the motion of thin films and droplets spreading over a surface (see Greenspan (1978), Bertozzi and Pugh (1996), Bernis and Friedman (1990) and many others). For this reason the fourth-order PDE is often referred to as the thin film equation.

When $n = 1$ the PDE models the flow of fluid between two plates in a Hele-Shaw cell (Beretta et al. (1995)), while for $n = 3$ a droplet with no-slip conditions is modelled (see, e.g., Greenspan (1978)). In this context, the function q

boundary. This work is concentrated on dipole solutions to the problem, where the point at x = 0 is not a moving boundary point. This half-line problem is also considered in Bernis et al. (2000).

It is also possible to consider problems which do not possess a zero contact angle boundary condition. Otto (1998) examines the existence of weak solutions for large-time in the n = 1 case. In this case a xed contact angle of $\frac{}{4}$ is used.

The issue of `dry spots' (alt. `dead cores'), in which the solution develops a region of zero-values inside the moving boundaries has also been studied. King and Taranets (2013) provide existence and uniqueness proofs for travelling wave solutions of this nature, as well as upper and lower bounds on such solutions.

In Giacomelli et al. (2008) the fourth-order problem is transformed from a moving to a xed domain for the n = 1 case. This paper notes the limiting expression for the boundary velocity which is important for the work in this thesis (see equation (6.3)). Existence, uniqueness, smoothness, decay of high derivatives and convergence to

shown that the value of n will determine the nature of the support of the solution. In particular, if n 3=2 then the support of u will not shrink over time, while for values of n 4 the support is constant.

Similar results are shown in King and Bowen (2001), where the initial condition is of the form

$$u^0(x) = I(x) + \quad ;$$

where $I(x) = 0$ for jxj a, $I(x) > 0$ for $jxj < a$ for initial interface position $x =$ a and > 0. The behaviour in the limit ! 0^+ is considered and it is shown that for $n < 3=2$ the interface velocity may be of either sign, while for $n > 3=2$ the velocity is strictly nonnegative. This therefore identi es $n = 3=2$ as a critical case. The support of the solution is shown in this paper to be constant for values of n 3, which identi es the region $n \in (0; 3)$ as being of importance for the moving boundary problem.

tance =

as $x \to b(t)$, where $\dot{b} = \frac{db}{dt}$ and where $B(t)$ needs to be determined as a part of the solution. For the case when $3/2 < n < 3$, the velocity of the boundary is strictly positive, while for $n < 3/2$ the velocity may take either sign.

Blowey et al. (2007) consider the results of King and Bowen (2001) and apply them to initial conditions of the form

$$u^0(x) \quad A_0(b_0 \quad x) \; + C_0(b_0 \quad x) \; ;$$

where , are constants such that > , and explore the small-time behaviour of the right-hand boundary. They demonstrate that, for various choices of n and , the boundary behaves as detailed in King and Bowen (2001) and provide numerical results to support the asymptotic analysis. In particular, they highlight a region in $(n; \;)$ space in which a variety of waiting-time scenarios are exhibited.

2.2.3 A Second-Order PDE

We now describe the second-order problem considered in this thesis. The second-order problem will not be covered in as much detail, but as the rst in the hierarchy of the generalised Reynolds equations (Flitton and King (2004)) it is worth detailing here.

A second-order PDE, obtained by setting $m = 0$ in (2.1) or by taking $q = u$ in (2.2) is

$$\frac{@u}{@t} = \frac{@}{@x} \; u^n \frac{@u}{@x} \; ; \qquad \text{in} \quad _T : \tag{2.7}$$

We shall seek solutions to (2.7) subject to the initial condition

$$u(x; t^0) = u^0(x); \qquad \text{for } x \; 2 \; (\; t^0); \tag{2.8}$$

and subject to suitable boundary conditions imposed at $x = a(t)$ and $x = b(t)$. For the second-order moving boundary problem we require two boundary conditions at each boundary in order for the problem to be well-posed. For the work in this thesis we shall consider boundary conditions

$$u = 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^0; \tag{2.9a}$$

$$uv + u^n \frac{@u}{@x} = 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^0; \tag{2.9b}$$

The rst condition is speci es the position of the moving boundary, while the second condition denosond

(1983), a discussion on waiting time behaviour for a second-order equation is given, with a focus on the initial movement of the boundary once the waiting time is over.

We note that the second-order problem (2.7){(2.9) has one important property missing from the fourth-order problem (2.3){(2.5), in that a maximum principle exists for the second-order problem but not for higher-order problems (as explained in Barrett et al. (1998)). This maximum principle can be used to obtain many of the important results for the second-order problem (2.7){(2.9), particularly uniqueness of solutions.

2.2.4 A Sixth-Order PDE

We now describe the sixth-order problem which will also be considered in this thesis, which is the third in the hierarchy of the generalised Reynolds equations. It shares many properties with the fourth-order problem (2.3){(2.5), and as such we shall consider this problem an extension of the fourth-order work. In particular, we note the lack of a maximum principle for this problem.

The sixth-order PDE can be obtained by setting $m = 2$ in (2.1) or by setting $q = \frac{\partial^2 p}{\partial x^2}$; $p = \frac{\partial^2 u}{\partial x^2}$; in (2.2) and writing it as

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u^n \frac{\partial q}{\partial x} \right); \qquad \text{in } \Omega_T; \tag{2.10a}$$

$$q = \frac{\partial^2 p}{\partial x^2}; \qquad \text{in } \Omega_T; \tag{2.10b}$$

$$p = \frac{\partial^2 u}{\partial x^2}; \qquad \text{in } \Omega_T: \tag{2.10c}$$

We seek solutions to (2.10) subject to the initial condition

$$u(x; t^0) = u^0(x); \qquad \text{for } x \in \Omega(t^0); \tag{2.11}$$

and subject to boundary conditions

$$u = 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^0; \tag{2.12a}$$

$$\frac{\partial u}{\partial x} = 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^0; \tag{2.12b}$$

$$\frac{\partial^2 u}{\partial x^2} = 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^0; \tag{2.12c}$$

$$uv + u^n \frac{\partial q}{\partial x} = 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^0: \tag{2.12d}$$

The boundary conditions (2.12a), (2.12b) and (2.12d) match those used in the fourth-order problem (2.3){(2.5), with boundary condition (2.12c) new, in comparison to the fourth-order problem.

From this point on, we shall refer to (2.10){(2.12) as the sixth-order problem (with (2.10) the sixth-order PDE).

Applications and Results from Literature

The sixth-order PDE with n

equation given in (2.10a). The existence of blow-up similarity solutions occurring in finite time is covered in the first of these papers, while global similarity solutions are explored in the second.

In Smith et al. (1996) the flow under a nitride cap is modelled using the sixth-order equation

$$u_t = \frac{C}{12}\frac{\partial}{\partial x}\left(u^3\frac{\partial^5 u}{\partial x^5}\right),$$

where C is a constant denoting the dimensionless flexural rigidity. This problem is solved subject to the conditions that $u > 1$ inside the moving boundary and $u = 1$ elsewhere, with the moving boundary not known a-priori and therefore found as part of the solution.

Moving Boundaries in the Sixth-Order Problem

It has also been noted that for the sixth-order problem (2.10)–(2.12) a variety of behaviours of the moving boundary can be observed, dependent upon the value of n.

In Flitton and King (2004), a number of initial boundary value problems relating to the sixth-order problem (2.10)–(2.12) are discussed. They note that the support of the initial condition changes only for $n < 5/2$, with the boundary velocity able to take either sign for $n < 5/3$. For values of $n \geq 5/2$ the boundary moves with a zero speed for all problems considered in the paper.

2.3 PDEs in a Moving Framework

The PDEs described in §2.2 are written in a fixed frame of reference with v appearing only in the boundary conditions. It is possible to rewrite the problems in a framework moving with a general velocity $v(x,t)$ by considering some arbitrary sub-region $\hat{\Omega}(t)$ of our domain $\Omega(t)$, which has a boundary $\partial\hat{\Omega}(t)$ moving with this velocity. We shall illustrate this framework for the alternate form (2.2) of the nonlinear PDE (2.1), but the process applies to any of the problems considered in §2.2.

Di erentiating the integral

$$\int_{\zeta(t)} u(x;t)\,dx;$$

with respect to t using Leibniz's Integral Rule with variable limits, we obtain

$$\frac{d}{dt}\int_{\zeta(t)} u(x;t)\,dx = \int_{\zeta(t)} \frac{@u}{@t}dx + \quad uv \Big|_{@\zeta(t)} : \tag{2.13}$$

Then, using the divergence theorem, along with an integral form of the PDE (2.2), we may rewrite (2.13) as

$$\begin{aligned}\frac{d}{dt}\int_{\zeta(t)} u(x;t)\,dx &= \int_{\zeta(t)} \frac{@}{@x}\left(u^n \frac{@q}{@x}\right) dx + \quad uv \Big|_{@\zeta(t)};\\ &= \quad uv + u^n \frac{@q}{@x}\Big|_{@\zeta(t)};\end{aligned} \tag{2.14}$$

which is now written in a framework moving with velocity v

Di erentiation of (2.15) with respect to time shows how the area under the support of $u(x;t)$ changes through time. We use Leibniz's Integral Rule with variable limits again, resulting in

$$\frac{d}{dt}\int_{a(t)}^{b(t)} u(x;t)\,dx = \int_{a(t)}^{b(t)} \frac{@u}{@t}dx + \frac{db}{dt}u(b(t);t) \quad \frac{da}{dt}u(a(t);t): \tag{2.16}$$

Substituting the expression for $\frac{@u}{@t}$ from equation (2.2) into (2.16) and writing

$$\frac{da}{dt} = v(a(t);t); \qquad \frac{db}{dt} = v(b(t);t);$$

gives

$$\begin{aligned}\frac{d}{dt}\int_{a(t)}^{b(t)} u(x;t)\,dx &= \int_{a(t)}^{b(t)} \frac{@}{@x}\left(u^n\frac{@q}{@x}\right)dx + \left[uv\right]_{a(t)}^{b(t)};\\ &= \left[u^n\frac{@q}{@x} + uv\right]_{a(t)}^{b(t)}; &(2.17a)\\ &= 0; &(2.17b)\end{aligned}$$

where the step from (2.17a) to (2.17b) holds due to the boundary condition $uv + u^n\frac{@q}{@x} = 0$ at the boundaries $a(t); b(t)$.

Integrating (2.17b) with respect to t we see that (t) as given by (2.15) is constant with respect to t (and hence we may write $(t) =$ for mass-conserving problems).

There is a link between the steps shown here to demonstrate mass conservation and the steps required to rewrite the problems of x2.2 in a moving framework. Comparing (2.14) to (2.17a) in the case where $\hat{(}t) = (t)$ demonstrates this link for the full domain.

2.4.2 Scale Invariance

An important property of PDEs of the form (2.1) is that they are scale invariant. A PDE is said to be scale invariant if there exists a group transformation of the underlying variables which satisfy the same equations (see, e.g., Budd and Piggott (2001)). In the

where is an arbitrary positive quantity, which leaves the underlying PDE (2.1) unchanged provided that

$$1 = (n + 1) \qquad 2(m + 1) \ : \tag{2.19}$$

An additional condition (such as that arising from the mass conservation property, (2.15)) allows the indices and

2.4.3 Similarity Solutions

For general initial conditions we are unable to find an analytical solution to the problems discussed in §2.2, but there may exist closed-form similarity solutions for certain choices of n and m in (2.1) which we are able to write down explicitly.

Smyth and Hill (1988) note that an explicit closed-form source-type similarity solu-

A Fourth-Order Similarity Solution

In the case of the fourth-order problem (2.3){(2.5) equation (2.25) (with m = 1) leads to

$$\frac{d}{d}(\quad) = \frac{d}{d} \quad {}_{n}\frac{d^3}{d^{\ 3}} \quad ; \tag{2.26}$$

where we have removed the negative sign from each side of (2.25) arising from the choice m = 1.

Closed-form solutions to (2.26) exist for the choice of n = 1. In this case, the ODE integrates simply to give (subject to boundary conditions $= \frac{d}{d} = 0$ at the $= {}_{b}$ and the symmetry condition $\frac{d}{d} = 0$ at $= 0$)

$$= \frac{1}{120} \quad {}_{b}^{2} \quad {}^{2} \quad {}_{+}^{2} : \tag{2.27}$$

In equation (2.27) the subscript + indicates the positive part of (${}_{b}^{2}$ ^{2}).

A Second-Order Similarity Solution

Setting m = 0 in (2.25) produces the ODE

$$\frac{d}{d}\left(\quad \right) = \frac{d}{d} \quad {}^{n}\frac{d}{d} \quad ; \tag{2.29}$$

which can be solved for any n, as noted in Smyth and Hill (1988), subject to the boundary condition $= 0$ at $= {}_{b}$ and the symmetry condition $\frac{d}{d} = 0$ at $= 0$. The solution to this ODE can be written as

$$= \quad \frac{n}{2(n+2)} \quad {}^{1=n} \quad {}^{2}_{b} \quad {}^{2} \quad {}^{1=n}_{+} \; ;$$

and hence (from (2.23)) the similarity solution is given as

$$u^{S}(x;t) = \frac{1}{t} \quad \frac{n}{2(n+2)} \quad {}^{1=n} \quad ! \; {}^{2} \quad \frac{x^{2}}{t^{2}} \quad {}^{1=n}_{+} \; ;$$

where $!$ is related to ${}_{b}$ by the initial condition and $= 1=(n+2)$ from (2.21) (since m = 0). The similarity solution is again symmetric about $= 0$ with moving compact support (in terms of x). An example of a similarity solution for the second-order problem (2.7){(2.9) is given by (Pattle (1959), also Barenblatt (1952))

$$u^{S}(x;t) = \frac{1}{t} \quad \frac{n}{2} \quad {}^{1=n} \quad ! \; {}^{2} \quad \frac{x^{2}}{t^{2}} \quad {}^{1=n}_{+} \; ; \tag{2.30}$$

where $= 1=(n+2)$.

A Sixth-Order Similarity Solution

Setting m = 2 in (2.25) produces the ODE

$$\frac{d}{d}\left(\quad \right) = \frac{d}{d} \quad {}^{n}\frac{d^{5}}{d^{5}} \quad : \tag{2.31}$$

As in the fourth-order case, closed-form solutions exist only in the case where $n = 1$, in which case the ODE (2.31) integrates, subject to the boundary conditions $= \frac{d}{d} = \frac{d^{3}}{d^{3}} = 0$ at $= {}_{b}$ and the symmetry condition $\frac{d}{d} = 0$ at $= 0$, to give

$$= \frac{1}{5040} \quad {}^{2}_{b} \quad {}^{2} \quad {}^{3}_{+} \; ;$$

The corresponding similarity solution to the sixth-order problem (2.10){(2.12) is of the form

$$u^S(x;t) = \frac{1}{5040\alpha t} \left(\omega^2 - \frac{x^2}{t^2} \right)_+^3 :$$

where ω is related to β by the initial condition and where $\alpha = 1/7$ from (2.21) in the case where $n = 1$. This solution is symmetric about $x = 0$ and has moving compact support.

An example of a sixth-order similarity solution (presented in Barrett et al. (2004)) is given by

$$u^S(x;t) = \frac{1}{5040\alpha(t+\beta)^{1/7}} \left(\omega^2 - \frac{x^2}{(t+\beta)^{2/7}} \right)_+^3 ; \tag{2.32}$$

for constants β, ω.

2.5 Numerical Methods for Nonlinear Diffusion

As mentioned in §2.4.3, for most choices of m or n in the PDEs (see (2.1), (2.2)) describing nonlinear diffusion in this thesis there are no closed form solutions available. Indeed, for those cases such that closed form solutions exist they are useful only if the initial condition is of the correct form. This lack of explicit solutions motivates the need for numerical methods for use in obtaining approximate solutions to such nonlinear diffusion equations.

We now present a review of some of the different types of numerical method which feature in the literature, with the main focus on numerical methods for fourth-order problems. There is a large body of literature to draw on, and a wide variety of methods which can be used to produce approximate solutions to the nonlinear PDEs, from which a selection is made here.

We shall mention both moving-mesh methods and fixed-mesh methods in 1D, as well as a brief mention on methods in higher dimensions. A particular class of moving-mesh methods, velocity-based methods based on conservation, is of particular relevance to the work presented in this thesis, and as such is covered separately in Chapter 3.

2.5.1 Moving-Mesh Methods for Nonlinear Diffusion

A moving-mesh method allows the mesh of points to evolve over time, which adds an additional complexity to the solution process but has the benefit of allowing moving features of the solution to be modelled more accurately (Budd et al. (2009)). Such methods are also known as *r*-refinement methods, as opposed to *h*-refinement methods in which mesh points are added or removed from the mesh where required. This latter form

rule and written here in terms of (2.2))

$$\frac{\partial u}{\partial t} - \frac{du}{dx}\frac{dx}{dt} = \frac{\partial}{\partial x}\left(u^n \frac{\partial q}{\partial x}\right); \tag{2.35}$$

where $\frac{du}{dx}$ denotes the rate of change of u in the moving frame.

We refer the reader to Huang et al. (1994), Budd et al. (2009), Huang and Russell (2011) and the references within for a detailed review of the MMPDE method. In particular, Budd et al. (1999) present an application of the MMPDE method to the porous medium equation for generating self-similar numerical solutions.

2.5.2 Fixed-Mesh Methods for Nonlinear Diffusion

A fixed-mesh method differs from a moving-mesh method (such as that discussed in §2.5.1) in that the discretisation of the domain is such that the mesh points are no longer moving over time. This necessitates a distinction between the boundary of the computational domain and the interfaces which define the support of the solution $u(x,t)$, as the interfaces may not lie exactly on the computational boundaries.

For the general PDE given by (2.1), solutions $u(x,t)$ are sought to

$$\frac{\partial u}{\partial t} = (-1)^m \frac{\partial}{\partial x}\left(u^n \frac{\partial^{2m+1} u}{\partial x^{2m+1}}\right); \qquad \text{in} \qquad (t^0, T),$$

where $\Omega := (-B, B)$ is a truncation of the real line such that $x = \pm B$ are fixed points chosen far enough away from the moving interfaces at $x = a(t)$, $x = b(t)$ (see Figure 2.1).

The approximate solution is then sought at each point on a mesh whose points remain fixed over time. Standard numerical methods for boundary value problems (BVPs) can then be used in obtaining the approximation (although there may be issues with representing sharp features of the solution if the density of the mesh points is low in the vicinity of the feature).

Fixed-mesh methods have difficulties modelling the position of a moving boundary, since in general the moving boundary is not located at a mesh point. This generally necessitates the inclusion of some mechanism for determining the position of the interface between two mesh points, such as interpolation or extrapolation. In particular a

Figure 2.1: Image highlighting the di erence between xed and moving domains $_T$. The left-hand image shows a xed-mesh domain, with interfaces given by the black asterisk. The solution outside the interfaces exists but is equal to the value at the interface. The right-hand plot shows a moving domain, where the black asterisk denotes the boundary of the $_T$. The solution is not de ned outside of the boundary.

xed-mesh method may have di culty when approximating singularities in a solution, with the qualitative behaviour of the solution di ering from the true behaviour (see for example Budd et al. (1996)).

An advantage of xed-mesh methods is that there exists much analysis associated with such methods, with known results on error estimates and convergence of numerical solutions. In particular, a large body of literature has arisen looking at constructing

Barrett et al. (1998) present a nite element scheme for solving the fourth-order problem (2.3){(2.5) without the above regularisation. Convergence of numerical solutions to a weak solution of the PDE in 1D is shown. Nonnegativity of the solution is imposed as a constraint, since numerical solutions cannot be guaranteed to satisfy this property otherwise.

Blowey et al. (2007) use the nite element scheme of Barrett et al. (1998) with some modi cations to demonstrate the various small-time behaviours of the free boundary outlined in King and Bowen (2001).

A mixed nite element method is proposed in Burger et al. (2010) for a class of second-order nonlinear PDEs, while Duque et al. (2013) use a nite element method to obtain solutions to a second-order problem with variable exponent n, which di ers from the problems considered in this thesis (which have constant n).

The literature for sixth-order problems is much less abundant than for the second and fourth-order problems, which is likely to be due to the fact that less is known about the behaviour of solutions to such problems. In Smith et al. (1996) a numerical method for solving a sixth-order problem related to ow under a nitride cap is given. The method uses nite di erences and Newton's method is employed at each time step to solve the resulting system. Evans et al. (2000) provide a nite di erence scheme for a sixth-order problem, which is restricted to one space dimension. In both cases, no analysis of the nite di erence schemes is given.

One of the rst attempts of producing a nite element implementation to the sixth-order problem (2.10){(2.12) is given in Barrett et al. (2004). The method is an extension of the method of Barrett et al. (1998) for the fourth-order problem (2.3){(2.5), and the paper notes the lack of research for this order of problem. The method presented is shown to converge for dimensions less than three (although some of the analysis is valid for $d \quad 1$). Flitton and King (2004) provide a brief description of a nite di erence scheme for solving a sixth-order problem, but as an appendix to the analytical results given in the paper.

2.5.3 Moving-to-Fixed Domain Mappings

It is common to create numerical methods for solving nonlinear di usion problems with moving boundaries, such as those considered here, by introducing a mapping from the moving domain to a xed domain. Once this mapping has been constructed, the transformed problem is then solved on the xed domain using standard BVP techniques.

An example of this type of method in the literature arises in the eld of modelling tumour growth. Breward et al. (2002) transform the moving domain $x \in [0; b(t)]$ to a xed domain $\in [0; 1]$ by using a coordinate transform based on a simple scaling and then solve the problem on the xed domain. This method is also described in Lee (2011) (also Lee et al. (2013)), who uses a moving nite di erence method to nd approximate solutions to the tumour growth problem. It is mentioned in Lee (2011) that, although standard techniques can be used in the transformed problem, care must be taken to avoid disrupting the balance between di usion and reaction laws present in the original problem.

Another example of this type of method for a fourth-order problem is given in Giacomelli et al. (2008), where after the moving-to- xed domain mapping has been performed the problem is further formulated by introducing a new variable which splits the operator into linear and nonlinear parts. The authors then prove existence, uniqueness and smoothness of the transformed problem. The problem considered in this paper is the fourth-order PDE (2.3a), subject to zero solution, zero contact angle and

$$V = \lim_{u>0,\, x \to f(u>0)} \frac{\partial^3 u}{\partial x^3}; \quad \text{on } \partial\{u > 0\};$$

at the boundaries, where V denotes the speed at the free boundary. This is a slightly di erent problem to those considered in this thesis in that conservation of mass is not enforced, at the expense of enforcing the limiting process for the boundary velocity. The problem also di ers in that the authors initially assume that $u_0 >$

Figure 2.2 provides an illustrative example of the type of initial condition considered in Giacomelli et al. (2008) for $s_0 = 2$.

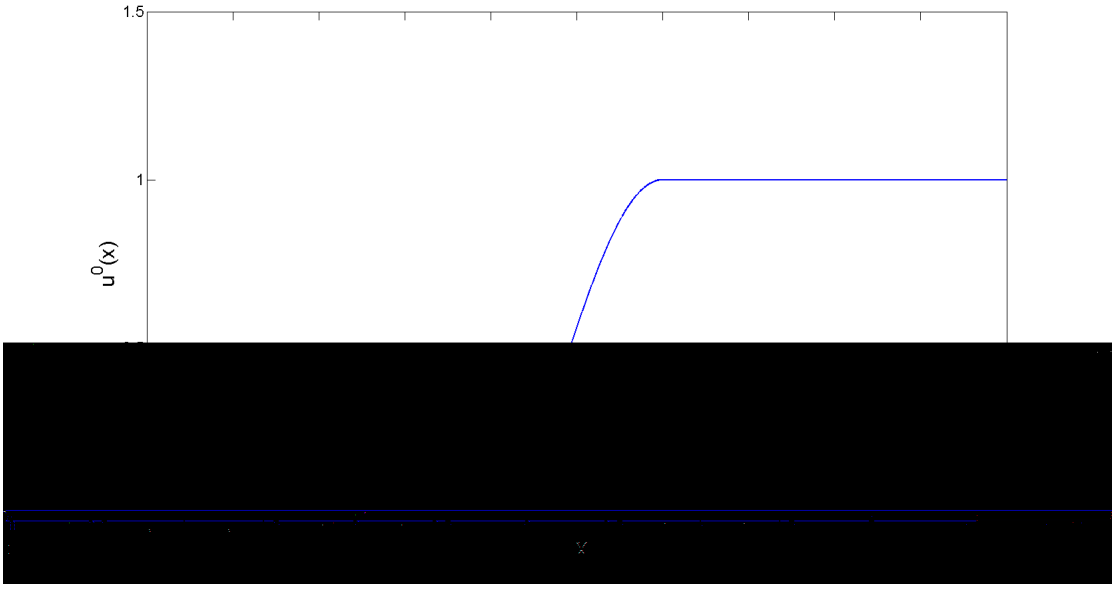

Figure 2.2: Example of the type of initial condition considered in Giacomelli et al. (2008) with $s_0 = 2$.

A further example of this type of method is given in Socolovsky (1988). The issue of how to track moving boundaries using numerical methods is covered, with regard to the porous medium equation and a generalised heat equation. Once transformed onto the xed domain, the problems in Socolovsky (1988) are solved using nite di erence and nite element discretisations, with numerical results presented for the nite di erence schemes for the various problems considered.

2.6 Numerical Methods in Higher Dimensions

All the methods mentioned so far have been in 1D, as these are relevant to the work presented in later chapters of this thesis. There also exist numerical methods in higher dimensions which provide approximate solutions. We now provide a brief insight into some of these methods.

2.6.1 Meshless Methods

The numerical methods described in x2.5.1{x2.5.2 all involve obtaining the solution on a given mesh of points in 1D. If considered in 2D, this mesh of points would include a strong form of connectivity, which links the di erent points together.

An alternative type of numerical method is known as a meshless, or mesh-free, method. In this type of method a number of points are distributed throughout the continuous domain, without concern for the connectivity of the points. The solution is then sought on this cloud of points.

By ignoring connectivity between the points, the issue of mesh tangling is avoided, which can cause problems for moving-mesh methods (see, e.g., Budd et al. (2009)) in that instabilities can arise in the solution which cause an undesired blow-up. Avoiding connectivity concerns also reduces the need for mesh generation procedures, which can be complex and/or computationally expensive (Ma et al. (2008)).

The meshless method has issues with poorly approximating the solution if the points are coarsely distributed through the domain (Belytschko et al. (1996)), or when points are not positioned to accurately capture the dynamics of the solution. This can also occur in xed/moving-mesh methods, particularly if the mesh points are not positioned to capture singularities (Nguyen et al. (2008)), so this issue is not limited to meshless methods.

For a review of meshless methods, we refer to Belytschko et al. (1996), or the more recent review paper of Nguyen et al. (2008). Ma et al. (2008) also present a study of an adaptive algorithm for solving the Euler equation of gas dynamics.

2.6.2 MMPDE Methods in 2D

The MMPDE method described in x2.5.1 can be extended into higher dimensions. A major di erence between 1D and higher dimensional formulations of the MMPDE method is that the mapping obtained through equidistribution is no longer unique (Budd et al. (2009)). In order to obtain a unique mapping additional concerns such as the skewness of the mesh and overlapping must be taken into consideration (Huang and Russell

(1997)).

There are a number of different ways of ensuring equidistribution in higher dimensions, but the method presented in Huang and Russell (1997) and Huang and Russell (1999) makes use of variational procedures and gradient flow equations to obtain the MMPDE required. These variational-based methods are based on minimising a functional $I(\ )$, which in d dimensions has the general form (Budd et al. (2009))

$$I(\ ) = \int_P G(M; ; r_i)dx; \quad i = 1; : : : ; d;$$

for some continuous function G, monitor function M (which could be scalar or matrix-valued) and where r is the gradient operator in terms of physical coordinates x.

The advantage of such formulations is that in a non-convex domain node tangling is less likely than with other formulations. The resulting systems are highly nonlinear however, which is a disadvantage of the method due to the additional computational costs.

We refer the reader to Huang and Russell (2011) and Budd et al. (2009) for more details on this method.

2.6.3 The Parabolic Monge-Ampere Method

If the mapping is represented by a gradient, then equidistribution in 2D leads to a Monge-Ampere equation. This process is related to minimisation of a functional

$$I = \int_C jF(; t) \quad j^2 d ;$$

where F is the mapping between computational and physical space. If the mapping is written as the gradient r of a mesh potential $P(; t)$, the Hessian $H(P)$ of such a potential gives rise to the Monge-Ampere equation

$$M(r \quad P; t)H(P) = \quad (t); \tag{2.36}$$

where (t) is defined as the 2D equivalent of (2.33).

Budd and Williams (2009) provide a description of the Parabolic Monge-Ampère method as a means of generating an equidistributed mesh in higher dimensions, noting that for some applications an equidistributed mesh which is as close as possible to a uniform mesh is desirable. The Parabolic Monge-Ampère solver then arises from a parabolic relaxation of (2.36), involving a time-evolving function

Chapter 3

Velocity-Based Moving Mesh Methods

3.1 Aims of this Chapter

In this chapter we discuss some velocity-based moving-mesh methods, one of which forms the focal point of the numerical methods discussed in this thesis. We also provide some insight into the literature that exists for this type of method.

We shall then present in detail a particular implementation of the approach for use in solving the fourth-order problem (2.3){(2.5). We also describe how this implementation

This is achieved through time stepping the ODE

$$\frac{dX_i}{dt} = V_i;$$

at each node of the mesh.

There are various methods of determining the point velocities in a velocity-based method and we shall highlight some of these in this section. One of these methods (the conservation based method) forms the basis of the Moving Point Conservation Method (MPCM) which is used in this thesis to approximate high order nonlinear di usion.

We now survey some of the key velocity-based methods in the literature.

3.2.1 Moving Finite Element Method

The Moving Finite Element (MFE) method is an early example of a velocity-based method developed by Miller and Miller (1981) and Miller (1981). While a potentially powerful tool for solving nonlinear PDEs it requires considerable care in the implementation in order to avoid inherent singularities.

In the discrete MFE method, the approximate solution $U(x;t)$ is written as a linear combination of time dependent basis functions $w_j(x;t)$ with moving nodes, as

$$U(x;t) = \sum_j U_j(t) w_j(x;t); \tag{3.1}$$

with coe cients $U_j(t)$.

The MFE method is a general method for PDEs of the form $\frac{@u}{@t} = Lu$, where L is a purely spatial operator. The method involves the minimisation over $\frac{dU_i}{dt}$ (the rate of change of U_i in the moving frame) and $\frac{dX_i}{dt}$ of the L^2 norm of the residual of the Pseudo-Lagrangian form of the PDE (see (2.35)) in the form

$$\int_{a(t)}^{b(t)} \left(\sum_j \frac{dU_j}{dt} w_j - \sum_j \frac{dU}{dx}\frac{dX}{dt} w_j - LU \right)^2 dx; \tag{3.2}$$

at each node of the mesh. Carrying out the minimisation over $\frac{dU_i}{dt}$ and $\frac{dX_i}{dt}$ of (3.2) leads to the following pair of equations for each node i:

$$\int_{a(t)}^{b(t)} w_i \left(\frac{dU}{dt} - V\frac{@U}{@x} \right) W\, dx = \int_{a(t)}^{b(t)} w_i LU\, dx; \tag{3.3}$$

$$\int_{a(t)}^{b(t)} \frac{\partial U}{\partial x} w_i \left(\frac{dU}{dt} - V \frac{\partial U}{\partial x} \right) dx = \int_{a(t)}^{b(t)} \frac{\partial U}{\partial x} w_i \, L U \, dx. \qquad (3.4)$$

In later developments a more general weight function was used as standard, with w_i replaced by $W_i = \rho \frac{1}{1+r|U_i|}$ (see Carlson and Miller (1998a,b)).

The solution of (3.3){(3.4) determines the mesh velocities $\frac{dX_i}{dt}$ and solution rates of change $\frac{dU_i}{dt}$ at each node in the mesh. Care must be taken however, in that the set of equations may become singular and as such cannot be solved uniquely (Miller (1981)). The two situations in which singularities occur are when a basis function is redundant in (3.1) or when nodes overtake. The possibility of a singular set of equations appearing leads to the need to introduce penalty functions into the minimisation of (3.2) (Miller and Miller (1981)).

The MFE method has been used successfully in Carlson and Miller (1998a) and Carlson and Miller (1998b), but the need for ne-tuning to avoid singularities is a big disadvantage for the method (Budd et al. (2009)).

3.2.2 The GCL Method

The Geometric Conservation Law (GCL) method (Cao et al. (2002)) arose as a means of deriving an equation for the mesh velocity through di erentiation of the equidistribution equation with respect to time (see x2.5.1 for information on this equation, and also Budd et al. (2009)). A simple explanation of the method in 1D is presented here (taken from Cao et al. (2002)), but the reader is referred to this paper plus that of Budd et al. (2009) and Baines et al. (2011) for more details, in particular the higher dimensional case.

Consider a coordinate transformation $x = x(\ ;t)$ between a computational domain $_C$ and a physical domain $_P(t)$. Let A_C be an arbitrary subset of $_C$ with boundary $@A_C$. Under the coordinate transformation there exists a corresponding subset of the physical domain $A_P(t)$ $_P(t)$.

The integral form of the GCL arises from noting that the change in volume of $A_P(t)$ is equal to the total inward ux over the boundary $@A_P(t)$, i.e.

$$\frac{d}{dt} \int_{A_P(t)} dx = v \qquad ; \qquad (3.5)$$
$$@A_P(t)$$

with v the velocity $\frac{dx}{dt}$ as previously defined.

Under the coordinate transformation we may write

$$\frac{d}{dt}\int_{A_P(t)} dx = \frac{d}{dt}\int_{A_C} J(\xi;t)\,d\xi = \int_{A_C} \frac{DJ}{Dt}\,d\xi; \tag{3.6}$$

where J is the Jacobian of the coordinate transformation and $\frac{DJ}{Dt}$ denotes the total derivative of J.

We may rewrite the right-hand side of (3.5) using the divergence theorem as

$$v_{\partial A_P(t)} = \int_{A_P(t)} \frac{\partial v}{\partial x}dx = \int_{A_C} \frac{\partial v}{\partial x} J(\xi;t)\,d\xi = \int_{A_C} \frac{DJ}{Dt}\,d\xi;$$

from (3.6). Since A_C is arbitrary, this leads to the differential form of the GCL

$$\frac{\partial v}{\partial x} = \frac{1}{J}\frac{DJ}{Dt}. \tag{3.7}$$

This differential form can be used as the basis for moving-mesh methods. The choice

$$J(\xi;t) = \frac{c(\xi)}{M(x;t)}; \tag{3.8}$$

as highlighted in Cao et al. (2002), for some positive monitor function M corresponds to equidistribution, where $c(\xi)$ is a constant independent of time determined from the initial coordinate transform.

Substituting (3.8) into (3.7) we see that

$$\begin{aligned} \frac{\partial v}{\partial x} &= \frac{1}{M}\frac{DM}{Dt}; \\ \Leftrightarrow \quad \frac{\partial M}{\partial t} + v\frac{\partial M}{\partial x} &= M\frac{\partial v}{\partial x}; \\ \Leftrightarrow \quad \frac{\partial (Mv)}{\partial x} + \frac{\partial M}{\partial t} &= 0. \end{aligned} \tag{3.9a}$$

For a given M, equation (3.9a) can be used to define v, with suitable boundary conditions on v. In 1D, this problem can be solved uniquely, but in higher dimensions additional conditions are required in order to solve uniquely for v (Budd et al. (2009)).

As an example, the monitor function $M = u$ can be used to conserve the mass of the solution. Then (3.9a) becomes

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(uv) = 0; \tag{3.10}$$

which is the Eulerian form of mass conservation. We shall use this property in the MPCM described in x3.3. Combined with $\frac{@u}{@t} = L u$

and non mass-conserving problems in terms of nite elements in the moving frame. Finite di erence implementations have also been used in Lee (2011), Blake (2001) and Partridge (2013) for a variety of di erent problems.

In Chapter 2 we demonstrated that the fourth-order ((2.3){(2.5)) , second-order ((2.7){(2.9)) and sixth-order ((2.10){(2.12)) problems were mass-conserving. We now use this property to de ne a consistent local conservation of mass principle for obtaining the velocity at all interior points of the domain (t).

3.2.5 Local Conservation of Mass

We rst de ne a local conservation of mass principle, consistent with conservation of total mass, in which the mass under the curve $u(x;t)$ from $a(t)$ up to an arbitrary point $\hat{x}(t)$ 2 (t) is conserved in time, i.e.

$$\int_{a(t)}^{\hat{x}(t)} u(x;t)\,dx = c(\hat{x}); \tag{3.11}$$

where $c(\hat{x})$ is independent of time. equivalently

$$\frac{d}{dt}\int_{a(t)}^{\hat{x}(t)} u(x;t)\,dx = 0: \tag{3.12}$$

We shall describe the derivation of the velocity in the case of the PDE (2.2), although the process applies for any of the problems found in x2.2 since they are all mass-conserving, being subject to zero- ux boundary conditions. The steps involved in de ning the local conservation of mass principle are similar to those taken in x2.4.1 for demonstrating that the total mass of the PDE is mass conserving.

We begin by di erentiating the left-hand side of (3.12) using Leibniz's Integral Rule, giving

$$\frac{d}{dt}\int_{a(t)}^{\hat{x}(t)} u(x;t)\,dx = \int_{a(t)}^{\hat{x}(t)} \frac{@u}{@t}dx + \frac{d\hat{x}}{dt}u(\hat{x}(t);t) \quad \frac{da}{dt}u(a(t);t):$$

From (3.12) and (2.2),

$$\int_{a(t)}^{\hat{x}(t)} \frac{@}{@x}\left(u^n\frac{@q}{@x}\right)dx + \frac{d\hat{x}}{dt}u(\hat{x}(t);t) \quad \frac{da}{dt}u(a(t);t) = 0;$$

Integrating and writing

$$\frac{da}{dt} = v(a(t); t); \qquad \frac{dx}{}$$

in the next section, but shall provide details on how the implementation can be modified for second and sixth-order problems in §3.4 and §3.5.

3.3 A Finite Difference Implementation of the MPCM for the Fourth-Order Problem

We now calculate approximate solutions to the fourth-order problem (2.3)–(2.5) using a finite difference implementation of the MPCM. In the case where a self-similar solution exists (i.e. for $n = 1$) $u(x;t)$ is a quartic function (Smyth and Hill (1988)), with $q(x;t)$ therefore a quadratic function from (2.3b). From (3.14), the velocity $v(x;t)$ is therefore a linear function. We shall use this information in the implementation of the finite difference method, although the method itself applies to non self-similar cases.

3.3.1 Discretisations

At each time t^k, the interval $(a(t^k); b(t^k))$ is discretised by an ordered mesh of N nodes $a(t^k) = x_1(t^k) < x_2(t^k) < \cdots < x_N(t^k) = b(t^k)$.

We shall denote by F_i^k the approximation of the function $f(x;t)$ at time $t = t^k$, node X_i^k, i.e. $F_i^k \approx f(X_i^k; t^k)$. We shall seek approximations

$$U^k := \{U_i^k\}, \qquad \text{for } i = 2, \ldots, N-1; \tag{3.16a}$$

$$Q^k := \{Q_i^k\}, \qquad \text{for } i = 1, \ldots, N; \tag{3.16b}$$

$$V^k := \{V_i^k\}, \qquad \text{for } i = 1, \ldots, N; \tag{3.16c}$$

to $u(x;t)$, $q(x;t)$ and $v(x;t)$ on the moving mesh, with $U_1^k = U_N^k = 0$ for all k from (2.5a).

3.3.2 Initial Conditions

On the initial mesh X^0 we sample the initial condition $u^0(x)$ at each node to get an initial approximation U^0 by setting

$$U_i^0 = u^0(X_i^0);$$

for i = 2;::::;N

The approach used here is that of extrapolation using a second-order Lagrange polynomial. Since $q(x;t)$ is a quadratic in the self-similar case this is the appropriate degree of polynomial required here.

Performing an extrapolation using second-order Lagrange polynomials we arrive at an equation for determining the value Q_1 at a given time step:

$$L_1(X_4)Q_1 + L_2(X_4)Q_2 + L_3(X_4)Q_3 \quad Q_4 = 0; \tag{3.18}$$

where the $L_1;\ldots;L_4$ are the second-order Lagrange polynomials.

Similarly, an equation for determining the value of Q_N at a given time step is

$$Q_{N\ 3} + L_{N\ 2}(X_{N\ 3})Q_{N\ 2} + L_{N\ 1}(X_{N\ 3})Q_{N\ 1} + L_N(X_{N\ 3})Q_N = 0: \tag{3.19}$$

The Matrix System

To obtain the values of Q at a given time step at each of the nodal positions we solve the matrix system

$$SQ = TU; \tag{3.20}$$

where $Q = fQ_ig$ $(i = 1;\ldots;N)$, $U = fU_ig$ $(i = 1;\ldots;N)$ and S, T are matrices of size $N \quad N$. Rows $2;\ldots;N \quad 1$ of these matrices come from equation (3.17), while the first and final rows are given by equations (3.18) and (3.19) respectively. The vector U includes the boundary values $U_1 = 0$ and $U_N = 0$ as well as the interior values.

By combining these equations, we see that the matrix S is of the form

$$S = \begin{pmatrix} L_1(X_4) & L_2(X_4) & L_3(X_4) & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & & & \vdots \\ & 0 & 1 & 0 & & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & 0 & 1 & 0 & \\ \vdots & & & & 0 & 1 & 0 \\ 0 & \cdots & 0 & 1 & L_{N\ 2}(X_{N\ 3}) & L_{N\ 1}(X_{N\ 3}) & L_N(X_{N\ 3}) \end{pmatrix}: \tag{3.21}$$

The matrix T is of the form

$$T = \begin{pmatrix} 0 & & & & & & 1 \\ 0 & \cdots & & & & \cdots & 0 \\ T_2 & T_2 & T_2^+ & 0 & & & \vdots \\ 0 & T_3 & T_3 & T_3^+ & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & T_{N-2} & T_{N-2} & T_{N-2}^+ & 0 \\ \vdots & & & 0 & T_{N-1} & T_{N-1} & T_{N-1}^+ \\ 0 & \cdots & & & & \cdots & 0 \end{pmatrix}; \qquad (3.22)$$

where for $i = 2; \ldots; N-1$

$$T_i = \frac{1}{\frac{1}{2}C - T_{1C}^{-1}}$$

Given U_i^k and Q_i^k, both for $i = 1;::::;N$, (denoted by U_i and Q_i respectively for ease of notation) we compute V_i^k, $i = 1;::::;N$, (denoted V_i for ease of notation) as follows.

The gradient q^0 of $q(x;t)$ in $(X_{i\ 1};X_i)$ is approximated by

$$Q^0_{i\ 1=2} = \frac{Q_i \quad Q_{i\ 1}}{X_i \quad X_{i\ 1}};$$

say, and is of first order accuracy. A second order approximation can be obtained by a linear interpolation of these gradients through the points $(X_{i\ 1=2};Q^0_{i\ 1=2})$ and $(X_{i+1=2};Q^0_{i+1=2})$ such that

$$V_i = \quad (U_i^{n\ 1}) \begin{pmatrix} 0 \\ B \\ B \\ @ \end{pmatrix} \frac{(Q_{i+1} \quad Q_i)}{(X_{i+1} \quad X_i)^2} + \frac{(Q_i \quad Q_{i\ 1})}{(X_{(}U^{n\ 1}}$$

We approximate the ODEs (3.25) by the explicit Forward Euler time stepping method

$$X_i^{k+1} = X_i^k + \Delta t V_i^k; \tag{3.26}$$

where Δt is the size of the time step used, for $i = 1; \ldots; N$. The scheme (3.26) is not scale invariant with a fixed Δt, which we shall discuss further in Chapter 4.

The size of Δt in (3.26) is of great importance to the successful updating of mesh positions from one time step to the next. In a moving-mesh method the value of Δt must be chosen in order to avoid node tangling occurring (see §3.3.7 for a description of node tangling and the consequences of such an occurrence). The size of Δt also influences the accuracy of the resulting mesh positions. The local truncation error of (3.26), which we denote by τ_i, is

$$\tau_i = \frac{\Delta t}{2} \frac{d^2x}{dx^2}\Big|_t;$$

for some $t \in (t; t + \Delta t)$. This truncation error indicates that a smaller value of Δt will lead to a more accurate resulting mesh position at the updated time step.

3.3.6 Updating the Solution Values

We now recover the solution on the updated mesh X^{k+1} using the mass constants of (3.11). The discretised mass can be defined using a discrete form of the total mass of the solution (2.15). We discretise (2.15) using the composite trapezoidal rule

$$\frac{1}{2} \sum_{j=2}^{N} (X_j^{k+1} - X_{j-1}^{k+1})(U_j^{k+1} - U_{j-1}^{k+1}); \tag{3.27}$$

which is constant for all k due to mass conservation.

Expanding and rearranging the terms in (3.27) we see that, using the boundary conditions $U_1^{k+1} = U_N^{k+1} = 0$,

$$\frac{1}{2} \sum_{j=2}^{N} (X_j^{k+1} - X_{j-1}^{k+1})(U_j^{k+1} - U_{j-1}^{k+1}) \equiv \frac{1}{2} \sum_{j=2}^{N} (X_{j+1}^{k+1} - X_{j-1}^{k+1}) U_j^{k+1}.$$

Denoting by μ_i the mass constants

$$\mu_i := \frac{1}{2}(X_{i+1}^{k+1} - X_{i-1}^{k+1}).$$

centred at node i of the mesh, we see that $\sum_i \alpha_i = \quad$.

As the total mass is conserved, it is consistent to make the α_i constant for all time and this is the approximation to (3.11) that we use. Due to conservation of mass, we can use the initial values X_i^0 and U_i^0 at time $t = t^0$ to calculate the values of the mass constants

$$\alpha_i = (X_{i+1}^0 \quad X_{i-1}^0)U_i^0; \qquad \text{for } i = 2;\ldots;N \quad 1: \tag{3.29}$$

We can use the α_i to recover the solution at any given time. From (3.28) and (3.29), we use the new nodal positions to update the solution values for our approximation U^{k+1} to $u(x;t$

One of the consequences of performing solution recovery in the manner described in §3.3.6 is that a nonnegative initial condition will remain nonnegative for all later times (provided monotonicity of the mesh is upheld). This consequence is desirable since it mimics the properties of the underlying problem (see §2.2.2 and Bernis and Friedman (1990)). Node tangling is therefore extremely important to avoid when choosing the size of Δt in (3.26).

We next detail how the implementation of the MPCM described above can be modified for use in the second and sixth-order problems detailed in §2.2.

3.4 Implementation of the MPCM for Second-Order Problems

We now outline the steps required for implementing the MPCM for the second-order problem given by (2.7)–(2.9).

The implementation of the MPCM described in §3.3 requires some modifications in order to be valid for the second-order problem (2.7)–(2.9), with the most important change being that $q(x,t) = u(x,t)$, so there is no need to solve for $q(x,t)$ in the second-order problem (2.7)–(2.9). We shall seek approximations U^k and V^k from (3.16a) and (3.16c) respectively.

The work described in this Section is an extension of that found in Parker (2010), which considers the second-order problem (2.7)–(2.9) for the specific value of $n = 3$.

3.4.1 Approximating $v(x,t)$

Obtaining the velocity $v(x,t)$ at each node in the mesh is also slightly different in the second-order problem (2.7)–(2.9), since there exists an expression for the velocity for any value of n (Smyth and Hill (1988)).

From (3.14) and noting that $q = u$, the velocity (at all points where $u > 0$) is given by

$$v = -u^{n-1}\frac{\partial u}{\partial \dot{x}}$$

which we may rewrite as

$$v = \frac{1}{n}\frac{\partial(u^n)}{\partial x}. \tag{3.31}$$

Given U_i^k, $i = 1;\ldots;N$, (denoted U_i for ease of notation) we compute V_i^k, $i = 1;\ldots;N$, (denoted V_i for ease of notation) as follows. The velocity is approximated from (3.31) at interior points by (3.24) with Q_i replaced by U_i^n

$$V_i = \frac{1}{n}\left(\frac{\dfrac{U_{i+1}^n - U_i^n}{(X_{i+1} - X_i)^2} + \dfrac{U_i^n - U_{i-1}^n}{(X_i - X_{i-1})^2}}{\dfrac{1}{X_{i+1} - X_i} + \dfrac{1}{X_i - X_{i-1}}}\right); \tag{3.32}$$

which is a linear interpolation through the points $(X_{i-1=2};(U_{i-1=2}^0)^n)$ and $(X_{i+1=2};(U_{i+1=2}^0)^n)$ and is of second order on the irregular grid.

The approximation (3.32) is used for interior points of the mesh, with v at the boundary approximated using a quadratic extrapolation as described in §3.3.4.

3.4.2 Mesh Movement and Solution Recovery

Updating the mesh and solution recovery is performed exactly as described in §3.3.5 and §3.3.6.

3.5 Implementation of the MPCM for the Sixth-Order Problem

The implementation of the MPCM for the sixth-order problem (2.10)–(2.12) is similar to that for the fourth-order implementation given in §3.3, with the added requirement of an approximation P^k to the p

3.5.1 Approximating $p(x;t)$

We seek an approximation to $p(x;t)$ at each node in the mesh. Given U_i^k, $i = 1;:::;N$, (denoted by U_i for ease of notation) we compute P_i^k, $i = 1;:::;N$, (denoted P_i for ease of notation) as follows.

On a variable mesh, a three-point central di erence approximation P_i is

$$P_i = \begin{matrix} 0 \\ @\frac{\frac{U_{i+1} \quad U_i}{X_{i+1} \quad X_i}}{\frac{1}{2}(X_{i+1} \quad X_{i\ 1})} \end{matrix} \begin{matrix} \frac{U_i \quad U_{i\ 1}}{X_i \quad X_{i\ 1}} \\ A \end{matrix} \begin{matrix} 1 \\ ; \end{matrix} \tag{3.34}$$

which has a local truncation error, $_i$, that can be shown to be equal to

$$_i = \frac{(X_{i+1} \quad 2X_i + X_{i\ 1})}{3} p_i^0 \quad \frac{1}{12} \frac{(X_{i+1} \quad X_i)^3 + (X_i \quad X_{i\ 1})^3}{X_{i+1} \quad X_{i\ 1}} p_i^{00};$$

for some $2 (X_{i\ 1}; X_{i+1})$.

At the boundary points, we use second-degree Lagrange polynomials (as described in x3.3.3), giving the following equations for obtaining P_1 and P_N:

$$L_1(X_4)P_1 + L_2(X_4)P_2 + L_3(X_4)P_3 \quad P_4 = 0; \tag{3.35}$$

and

$$P_{N\ 3} + L_{N\ 2}(X_{N\ 3})P_{N\ 2} + L_{N\ 1}(X_{N\ 3})P_{N\ 1} + L_N(X_{N\ 3})P_N = 0: \tag{3.36}$$

To obtain the values of P (from (3.33)) at a given time step we solve the matrix system

$$SP = TU;$$

arising from (3.34), (3.35) and (3.36), where $P = fP_ig$ $(i = 1;:::;N)$, $U = fU_ig$ $(i = 1;:::;N$, including boundary values $U_1 = U_N = 0)$ and where the matrices S and T are as given in (3.21) and (3.22) respectively.

3.5.2 Approximating $q(x;t)$

Given the P_i, $i = 1;:::;N$, obtained from x3.5.1, we now seek Q_i^k, $i = 1;:::;N$, (denoted by Q_i for ease of notation) approximating $q(x;t)$ from (2.10b) on each node of the mesh.

On a variable mesh, a three-point central di erence approximation Q_i at node i is

$$Q_i = \frac{\frac{P_{i+1} - P_i}{X_{i+1} - X_i} - \frac{P_i - P_{i-1}}{X_i - X_{i-1}}}{\frac{1}{2}(X_{i+1} - X_{i-1})} : \tag{3.37}$$

The local truncation error, $_i$, of (3.37) can be shown to be equal to

$$_i = \frac{(X_{i+1} - 2X_i + X_{i-1})}{3} q_i^0 - \frac{1}{12} \frac{(X_{i+1} - X_i)^3 + (X_i - X_{i-1})^3}{X_{i+1} - X_{i-1}} q_i^{00} ;$$

for some $2 (X_{i-1}; X_{i+1})$.

3.6 The S Property

In the next chapter, we shall show that by modifying some of the steps in the MPCM (i.e. by modifying the appropriate schemes such that they are of the required order to match the exact solution for the relevant problem) and by use of a scale-invariant time stepping scheme, it is possible to propagate initial conditions coinciding with a similarity solution forward in time (over a single time step) to within rounding error. We shall define this property as the S Property:

De nition: The S Property

A numerical method is said to possess the S Property in some norm if, given a self-similar scaling solution to a time dependent problem, the norm of the error in the numerical solution after a single time step is zero. In other words the exact solution is preserved in that norm.

3.7 Summary of this Chapter

In this chapter we have provided a brief review of velocity-based moving-mesh methods, including some of the more common approaches seen in the literature. We then introduced the MPCM, which is a nite di erence implementation of a conservation-based method. The method has been discussed in terms of the fourth-order problem detailed in x2.2, with further details on how the method is adapted for second and sixth-order problems provided.

In the next chapter we shall provide a discussion on how the MPCM as described in this chapter can be modi ed such that it then possesses the S Property. We shall detail the required modi cations to the basic MPCM presented in this chapter as well as the other requirements for the MPCM (such as initial condition and time stepping method).

Chapter 4

An Implementation of the MPCM Possessing the S Property

4.1 Aims of this Chapter

In this chapter we shall demonstrate the ability of the MPCM described in Chapter 3 (with some modifications) to possess the S Property (described in §3.6) in the l^1 norm. This ability shall be proven for the fourth-order problem (2.3)–(2.5) (with $n = 1$), with numerical results shown to validate the claim of the method possessing the S Property.

We shall also outline and provide numerical results which demonstrate the S Property for implementations of the MPCM in the second-order problem (2.7)–(2.9) (for any n) and the sixth-order problem (2.10)–(2.12) with $n = 1$.

4.2 The Fourth-Order Problem with $n = 1$

In this chapter we shall mainly focus on the fourth-order problem (2.3)–(2.5) with the specific choice of $n = 1$. We recall that for this choice of n, closed-form similarity solutions of the fourth-order PDE (2.3) exist. We shall demonstrate that if such a similarity solution is used to provide the initial approximation U^0 at the nodes of the initial mesh, the MPCM given in §3.3 can be modified such that it possesses the S

Property (see x3.6) in the l^1 norm.

Throughout this chapter we shall make use of the scaled time variable s rather than t, since a scale-invariant time stepping scheme will be used (see x4.3.3 for details on the scale-invariant time stepping scheme used).

We restate the problem being considered for clarity. We seek a solution $u(x;s)$ to the 1D fourth-order nonlinear di usion equation with $n = 1$,

$$\frac{@u}{@s} = \frac{@}{@x}\left(u\frac{@q}{@x}\right); \tag{4.1a}$$

$$q(x;s) = \frac{@^2u}{@x^2}; \tag{4.1b}$$

for $x \in S$, subject to the initial condition at time $s = s^0$ given by

$$u(x;s^0) = u^0(x); \qquad \text{for } x \in (s^0);$$

We are given an initial mesh X^0 discretisating the domain (s^0) and an approximation U^0 to the initial condition $u^0(x)$ such that $u^0(x) = u^S(x; s^0)$ at each node in the mesh. We therefore have that

$$U_i^0 = u^S(X_i^0; s^0); \qquad \text{for } i = 1; : : : ; N:$$

4.3.1 Approximating $q(x; s)$

As noted in x3.3.3, we first seek an approximation Q^k at the nodes. The scheme given in x3.3.3 has a local truncation error proportional to $\frac{@q}{@x}$ and as such is not zero for quadratic $q(x; s)$, quartic $u(x; s)$. We therefore seek to modify the finite difference scheme used so that it is exact for quadratic $q(x; s)$, quartic $u(x; s)$. Given U_i^k, $i = 1; : : : ; N$, (denoted by U_i for ease of notation) we compute Q_i^k, $i = 1; : : : ; N$, (denoted by Q_i for ease of notation) as follows.

We obtain Q at interior points of the mesh using a modified three-point central-difference approximation of (4.1b) of the form

$$Q_i \quad \frac{\frac{U_{i+1} \quad U_i}{X_{i+1} \quad X_i} \quad \frac{U_i \quad U_{i \quad 1}}{X_i \quad X_{i \quad 1}}}{\frac{1}{2}(X_{i+1} \quad X_{i \quad 1})} + C_1 \frac{Q_{i+1} \quad Q_{i \quad 1}}{X_{i+1} \quad X_{i \quad 1}} + C_2 \frac{\frac{Q_{i+1} \quad Q_i}{X_{i+1} \quad X_i} \quad \frac{Q_i \quad Q_{i \quad 1}}{X_i \quad X_{i \quad 1}}}{\frac{1}{2}(X_{i+1} \quad X_{i \quad 1})}; \tag{4.3}$$

where we choose C_1 and C_2 such that terms relating to the first and second derivatives of q in the local truncation error will vanish. The method will then be exact for quartic $u(x; s)$, quadratic $q(x; s)$.

The local truncation error, $_i$, of the modified approximation (4.3) can be shown to be equal to

$$\begin{aligned} {}_i = {} & q_i + u_i^{00} \quad \frac{1}{3}(X_{i+1} \quad 2X_i \quad X_{i \quad 1}) + C_1 \quad q_i^0 \\ & \frac{1}{12} \quad \frac{(X_{i+1} \quad X_i)^3 + (X_i \quad X_{i \quad 1})^3}{X_{i+1} \quad X_{i \quad 1}} \quad + \frac{1}{2}(X_{i+1} \quad 2X_i \quad X_{i \quad 1})C_1 + C_2 \quad q_i^{00} \; ; \end{aligned} \tag{4.4}$$

for some $2 (X_{i \quad 1}; X_{i+1})$, where $q_i = q(X_i)$, $q_i^0 = \frac{@q}{@x}_{X_i}$, $q_i^{00} = \frac{@^2 q}{@x^2}_{X_i}$.

From equation (4.4) we can see that using $q = u^{00}$ at X_i and choosing

$$C_1 = \frac{(X_{i+1} \quad 2X_i \quad X_{i \quad 1})}{3}; \tag{4.5}$$

removes all derivatives up to q_i^0 from the local truncation error, which then becomes equal to

$$_i = \frac{C_1}{2}(X_{i+1} \quad 2X_i \quad X_{i\ 1}) + C_2 + \frac{1}{12}\frac{(X_{i+1} \quad X_i)^3 + (X_i \quad X_{i\ 1})^3}{X_{i+1} \quad X_{i\ 1}} \quad q_i^{00} ;$$

for some $2 (X_{i\ 1}; X_{i+1})$.

Therefore, choosing

$$\begin{aligned} C_2 &= \frac{C_1}{2}(X_{i+1} \quad 2X_i \quad X_{i\ 1}) \quad \frac{1}{12}\frac{(X_{i+1} \quad X_i)^3 + (X_i \quad X_{i\ 1})^3}{X_{i+1} \quad X_{i\ 1}}; \\ &= \frac{(X_{i+1} \quad 2X_i \quad X_{i\ 1})^2}{6} \quad \frac{1}{12}\frac{(X_{i+1} \quad X_i)^3 + (X_i \quad X_{i\ 1})^3}{X_{i+1} \quad X_{i\ 1}}; \\ &= \frac{(X_{i+1} \quad X_i)^2 \quad 3(X_{i+1} \quad X_i)(X_i \quad X_{i\ 1}) + (X_i \quad X_{i\ 1})^2}{12}; \end{aligned} \tag{4.6}$$

removes all q_i^{00} terms from the local truncation error, which then vanishes for quadratic $q(x; s)$.

We can therefore use equation (4.3) to construct a system of equations, with the values of C_1 and C_2 from equations (4.5) and (4.6) respectively. Once solved, the system will produce an approximation Q at each interior point of the mesh having zero local lo

By combining these equations, we see that the matrix S is of the form

$$S = \begin{pmatrix} L_1(X_4) & L_2(X_4) & L_3(X_4) & 1 & 0 & \cdots & 0 \\ S_2 & S_2 & S_2^+ & 0 & & & \vdots \\ 0 & S_3 & S_3 & S_3^+ & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & S_{N-2} & S_{N-2} & S_{N-2}^+ & 0 \\ \vdots & & & 0 & S_{N-1} & S_{N-1} & S_{N-1}^+ \\ 0 & \cdots & 0 & 1 & L_{N-2}(X_{N-3}) & L_{N-1}(X_{N-3}) & L_N(X_{N-3}) \end{pmatrix}; \tag{4.7}$$

where for $i = 2, \ldots, N-1$

$$S_i = \begin{pmatrix} \dfrac{C_1}{X_{i+1} - X_{i-1}} & \dfrac{C_2}{\frac{1}{2}(X_i - X_{i-1})(X_{i+1} - X_{i-1})} \end{pmatrix};$$

$$S_i = \begin{pmatrix} 1 + \dfrac{C_2}{\frac{1}{2}(X_{i+1} - X_{i-1})} & \dfrac{1}{X_{i+1} - X_i} + \dfrac{1}{X_i - X_{i-1}} \end{pmatrix};$$

$$S_i^+ = \begin{pmatrix} \dfrac{C_1}{X_{i+1} - X_{i-1}} & \dfrac{C_2}{\frac{1}{2}(X_{i+1} - X_i)(X_{i+1} - X_{i-1})} \end{pmatrix}:$$

The matrix T is of the form

$$T = \begin{pmatrix} 0 & \cdots & & & & \cdots & 0 \\ T_2 & T_2 & T_2^+ & 0 & & & \vdots \\ 0 & T_3 & T_3 & T_3^+ & 0 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & T_{N-2} & T_{N-2} & T_{N-2}^+ & 0 \\ \vdots & & & 0 & T_{N-1} & T_{N-1} & T_{N-1}^+ \\ 0 & \cdots & & & & \cdots & 0 \end{pmatrix}; \tag{4.9}$$

where for $i = 2, \ldots, N-1$

$$T_i = \frac{1}{\frac{1}{2}(X_i - X_{i-1})(X_{i+1} - X_{i-1})};$$

$$T_i = \begin{pmatrix} \dfrac{1}{\frac{1}{2}(X_{i+1} - X_{i-1})} & \dfrac{1}{X_{i+1} - X_i} + \dfrac{1}{X_i - X_{i-1}} \end{pmatrix};$$

$$T_i^+ = \frac{1}{\frac{1}{2}(X_{i+1} - X_i)(X_{i+1} - X_{i-1})}:$$

The matrix S is pentadiagonal due to the first and last rows, with the remaining rows tridiagonal. Through the application of two Gaussian elimination steps the whole system becomes tridiagonal.

4.3.2 The Nodal Velocity

The nodal velocities V are found using the steps detailed in §3.3.4 with no modifications.

4.3.3 Updating the Nodal Positions

We noted in x3.3.5 that the explicit Forward Euler time stepping scheme was not scale-invariant. It is possible to make the time stepping scheme scale invariant however, through the introduction of a scaled time step with a new time variable s, where $s = t$ (Budd and Piggott (2001), Budd et al. (2001), Baines et al. (2006)). Writing (3.15) in terms of the known function $v(x;t)$,

$$\frac{dx}{ds} = \frac{dx}{dt}\frac{dt}{ds} = \frac{dx}{dt}\frac{1}{t^{\ 1}} = \frac{1}{}s^{1=\ 1}v(x;t); \tag{4.12}$$

the explicit Euler time stepping scheme for (4.12),

$$X_i^{k+1} = X_i^k + \frac{1}{}(s^k)^{1=\ 1}\ sV_i^k; \tag{4.13}$$

is then scale invariant with constant s. We shall refer to (4.13) throughout this thesis as a scale invariant time stepping scheme.

For the fourth-order problem (2.3){(2.5) with $n = 1$ we have $= 1=5$ from (2.21)

from which we have, at any xed time,

$$\frac{d^2x}{ds^2} = 0: \tag{4.16}$$

From (4.14) and (4.16), we see that in the case where a scale invariant time stepping scheme is used, the LTE relating to the time discretisation of the mesh movement is equal to zero. This implies that the mesh points are updated to within rounding error

We write (4.17) in terms of and as

$$
\begin{aligned}
U_i^{k+1} \quad u^S(X_i^{k+1}; s^{k+1}) &= \frac{(\ {}^0_{i+1} \quad {}^0_{i\ 1})}{(\ {}^{k+1}_{i+1} \quad {}^{k+1}_{i\ 1})} \frac{{}^0_i}{s^{k+1}} \quad \frac{{}^{k+1}_i}{s^{k+1}}; \\
&= \frac{(\ {}^0_{i+1} \quad {}^0_{i\ 1})}{(\ {}^{k+1}_{i+1} \quad {}^{k+1}_{i\ 1})} \frac{C}{s^{k+1}} \ D^2 \quad ({}^0_i)^2 \ {}^{2}_{+} \quad \frac{C}{s^{k+1}} \ D^2 \quad ({}^{k+1}_i)^2 \ {}^{2}_{+}; \\
&= \frac{C}{s^{k+1}} \quad \frac{(\ {}^0_{i+1} \quad {}^0_{i\ 1})}{(\ {}^{k+1}_{i+1} \quad {}^{k+1}_{i\ 1})} \ D^2 \quad ({}^0_i)^2 \ {}^{2}_{+} \quad D^2 \quad ({}^{k+1}_i)^2 \ {}^{2}_{+} \ :
\end{aligned}
\tag{4.19}
$$

Since and are invariant, they are independent of time and therefore independent of k. From this invariance, ${}^0_i = {}^{k+1}_i$, for $i = 2; : : : ; N \quad 1$, provided that the mesh points (and hence $_i$ values) have been moved exactly. The right hand side of (4.19) is equal to zero and hence the error at node i is equal to zero. The solution is therefore recovered exactly on each node of the mesh in the case where the error in the mesh positions is equal to zero.

4.3.5 Implications

We have shown that in the case where the initial condition coincides with a similarity solution at the nodes of an initial mesh, the modi ed MPCM implemented in this section will approximate q(x; s) and v(x; s) with zero local truncation error. These approximations are then equal to the exact values to within rounding error at the nodes over a single time step. The local truncation error incurred by the method in moving mesh points is also equal to zero, and hence mesh points are moved to within rounding error, if a scale-invariant time stepping scheme is used. The approximation to u(x; s) is then also recovered exactly since the mesh points are moved exactly.

We therefore argue that the modi ed MPCM described here possesses the S Property in the l^1 norm (although in practice rounding errors will be present in all approximations).

4.3.6 Numerical Results

We now present numerical results for the implementation of the MPCM given in §4.3. We shall demonstrate that over a single time step the method is able to propagate a similarity solution to within rounding error in the l^1 norm. We then present results over a larger time window consisting of multiple time steps in order to illustrate how the MPCM performs over multiple steps.

Our choice of the l^1 norm for the numerical results in this section stems from the point-wise nature of the finite difference method. If our claim that the MPCM possesses the S Property

time $s^0 = 1$, such that

$$U^0$$

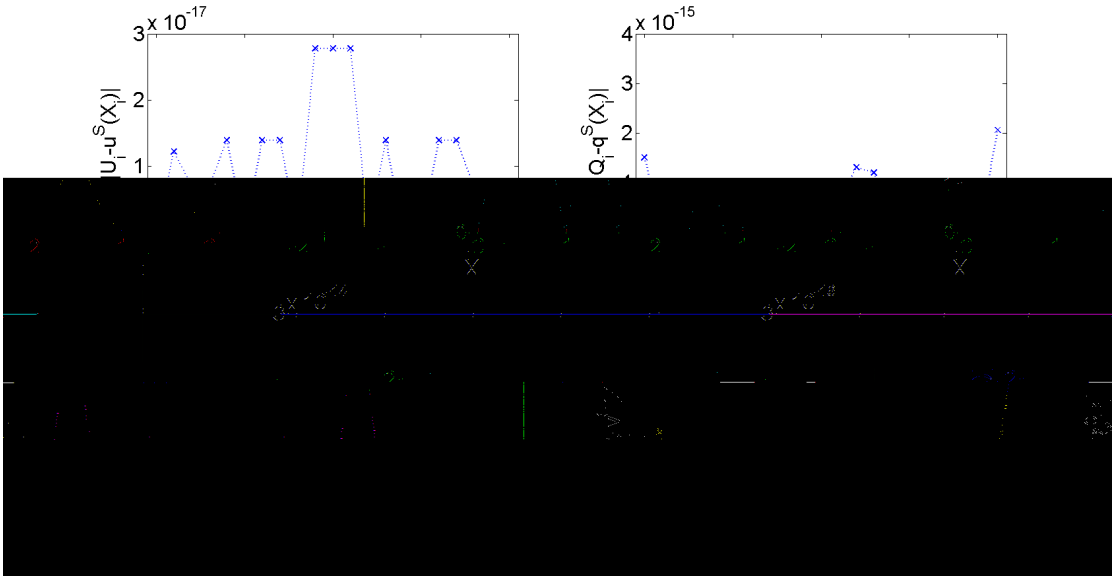

Figure 4.1: Absolute error in the MPCM over a single time step. The top left window contains the absolute error in U^1 ($O(10^{-17})$), the top right window the error in Q^0 ($O(10^{-15})$), bottom left the error in V^0 ($O(10^{-14})$) and bottom right the error in the mesh positions ($O(10^{-16})$).

error in the approximations and the boundary nodes.

At the end of each time step, we determine the accuracy of the method by calculating the l^1 norm of the error, given by (in the case of the error in the solution U^k)

$$E_N^U = \max_{i=1,\ldots,N} |U_i^k - u^S(X_i^k, s^k)| \qquad \text{for } k = 1, \ldots, K,$$

where K is the total number of time steps performed. Similar quantities for the error in Q^k (denoted by E_N^Q) and V^k (denoted by E_N^V) are calculated, using (4.21) for the exact values of q^S and v^S.

The error in the boundary node positions is denoted by E_N^B (for the right-hand boundary node) and is given by

$$E_N^B = \frac{|X_N^k - b(s^k)|}{|b(s^k)|},$$

where $b(s^k)$ denotes the exact position of the right-hand boundary at time s^k and from

We see that the error increases at the start of the time window, reaching a peak at time s 1:25 which is approximately 3:25 10 ^{14}. The error then remains of this magnitude throughout the remainder of the time window, with slight uctuations observed.

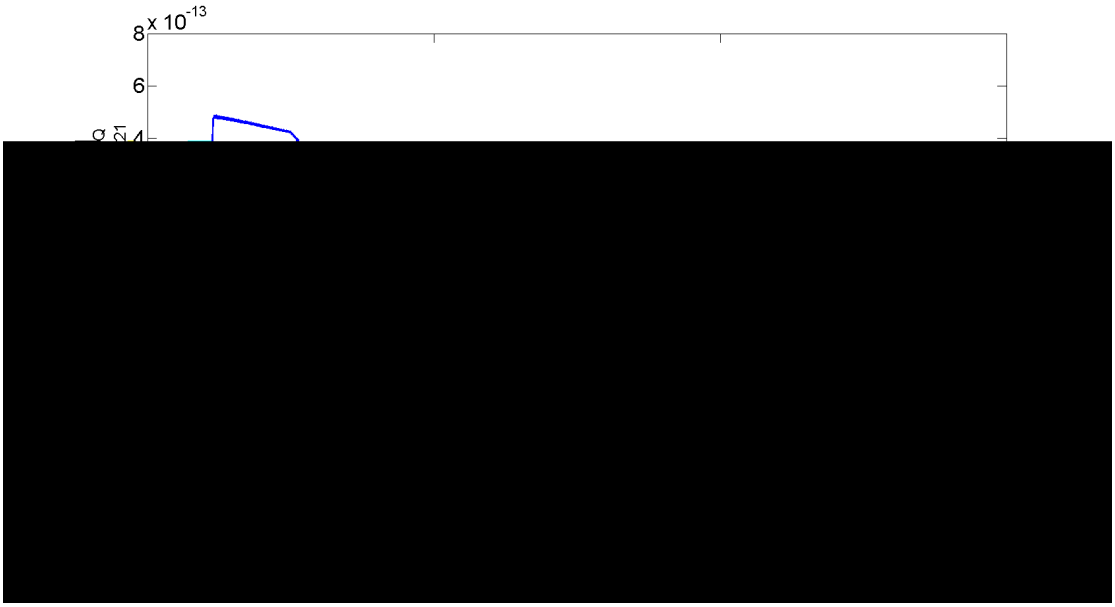

Figure 4.3: l^1 error in Q^k (top) and V^k (bottom) over the time window s 2 [1; 2:5]. Vertical scale of the top plot is O(10 ^{13}), while the vertical scale for the bottom plot is O(10 ^{12})

The errors E_{21}^Q and E_{21}^V for this run are shown in Figure 4.3. In these cases the error appears to be much more random, with an increase in the error at the beginning of the time window which has fallen by the end of the window. This suggests that the approximations Q^k and V^k are being calculated to within rounding error in each time step.

Figure 4.4 shows the error in the right-hand boundary position E_{21}^B for this run of the MPCM. We observe that the error increases throughout the time window, but initially remains very small, with the increase only visible at s 1:1. The di erence between the computed and exact value of the boundary is therefore increasing as time progresses, but still remains within 10 ^{11} at the end of the time window. The maximum di erence

between two successive time steps is approximately 2 10 ^{15}, which suggests that the S Property is being upheld in each individual time step.

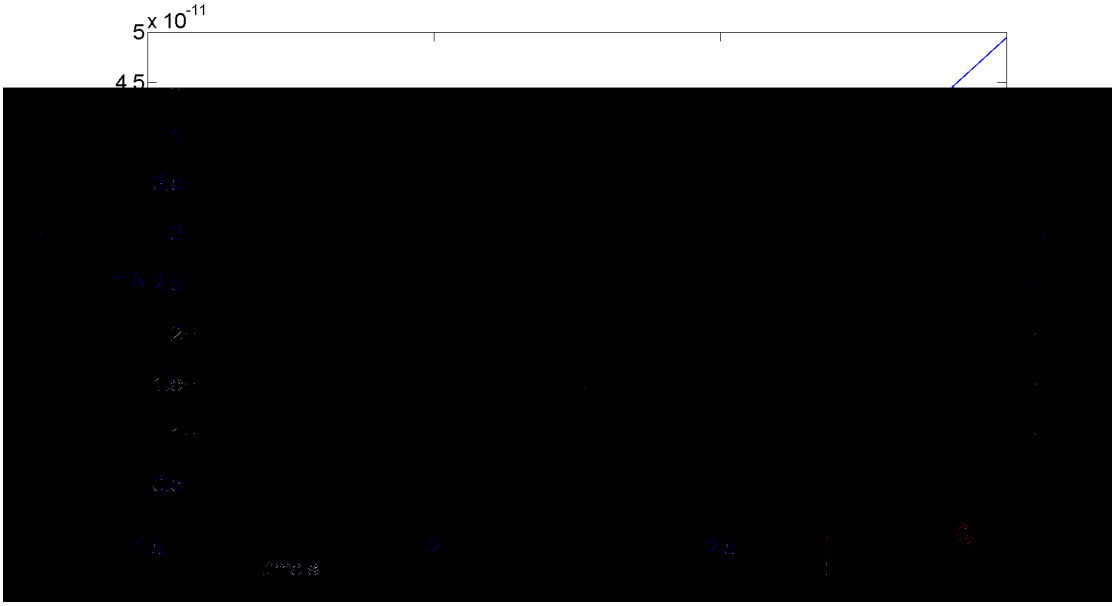

Figure 4.4: Boundary node error E^{B}_{21} over the time window s 2 [1; 2:5]. Vertical scale of plot is O(10 ^{11}).

The errors observed in Figures 4.2{4.4 are a result of the large number of steps being taken (60,000 in the results from this subsection). The di erence in errors between successive time steps can be attributed to rounding error, which indicates that a propagation of rounding error is taking place over multiple time steps.

In the case where we are propagating an initial condition coinciding with a similarity solution, the results displayed in this section demonstrate that a single time step of the required size is more bene cial than running several smaller time steps and is in fact a major advantage of the MPCM. Not only is the risk of node tangling eliminated, but the build up of rounding error highlighted in Figures 4.2{4.4 is also avoided.

We shall now attempt to explain the source of this accumulation of rounding error in the MPCM.

4.4 Propagation of Rounding Error

We observed in §4.3.6 that the l^1 error in the solution does not always remain in the region of rounding error, despite each individual time step of the MPCM arguably possessing the *S* Property. The reason for this appears to be a propagation of rounding

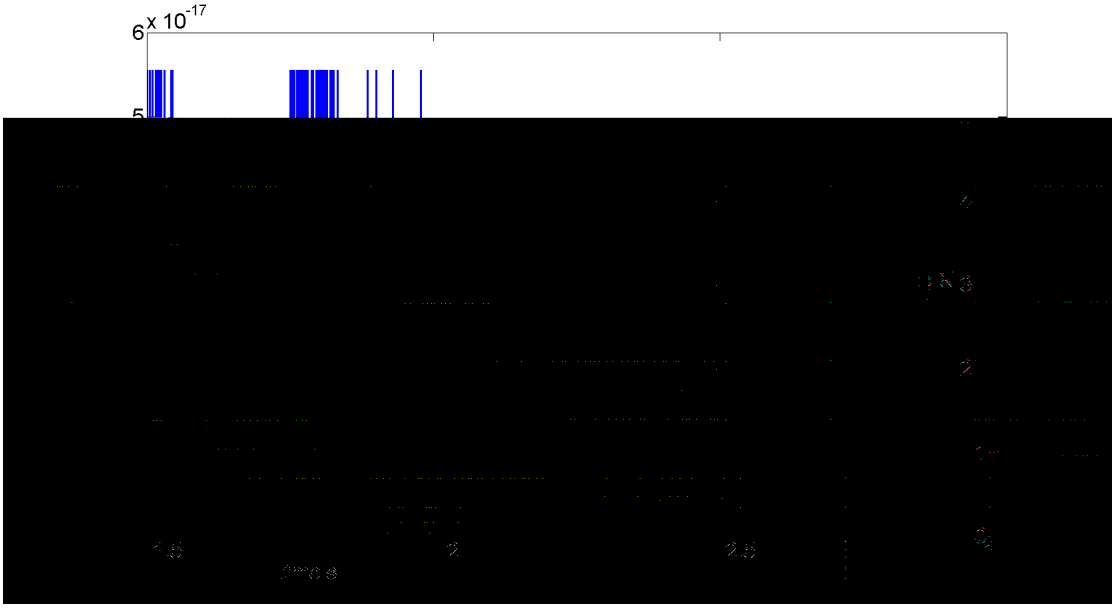

Figure 4.5: l^1 error in the solution when Q, V and X in the method are set equal to the exact value, but the solution is recovered approximately. Scale of y-axis is 10^{-17}.

Performing these alterations produces zero error over the whole time window (not shown), since no quantity is being approximated in the method.

We then repeat the experiment, but recover the solution using the approximation (3.30) while setting Q_i^k, V_i^k, X_i^k equal to the exact values. This solution recovery is expected to be within rounding error of the similarity solution (4.20). Figure 4.5 shows that this is the case, with E_{21}^U remaining proportional to 10^{-17} throughout the time window. Note that in Figure 4.5 the large number of time steps taken has forced the noise-like structure of the errors to be compressed and hence less distinguishable.

Finally, we repeat the experiment one further time. In this instance we update the nodal positions using the time stepping scheme (4.14) and recover the solution using the approximation (3.30), while keeping all other functions equal to their exact values. Figure 4.6 shows the value of E_{21}^U for this experiment. We see that in this case the error increases throughout the time window and has reached a larger value than observed in Figure 4.2 by one order of magnitude. Since the only change made in the method

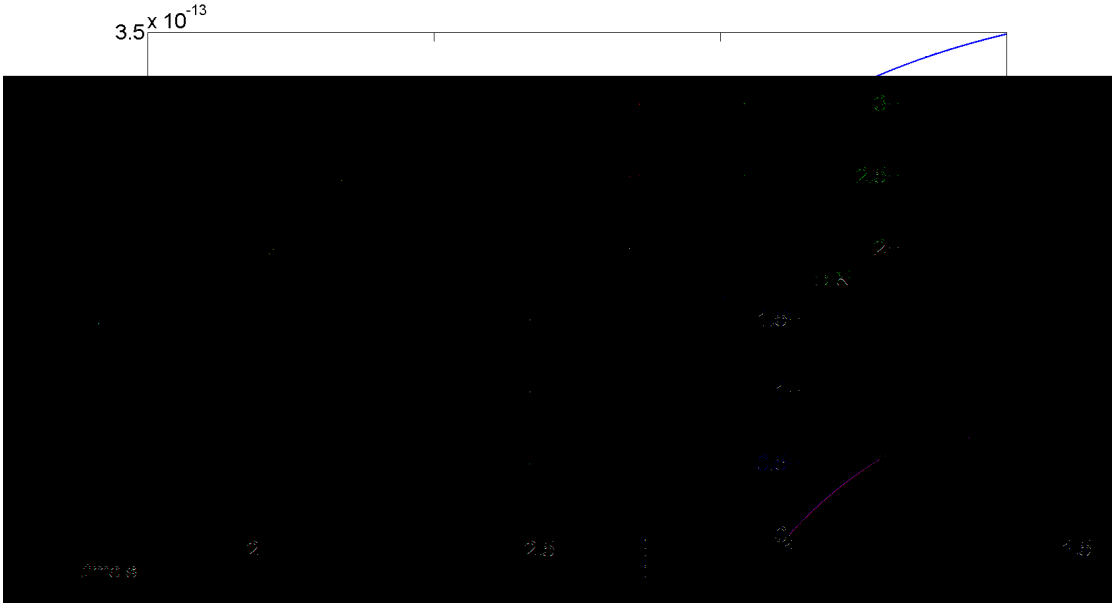

Figure 4.6: l^1 error in the solution when Q and V in the method are set equal to the exact value, but the nodes are updated using a scale-invariant time stepping scheme and the solution is recovered approximately. Scale of y-axis is 10^{13}.

between this and the previous experiment was to perform time stepping of the nodal positions, we conclude that this is the step responsible for the increase of error in the solution.

Interestingly, the original experiment (in which every function was approximated) performs more accurately than this last experiment. This implies that the process of approximating $q(x;s)$ and $v(x;s)$ may help to `correct' some of the error caused by the time stepping scheme.

4.4.2 Examining the Time Stepping Scheme

The scale invariant time stepping scheme (4.13) is

$$X_{k+1} = X_k + \Delta s \hat{v}_k;$$

where Δs is the size of the time step using the scaled time variable s and $\hat{v}_k$ is the mesh velocity in terms of the s time variable. The subscript k denotes the time level (written as a subscript as opposed to the previous superscript time level notation).

In the case where the similarity solution is known, the exact value of the nodes is

$$x^S_{k+1} = x^S_k + \Delta s \hat{v}^S_k; \tag{4.23}$$

where the similarity velocity $\hat{v}^S_k$ is

$$\hat{v}^S_k = \frac{x^S_k}{s_k^{1=}}: \tag{4.24}$$

Substituting (4.24) into (4.23) gives

$$x^S_{k+1} = \left(1 + \frac{\Delta s}{s_k^{1=}}\right) x^S_k:$$

Hence the global error, which we denote by ε^S_k, is de ned by

$$\begin{aligned} \varepsilon^S_k &= x^S_k - X^S_k, \\ \varepsilon^S_{k+1} &= \left(1 + \frac{\Delta s}{s_k^{1=}}\right) \varepsilon^S_k, \\ \varepsilon^S_p &= \prod_{k=0}^{p-1} \left(1 + \frac{\Delta s}{s_k^{1=}}\right) \varepsilon^S_0; \end{aligned} \tag{4.25}$$

at a particular time step p.

We see that there is an ampli cation factor multiplying ε_0 of

$$1 + \frac{s}{s_k^{1=}}\Delta t \; ;$$

which we know is always greater than 1 since all quantities are positive.

We know that ε_0 should equal zero, so $\varepsilon_p = 0$ for all p. In a computational sense however, we will have ε_0 in the region of rounding error (O(10^{-16}), say) and we see that this global error ε_p is magni ed over time.

We now seek to obtain an upper bound for ε_p in order to demonstrate that this error will not grow unboundedly over time.

4.4.3 Bounding the Error

We begin by noting that

$$1 + \frac{s}{s_k^{1=}}\Delta t 1$$

which implies that

$$\sum_{k=1}^{p-1} \frac{1}{(s_0 + k\ s)^5} \quad \int_0^{p-1} \frac{1}{(s_0 + x\ s)^5}\, dx;$$

$$= \frac{1}{4\ s}\ \frac{1}{s_0^4}\ \frac{1}{(s_0 + (p\ 1)\ s)^4}\ :$$

Therefore,

$$\frac{s}{5} \sum_{k=0}^{p-1} \frac{1}{(s_0 + k\ s)^5} \quad \frac{s}{5s_0^5} + \frac{1}{20}\ \frac{1}{s_0^4}\ \left[\ \frac{1}{s_{p\ 1}^4}\ \right] \tag{4.29}$$

Using (4.27) and (4.29) we see that

$$\begin{aligned} {}^{S}_{p} \quad & \exp\ \left[\ \frac{s}{5s_0^5} + \frac{1}{20}\ \frac{1}{s_0^4}\ \left[\ \frac{1}{s_{p\ 1}^4}\ \right]\ \right]\ {}^{S}_{0}; \\ & \exp\ \left[\ \frac{s}{5s_0^5} + \frac{1}{20s_0^4}\ \right]\ {}^{S}_{0}; \end{aligned} \tag{4.30}$$

since $s_p{}^4{}_1 > 0$.

The bound is dependent upon the values of s_0 and s (both fixed). Let us consider the values used to generate the numerical results presented in §4.3.6. In the numerical results run over multiple time steps, values of $s_0 = 1$ and $s \quad 2{:}5 \quad 10^{\ 5}$ were used. The bound (4.30) is then

$${}^{S}_{p} \quad \exp\ \left[\ \frac{1{:}6 \quad 10^{\ 6}}{5} + \frac{1}{20}\ \right]\ {}^{S}_{0};$$

$$1{:}0513\ {}^{S}_{0}:$$

If ${}^{S}_{0}$ is in the region of rounding error, then the bound indicates that the error at the p

4.5 The S Property in the MPCM for the Second-Order Problem

We shall now demonstrate that the implementation of the MPCM for the second-order problem (2.7){(2.9) possesses the S Property (see x3.6) in the l^1 norm for any value of n.

4.5.1 Numerical Results

We now present numerical results for the implementation of the MPCM for the second-order problem (2.7){(2.9) given in x4.5 for values of$n = 1; 1:5; 2$.

We shall perform experiments over a single time step in order to demonstrate that the implementation of the MPCM possesses the$S$ Property in the l^1 norm. We shall then present results for the MPCM run over multiple time steps in order to illustrate how the method performs over multiple steps.

We shall make use of the similarity solution (2.30) from Pattle (1959) and rewritten for general$n$:

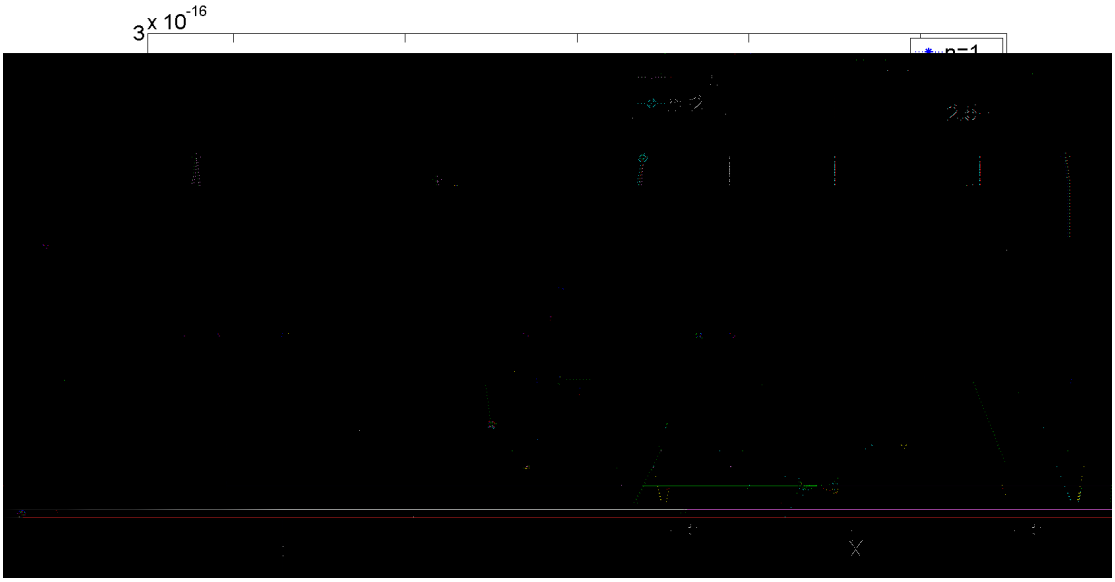

Figure 4.7: Absolute error in the velocity in the MPCM over a single time step. The blue line with asterisk markers denotesn = 1, the green line with + markers denotes n = 2 and the red line with o markers denotesn = 3. Vertical scale on the plot is O(10 ^{16}).

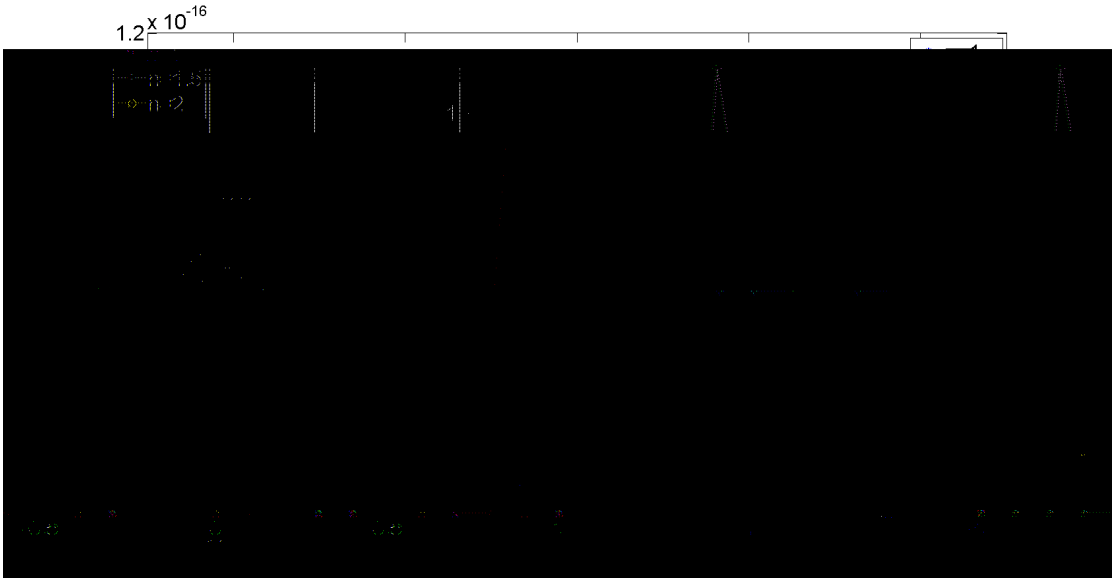

Figure 4.8: Absolute error in the approximate solution in the MPCM over a single time step. The blue line with asterisk markers denotes $n = 1$, the green line with + markers denotes $n = 1:5$ and the red line with o markers denotes $n = 2$. Vertical scale on the plot is $O(10^{-16})$.

We can see that the errors shown in Figures 4.7 and 4.8 are in the region of rounding error, as expected, for each value of n. This indicates that the MPCM is operating as expected over a single time step.

Multiple Time Steps

If we repeat the experiment performed above for a single time step, but instead allow the method to run for multiple time steps, we would anticipate that an accumulation of rounding error would cause the error in the approximate solution to increase over the time window (as observed for the fourth-order implementation in x4.3.6).

We therefore run the MPCM implementation using the same value of s as in the single time step case ($s = 10^{4}$

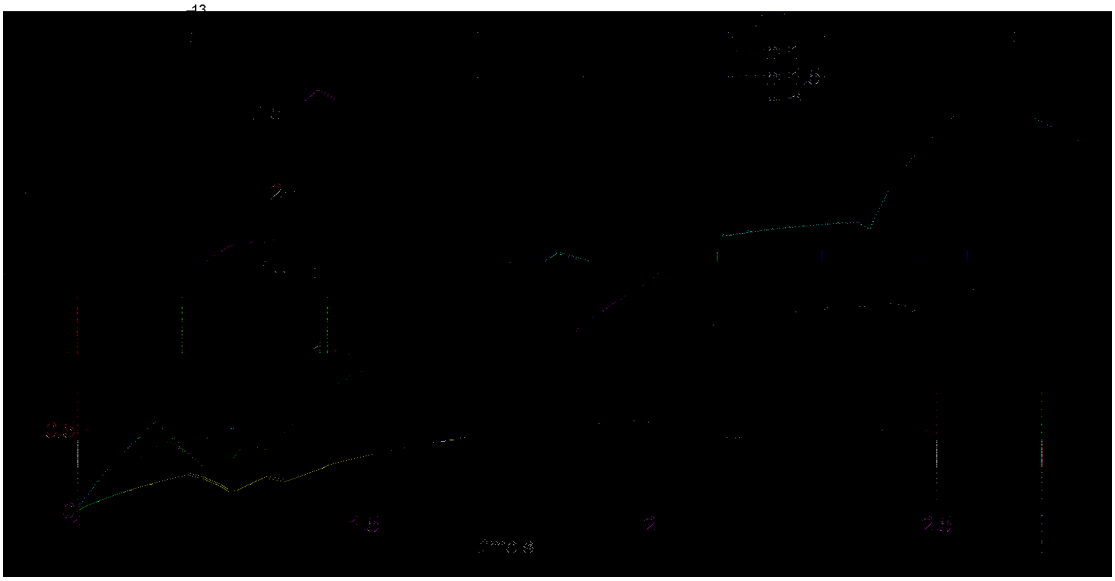

Figure 4.9: l^1 error in the approximate solution U^k from the MPCM over multiple time steps. The solid blue line is for $n = 1$, the dashed green line for $n = 1:5$ and the dash-dotted red line for $n = 2$. Vertical scale of the plot is $O(10^{-13})$.

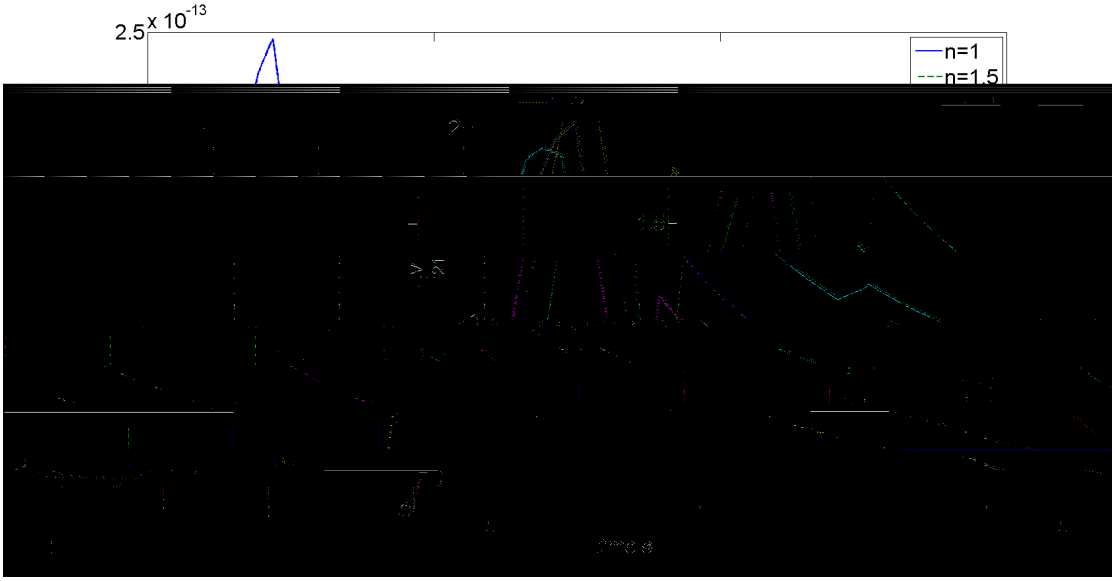

Figure 4.10: l^1 error in the velocity V^k from the MPCM over multiple time steps. The solid blue line is for $n = 1$, the dashed green line for $n = 1:5$ and the dash-dotted red line for $n = 2$. Vertical scale of the plot is $O(10^{-13})$.

We also calculate E_N^B, the error in the right-hand boundary position, given by

$$E_N^B$$

4.6 The S Property in the MPCM for the Sixth-Order Problem

We shall now demonstrate that the implementation of the MPCM given in x3.5 for the sixth-order problem (2.10){(2.12) can be modi ed in order to possess the S Property (see x3.6) in the l^1 norm. As closed-form similarity solutions exist only in the case when $n = 1$, we shall focus on this choice of n in this section. We shall also describe the implementation in terms of the scaled time variable s as we use a scale-invariant time stepping scheme in this implementation.

For clarity, we summarise the problem being considered, for the choice of $= 1$. We seek solutions $u(x; s)$ to the sixth-order PDE

$$\frac{@u}{@s} = \frac{@}{@x}u$$

Similarity solutions to the sixth-order problem (2.10){(2.12) exist in the case when n = 1 which are sextic functions (Smyth and Hill (1988)). The function $p(x;s)$ is therefore a quartic function, while $q(x;s)$ will be a quadratic. We modify our implementation of the MPCM given in x3.5 such that it is exact for sextic $u(x;s)$, quartic $p(x;s)$ and quadratic $q(x;s)$ in the case when$n = 1$.

We are given an initial meshX^0 discretising the domain (s^0) and an approximation U^0 to the initial condition such that $u^0(x) = u^S(x;s^0)$ at each node of the mesh.

4.6.1 Approximating $p(x;s)$

We seek to nd an approximation P^k to $p(x;s)$. The scheme (3.34) given inx3.5.1 has a local truncation error proportional to $\frac{@p}{@x}$at node i of the mesh and as such is not exact for sextic $u(x;s)$, quartic $p(x;s)$. We therefore wish to modify the rst approximation (3.34) such that it becomes exact for quartic$p(x;s)$.

Given U_i^k, $i = 1;:::;N$, (denoted by U_i for ease of notation), we computeP_i^k, $i = 1;:::;N$ (denoted by P_i for ease of notation) as follows.

The modi ed scheme for obtaining P at interior points in the mesh is given by a seven-point central di erencing scheme of the form

$$
\begin{aligned}
P_i = {} & \left(\frac{\frac{U_{i+1}-U_i}{h_i} - \frac{U_i-U_{i-1}}{h_{i-1}}}{\frac{1}{2}(h_i+h_{i-1})}\right) + C_1\left(\frac{P_{i+1}-P_{i-1}}{h_i+h_{i-1}}\right) + C_2\left(\frac{\frac{P_{i+1}-P_i}{h_i} - \frac{P_i-P_{i-1}}{h_{i-1}}}{\frac{1}{2}(h_i+h_{i-1})}\right) \\
& + \frac{C_3}{(h_i+h_{i-1})}\left(\left(\frac{\frac{P_{i+2}-P_{i+1}}{h_{i+1}} - \frac{P_{i+1}-P_i}{h_i}}{\frac{1}{2}(h_{i+1}+h_i)}\right)\left(\frac{\frac{P_i-P_{i-1}}{h_{i-1}} - \frac{P_{i-1}-P_{i-2}}{h_{i-2}}}{\frac{1}{2}(h_{i-1}+h_{i-2})}\right)\right) \\
& + \frac{C_4}{\frac{1}{2}(h_{i+1}+h_i+h_{i-1}+h_{i-2})}\left(\left(\frac{\frac{P_{i+3}-P_{i+2}}{h_{i+2}} - \frac{P_{i+2}-P_{i+1}}{h_{i+1}}}{\frac{1}{2}h_{i+1}(h_{i+2}+h_{i+1})}\right)\left(\frac{\frac{P_{i+2}-P_{i+1}}{h_{i+1}} - \frac{P_{i+1}-P_i}{h_i}}{\frac{1}{2}h_{i+1}(h_{i+1}+h_i)}\right)\right) \\
& \left(\frac{\frac{P_i-P_{i-1}}{h_{i-1}} - \frac{P_{i-1}-P_{i-2}}{h_{i-2}}}{\frac{1}{2}h_{i-2}(h_{i-1}+h_{i-2})}\right)\left(\frac{\frac{P_{i-1}-P_{i-2}}{h_{i-2}} - \frac{P_{i-2}-P_{i-3}}{}}{}\right)
\end{aligned}
$$

$h_{i+2} = X_{i+3} - X_{i+2}$, $h_{i+1} = X_{i+2} - X_{i+1}$, $h_i = X_{i+1} - X_i$, $h_{i-1} = X_i - X_{i-1}$, $h_{i-2} = X_{i-1} - X_{i-2}$ and $h_{i-3} = X_{i-2} - X_{i-3}$ for ease of notation.

The local truncation error $_i$ in the modi ed method (4.36) can be shown to be equal to

$$\begin{aligned} {}_i = {} & p_i + u_i^{00} \left(\frac{h_i - h_{i-1}}{3} \right) + C_1 \left(p_i^0 - \frac{h_i^3 + h_{i-1}^3}{12(h_i + h_{i-1})} \right) + \frac{C_1}{2}(h_i - h_{i-1}) + C_2 \left(p_i^{00} - \frac{h_i^4 - h_{i-1}^4}{60(h_i + h_{i-1})} \right) \\ & + \frac{C_1}{6}(h_i^2 - h_i h_{i-1} + h_{i-1}^2) + \frac{C_2}{3}(h_i - h_{i-1}) + C_3 \left(p_i^{000} - \frac{h_i^5 + h_{i-1}^5}{360(h_i + h_{i-1})} \right) \\ & + \frac{C_1}{24}(h_i^3 - h_i^2 h_{i-1} + h_i h_{i-1}^2 - h_{i-1}^3) + \frac{C_2}{12}(h_i^2 - h_i h_{i-1} + h_{i-1}^2) + \frac{C_3}{2}(h_i - h_{i-1}) + C_4 \, p_i^{0000}; \end{aligned} \tag{4.37}$$

for some $2 (X_{i-3}; X_{i+3})$, where $p_i = p(X_i)$, $p_i^0 = \left.\frac{\partial p}{\partial x}\right|_{X_i}$, $p_i^{00} = \left.\frac{\partial^2 p}{\partial x^2}\right|_{X_i}$, $p_i^{000} = \left.\frac{\partial^3 p}{\partial x^3}\right|_{X_i}$, $p_i^{0000} = \left.\frac{\partial^4 p}{\partial x^4}\right|_{X_i}$.

In order to remove the p_i^0 terms from equation (4.37), we choose our constant C_1 such that

$$C_1 = -\frac{(h_i - h_{i-1})}{3}. \tag{4.38}$$

Similarly, the p_i^{00} terms can be removed from equation (4.37) through utilising the constant C_1 in equation (4.38), and setting

$$\begin{aligned} C_2 &= \frac{C_1}{2}(h_i - h_{i-1}) - \frac{h_i^3 + h_{i-1}^3}{12(h_i + h_{i-1})}; \\ &= \frac{(h_i - h_{i-1})^2}{6} - \frac{h_i^3 + h_{i-1}^3}{12(h_i + h_{i-1})}. \end{aligned}$$

Removing the p_i^{000} terms in (4.37) involves setting the constant C_3 such that

$$\begin{aligned} C_3 &= \frac{h_i^4 - h_{i-1}^4}{60(h_i + h_{i-1})} - \frac{C_1}{6}(h_i^2 - h_i h_{i-1} + h_{i-1}^2) - \frac{C_2}{3}(h_i - h_{i-1}); \\ &= \frac{(2h_i^4 + 8h_{i-1}^4 + 5h_i^3 h_{i-1} - 15 h_i h_{i-1}^3)}{} \end{aligned}$$

Finally, setting

$$C_4 = \frac{h_i^5 + h_{i\ 1}^5}{360(h_i + h_{i\ 1})} \quad \frac{C_1}{24}(h$$

The values of P are then obtained by solving the matrix system

$$SP = TU;$$

where the matrices S, T are of size $N \quad N$. The vector U contains the boundary conditions $U_1 = U_N = 0$ as well as the interior values.

4.6.2 Approximating $q(x;s)$

Given P_i^k, $i = 1;:::;N$ (denoted P_i for ease of notation) detailed in x4.6.1, we now seek an approximation Q_i^k, $i = 1;:::;N$, (denoted by Q_i for ease of notation) which is exact for quadratic $q(x;s)$, quartic $p(x;s)$.

We have previously outlined the steps required in obtaining a quadratic $q(x;s)$ from a quartic $u(x;s)$ in the fourth-order problem (2.3){(2.5) (see x4.3.1), so we use the same steps in obtaining a quadratic $q(x;s)$ from quartic $p(x;s)$ by substituting any references to U by P. We shall restate the appropriate schemes in terms of P for completeness.

For interior points we obtain Q using

$$Q_i \quad \frac{\frac{P_{i+1} \quad P_i}{h_i} \quad \frac{P_i \quad P_{i \ 1}}{h_{i \ 1}}}{\frac{1}{2}(h_i + h_{i \ 1})} + C_1 \frac{Q_{i+1} \quad Q_{i \ 1}}{h_i + h_{i \ 1}} + C_2 \frac{\frac{Q_{i+1} \quad Q_i}{h_i} \quad \frac{Q_i \quad Q_{i \ 1}}{h_{i \ 1}}}{\frac{1}{2}(h_i + h_{i \ 1})};$$

with choices of

$$C_1 = \quad \frac{(h_i + h_{i \ 1})}{3}; \qquad C_2 = \frac{h_i^2 \quad 3h_i h_{i \ 1} + h_{i \ 1}^2}{12};$$

from (4.5) and (4.6).

The local truncation error of this approximation is then proportional to q_i^{000} and will therefore be exact for a quadratic $q(x;s)$.

At the boundary points we use the second-degree Lagrange interpolating polynomials described in x3.5.2.

Q is then found by solving the matrix system

$$SQ = TP;$$

where the matrices S and T are from (4.7) and (4.9) respectively.

4.6.3 Meshvemen

(2005, 2006, 2011)), before showing that through careful selection of basis functions it is possible to construct a finite element method which possesses the S_n Property in the same cases as were demonstrated in this chapter (i.e. $n = 1$ for fourth/sixth order and any n for second-order).

Chapter 5

5.2 Comparison between the MPCM and FEMPCM

In this section we shall provide a brief description of the main di erences between the

5.3 Weak Forms

Motivated by a desire to implement a conservation-based method using finite elements, the required weak forms of equations in the fourth-order problem (2.3)–(2.5) are constructed.

Weak forms are generally constructed by multiplying the equation by a time-dependent test function $w(x;t) \in W$, where W is a suitable test space of functions to be defined in due course. The resulting equation is then integrated with respect to x. In the approach used here weak forms are constructed in a moving framework.

Remark. Throughout this chapter we shall assume that functions in the test space W are suitably smooth. In particular, if $\frac{\partial u}{\partial t}$ and $u^n \frac{\partial q}{\partial x}$ are both L^2 in space then H^1 is a suitable choice for W. This chapter is motivated by the implementation of the FEMPCM, as opposed to a thorough analysis of the required spaces or the spaces in which functions lie, so these details are left deliberately vague.

5.3.1 Weighted Conservation of Mass

In order to determine a weak form for the velocity $v(x;t)$ using conservation, we require the implementation of a weighted conservation of mass principle, as opposed to the local conservation of mass principle described in §3.2.5.

Let $w(x;t)$ be a weight function in W. We then differentiate the weighted mass $\int_{a(t)}^{b(t)} w(x;t)u(x;t)\,dx$ with respect to t using Leibniz's Integral Rule, giving

$$\frac{d}{dt}\int_{a(t)}^{b(t)} w(x;t)u(x;t)\,dx = \int_{a(t)}^{b(t)} \frac{\partial}{\partial t}(w(x;t)u(x;t))\,dx + w(x;t)u(x;t)v(x;t)\Big|_{a(t)}^{b(t)}$$

where

$$v(b(t);t) = \frac{db(t)}{dt}; \qquad v(a(t);t) = \frac{da(t)}{dt}. \tag{5.1}$$

This implies that

$$\frac{d}{dt}\int_{a(t)}^{b(t)} w(x;t)u(x;t)\,dx = \int_{a(t)}^{b(t)} w(x;t)\frac{@u}{@t} + \frac{@w}{@t}u(x;t) + \frac{@}{@x}(wuv)\,dx;$$
$$= \int_{a(t)}^{b(t)} w\frac{@u}{@t} + \frac{@w}{@t}u + \frac{@u}{@x}wv + \frac{@w}{@x}uv + \frac{@v}{@x}wu \quad dx; \quad (5.2)$$
$$= \int_{a(t)}^{b(t)} u \quad \frac{@w}{@t} + v\frac{@w}{@x} \quad + w \quad \frac{@u}{@t} + \frac{@}{@x}(uv) \quad dx;$$

where $v(x;t)$ is any velocity consistent with the boundary velocities (5.1). We now specify the evolution of functions $w(x;t)$ in the space W which vary in time such that $w(x;t)$ is convected with the velocity $v(x;t)$ in the weak sense that

$$\int_{a(t)}^{b(t)} u(x;t) \quad \frac{@w}{@t} + v(x;t)\frac{@w}{@x} \quad dx = 0 \qquad 8w\ 2\ W; \quad (5.3)$$

which reduces equation (5.2) to

$$\frac{d}{dt}\int_{a(t)}^{b(t)} w(x;t)u(x;t)\,dx = \int_{a(t)}^{b(t)} w(x;t) \quad \frac{@}{@x}(uv) + \frac{@u}{@t} \quad dx: \quad (5.4)$$

for all $w\ 2\ W$ satisfying (5.3).

Using equation (2.3a) we may write equation (5.4) in the moving integral form

$$\frac{d}{dt}\int_{a(t)}^{b(t)} w(x;t)u\,dx \quad \int_{a(t)}^{b(t)} w(x;t)\frac{@}{@x}(uv)\,dx = \int_{a(t)}^{b(t)} \frac{@w}{@x} \quad u^n\frac{@q}{@x} \quad dx; \quad (5.5)$$

where $v(x;t)$ is yet to be chosen but must be consistent with (5.1) and (5.3).

We now de ne the velocity $v(x;t)$ implicitly by the weighted conservation of mass principle,

$$\int_{a(t)}^{b(t)} w(x;t)u(x;t)\,dx = \quad (w); \qquad 8w\ 2\ W; \quad (5.6)$$

where the weighted masses(w) remain constant in time. In order to be consistent with the global conservation of mass (2.15) the functions$w(x;t)$ must constitute a partition of unity (this is tacitly assumed in what follows). Equation (5.6) implies that

$$\frac{d}{dt}\int_{a(t)}^{b(t)} w(x;t)u(x;t)\,dx = 0; \qquad 8w\ 2\ W; \quad (5.7)$$

and hence from (5.4) that

$$\int_{a(t)}^{b(t)} w(x;t) \quad \frac{@}{@x}(uv) + \frac{@u}{@t} \quad dx = 0; \qquad 8w\ 2\ W; \quad (5.8)$$

which are respectively weak forms of Lagrangian and Eulerian conservation principles for the function $u(x;t)$.

for $q(x;t)$ and

$$\int_{a(t)}^{b(t)} w(x;t)p(x;t)\,dx = \int_{a(t)}^{b(t)} \frac{\partial w}{\partial x}\frac{\partial u}{\partial x}\,dx; \qquad \forall w \in W. \tag{5.13}$$

for $p(x;t)$.

5.4 An Implementation of the FEMPCM for the Fourth-Order Problem

We now describe an implementation of the FEMPCM for the fourth-order problem (2.3){(2.5). The implementation is based on the weak forms (5.6), (5.10) and (5.11) constructed in x5.3.

5.4.1 Finite Dimensional Subspaces

From the weak forms described inx5.3 we can construct a moving-mesh nite element method. We select nite-dimensional subspaces$S_E^1(t); S^1(t); S^{h1}(t); S^{e1}(t) \in W$, where the subscript E indicates that functions in that subspace satisfy zero Dirichlet bS

5.4.3 Updating the Nodal Positions

Once $Q(x;t)$ and $V(x;t)$ have been obtained, the evolution of each moving point with coordinate $X(t)$ is determined by solving the ODE

$$\frac{dX(t)}{dt} = V(X(t);t);$$

using a suitable time stepping method.

5.4.4 Recovering the Solution $U(x;t)$

Since we have stipulated that the distributed masses $_i$ remain constant over time, we are able to use the evolved moving points $X(t)$ in order to update our numerical solution $U(x;t)$. This is achieved by inverting the distributed weighted form of the Conservation of Mass Principle (5.6) within the appropriate subspace.

one dimensional implementation of this method, following the steps in §5.4. We shall provide an overview of the implementation, since many of the details are found in the given papers (with the main difference being in the solution recovery step, see §5.5.3). Additionally, numerical results on this implementation for $n = 2$ shall be provided.

The first step is to discretise the domain defined by the support of the solution $U(x;t)$ at a given time level t^k into elements using a number of nodes. If there are N nodes in the domain this implies that there are $N - 1$ elements to consider. The nodal positions are given by the vector $X^k = \{X_i^k\}$, where X_i^k is the position of node i, $(i = 1;\ldots;N)$ at time level t^k.

We shall require that $X_i^k < X_{i+1}^k$, for $i = 1;\ldots;N-1$, at any given time t^k otherwise the nodes will have tangled. If the nodes tangle in any time step, the solution recovery step (see §5.5.3) will cause the solution to become negative, which is inadmissible as the the local conservation of mass principle (3.30) will no longer be consistent with conservation of the total mass of the solution. The solution becoming negative also causes the method to fail to preserve nonnegativity of the solution.

Remark. The results given in Section 4.2 (in particular Figure 4) of Baines et al. (2005) demonstrate that in a finite element method for obtaining approximate solutions to the fourth-order problem (2.3)–(2.5), the choice of piecewise linear basis functions is suitable in terms of accuracy of the resulting solution. The authors demonstrate results with fourth order accuracy when $n = 1$ for both the solution and mesh error, when the initial condition is sampled from a similarity solution.

5.5.1 Basis Functions

We next define the basis functions to be used in this implementation of the moving mesh method. We choose the basis functions $w(x;t)$ to be the linear Lagrange polynomials $\phi_i(x;t)$, $i = 1;\ldots;N$, lying in the space of piecewise linear functions. An example of such a function is given in Figure 5.1.

The test functions have a time dependence due to the fact that the nodal positions move with the velocity $v(x;t)$.

5.5.3 An Algorithm for the FEMPCM

With the above information, we next describe the algorithm used to obtain the approximations $Q(x;t)$, $V(x;t)$ and $U(x;t)$:

Obtaining the $Q(x;t)$ Approximation

Given $U(x;t)$ (either from the initial approximation or the previous time step), we obtain $Q(x;t)$. The coefficients

Obtaining the Velocity $V(x;t)$

Using the $Q(x;t)$ obtained in the previous step, we can now obtain the approximation $V(x;t)$ to the velocity $v(x;t)$. The coe cients $V_j(t)$ of

Introducing a Velocity Potential $z(x;t)$

The skew-symmetric matrix $B(U)$ may be ill-conditioned or singular in some cases (when U is uniform and N is odd, $B(U)$ is singular for example) and hence we may be unable to accurately determine V using (5.20) (see Baines et al. (2005, 2011)). We can overcome this issue through the introduction of a velocity potential $z(x;t)$ such that

$$v(x;t) = \frac{\partial z}{\partial x}.$$

The weak form (5.10) now becomes

$$\int_{a(t)}^{b(t)} u\frac{\partial w}{\partial x}\frac{\partial z}{\partial x} + u^{n-1}\frac{\partial q}{\partial x}\,dx = 0; \qquad \forall w \in W.$$

We then seek the finite element approximation $Z(x;t)$ of $z(x;t)$ for a given $U(x;t) \in S_E^1(t)$, $Q(x;t) \in S^{-1}(t)$ in the form: Find $Z(x;t) \in S^{-1}(t)$ such that

$$\int_{a(t)}^{b(t)} U(x;t)\frac{\partial w}{\partial x}\frac{\partial Z}{\partial x} + U^{n-1}\frac{\partial Q}{\partial x}\,dx = 0; \qquad \forall w \in S^{-1}(t); \tag{5.23}$$

in place of (5.15).

Choosing w in (5.23) to be one of the $\phi_i(x;t)$ functions, writing $Q(x;t)$ and $U(x;t)$ using (5.17b) and (5.17a) respectively, and writing $Z(x;t)$ as the linear combination

$$Z(x;t) = \sum_{j=1}^{N} \phi_j(x;t) Z_j(t);$$

we can obtain the coefficients $Z_j(t)$ by solving the matrix system

$$\mathcal{K}^e(U)Z = \mathcal{K}^e(U^n)Q;$$

where $\mathcal{K}^e(U)$ and $\mathcal{K}^e(U^n)$ have entries given by (5.22) (using $n = 1$ in (5.22) in the case of the former).

The $\mathcal{K}^e(U)$ matrix is singular (see Remark 5.1 below) and hence non-invertible, but we may obtain a unique Z through the imposition of a suitable constraint, such as

$$\sum_{j=1}^{N} Z_j(t) = 0; \tag{5.24}$$

thereby also putting a constraint on $S^{-1}(t)$. There is no loss of generality since $z(x;t)$ is a potential and has a free parameter.

Remark 5.1. The matrix $\mathbf{K}^e(U)$ is singular as a result of the choice of basis functions made in this implementation of the FEMPCM. Over a single element, the basis functions have been chosen in order that they sum to one (i.e. that they form a partition of unity). The derivative of such functions will therefore sum to zero over a single element.

This results in the determinant of the $\mathbf{K}^e(U)$ matrix being equal to zero, and hence that the matrix is singular.

Using $Z(x;t)$ to obtain $V(x;t)$

An additional step is required to obtain the velocity $V(x;t)$ from $Z(x;t)$. We use the weak form

$$\int_{a(t)}^{b(t)} w(x;t) \quad v(x;t) \quad \frac{@z}{@x} \; dx = 0; \qquad 8w \; 2 \; W;$$

from which we seek the nite element approximation $V(x;t)$ of $v(x;t)$ given $Z(x;t) \; 2$ $S^1(t)$ in the form: Find $V \; 2 \; S^1_h(t)$ such that

$$\int_{a(t)}^{b(t)} w(x;t)V(x;t)\,dx = \int_{a(t)}^{b(t)} w(x;t)\frac{@Z}{@x}dx.023\ 0\ Td\ [())-383051$$

where the vector V^k contains the coefficients $V_j(t^k)$ of the velocity from (5.17c) calculated from (5.20) and where Δt denotes the size of the time step.

Recovery of Solution $U(x;t)$ on the Updated Mesh

With the new nodal positions calculated, we can recover the solution $U(x;t)$ using the updated mesh values X^{k+1}. This step is the main difference between the implementation presented in Baines et al. (2005, 2006, 2011) and the method presented here. The coefficients $U_j(t)$ in (5.17a) can be found by solving the matrix system

$$AU = \Gamma; \tag{5.25}$$

where A is of dimension $N \times N$ 2 and has time-dependent entries given by

$$A_{ij}(t^k) = \begin{cases} \displaystyle\int_{X_1^k}^{X_2^k} {}_1(x;t)\ {}_j(x;t)\,dx; & \text{for } i = 1; \\ \displaystyle\int_{X_{i-1}^k}^{X_{i+1}^k} {}_i(x;t)\ {}_j(x;t)\,dx; & \text{for } i = 2;\ldots;N-1; \\ \displaystyle\int_{X_{N-1}^k}^{X_N^k} {}_N(x;t)\ {}_j(x;t)\,dx; & \text{for } i = N; \end{cases}$$

for $j = 2;\ldots;N-1$. The lack of values for $j = 1$ and N is due to the reduced number of ${}_j(x;t)$ in $U(x;t)$ (see (5.17a)). The values of $\Gamma = \{\Gamma_i\}$, $i = 1;\ldots;N$, can be obtained using the initial condition $u^0(x)$, where

$$\Gamma_i = \int_{a(t^0)}^{b(t^0)} {}_i(x;t^0)u^0(x)\,dx; \qquad \text{for } i = 1;\ldots;N: \tag{5.26}$$

The integrals in (5.26) can be obtained using a suitable numerical integration scheme (such as Gaussian quadrature), or in some cases can be evaluated exactly depending upon the choice of $u^0(x)$.

The system given in (5.25) is overdetermined at present, since the matrix A has dimension $N \times N$ 2. In order to alleviate this issue we may introduce modified basis

functions $\tilde{\gamma}_i(x;t$

Then,

$$\tilde{A}U = \tilde{\sim}; \tag{5.29}$$

where $\tilde{A}$ is a mass matrix of dimension $N - 2 \times N - 2$ with entries given by

$$\tilde{A}_{ij}(t^k) = \int_{a(t)}^{b(t)} \tilde{\sim}_i(x;t)\ _j(x;t)\,dx; \qquad \text{for } i;j = 1;\ldots;N - 2:$$

The coefficients $U_j(t)$ obtained through solving the matrix system (5.29) give $U(x;t)$ at the interior mesh points. If we had solved for $U(x;t)$ at all N nodes of the mesh, the zero boundary condition (2.5a) would only have been enforced weakly and hence U_1 and U_N would have been unknowns (Baines et al. (2005)). By introducing the modified basis functions with their partition of unity property we are able to solve for unknown $U_2;\ldots;U_{N-1}$ and combine this information with the boundary condition $U_1 = U_N = 0$ to ensure that the zero boundary condition (2.5a) is strongly enforced. Further details of the strong enforcement of boundary conditions can be found in Hubbard et al. (2009).

5.5.4 Numerical Results

We shall now present numerical results for this implementation of the FEMPCM described in §5.5, for the choice of $n = 2$.

We shall perform experiments on meshes of different numbers of nodes. In each successive experiment we increase the number of nodes such that the initial mesh spacing is halved (on an initially uniformly-spaced mesh), resulting in meshes of $N = 21; 41, 81$ and 161 nodes. In each experiment the method is run over the time window $t \in [1; 1{:}25]$, with the time step size Δt kept proportional to $1/N^4$ in order to reduce the risk of node tangling occurring. Specifically, we choose Δt such that

$$\Delta t = \frac{10^{-4}}{16^j} \qquad \text{for } j = 1;2;\ldots:$$

The initial condition used in the experiments is

$$u^0(x) = \frac{1}{120}\left(4 - x^2\right)_+^2;$$

which is incidentally a similarity solution in the $n = 1$ case at time $t^0 = 1$ but has no such relevance to $n = 2$. This initial condition is symmetric about $x = 0$ with initial

boundary positions at $x = \quad 2$. This initial condition is used to calculate the $_i$ values and the initial approximation $U^0(x)$ following the steps in x5.5.3.

At the end of the time window, we record the nodal values of the approximations $U(x; 1{:}25)$, $Q(x; 1{:}25)$ and $V(x; 1{:}25)$. We also record the position of the right-hand boundary point at each time step of the method.

Figures 5.3, 5.4 and 5.5 show the nodal values for the various meshes. In the case of the approximations $U(x; 1{:}25)$ and $Q(x; 1{:}25)$ there appears to be little visible di erence between the various meshes. In the case of the approximate velocity $V(x; 1{:}25)$ however, there is a noticeable di erence in the nodal values near to the boundary points.

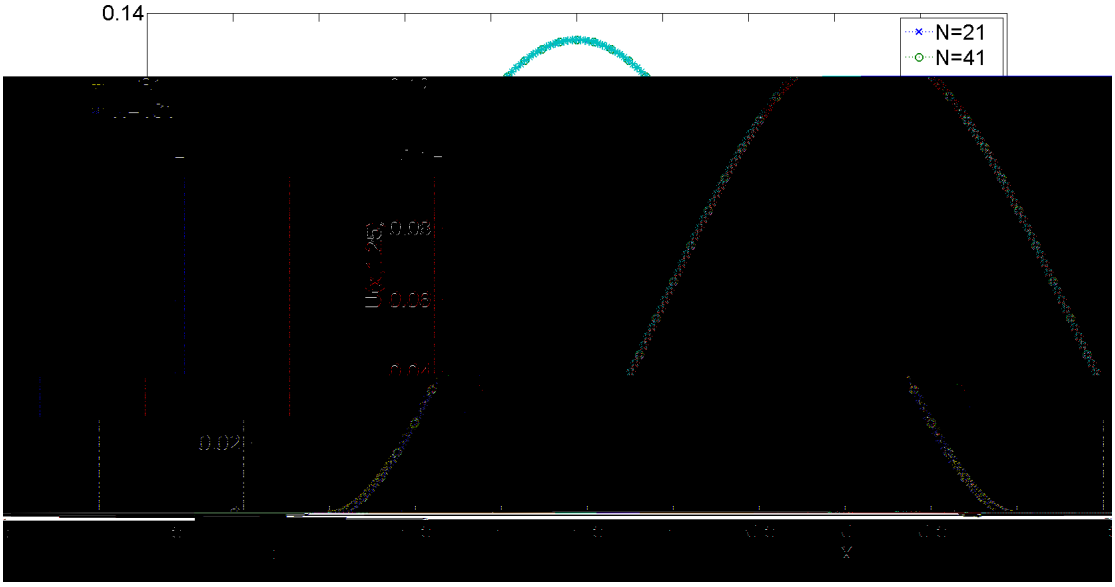

Figure 5.3: Nodal values of $U(x; 1{:}25)$ on meshes of $N = 21$ (blue xs), 41 (green os), 81 (red +s) and 161 (cyan s) nodes.

This di erence in the nodal values of the velocity $V(x; 1{:}25)$ causes the nodes to move di erently in each of the di erent meshes. This is illustrated clearly in the case of the right-hand boundary, which is shown in gure 5.6. As the number of nodes in the mesh increases, the boundary moves a smaller distance over the time window. Examining the approximate velocities in gure 5.5, the boundary velocity is approaching zero as the

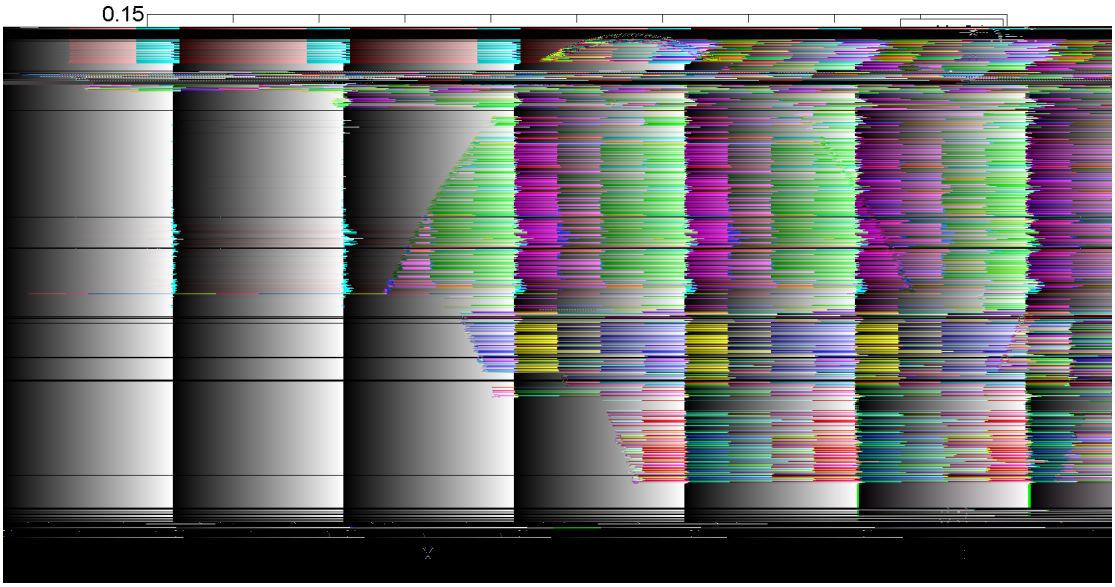

Figure 5.4: Nodal values of $Q(x; 1{:}25)$ on meshes of $N = 21$ (blue xs), 41 (green os), 81 (red +s) and 161 (cyan s) nodes.

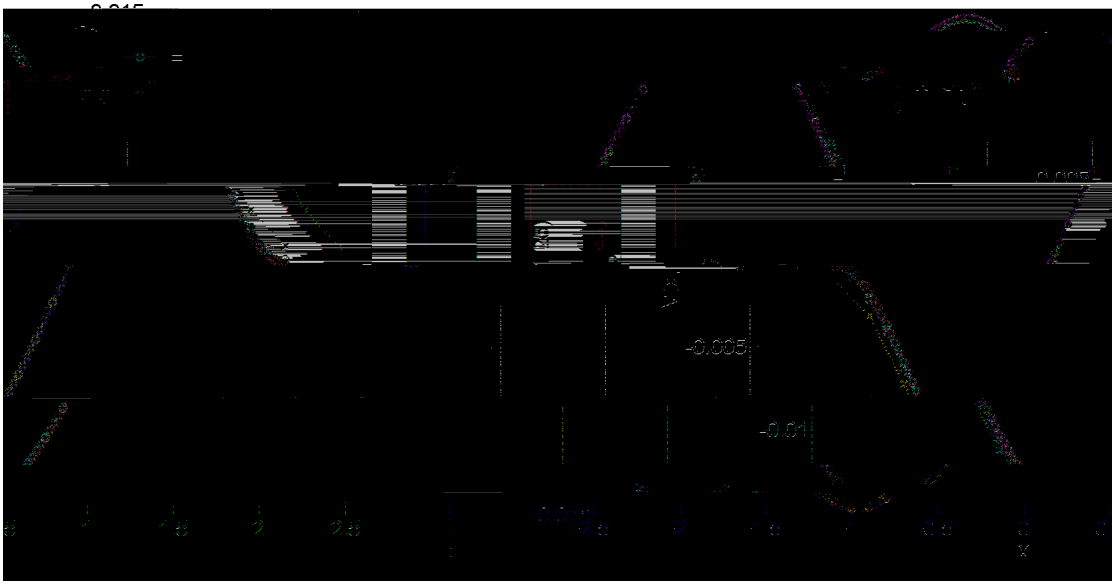

Figure 5.5: Nodal values of $V(x; 1{:}25)$ on meshes of $N = 21$ (blue xs), 41 (green os), 81 (red +s) and 161 (cyan s) nodes.

number of nodes increases. This will then cause the boundary to move outwards less (and is consistent with the expected velocity as we shall discuss in Chapter 6).

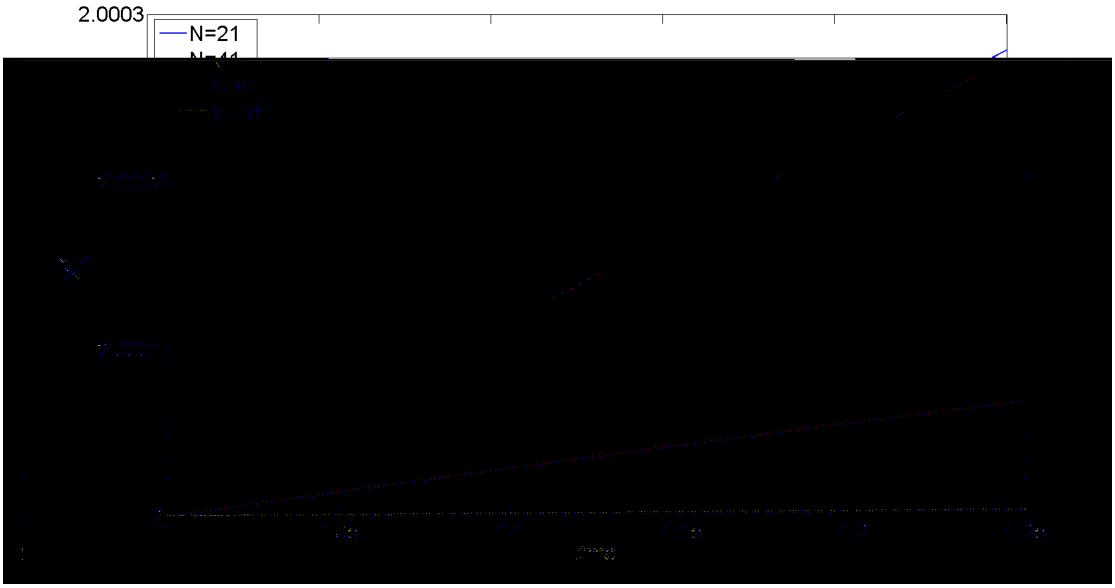

Figure 5.6: Position of the right-hand boundary node on meshes of N = 21 (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes.

Figure 5.6 also provides some i7cFurmaion is at phe pons28(vi28(e)rgenc)-347(pf)-347(phe)-3476lir

$n = 1$ only, possesses the S Property in the L^2 norm.

The steps involved in the implementation are similar to those described in §5.5, which leads to some repetition between the information in this section and §5.5. The basis functions used in this section are of higher order however, and as such many of the details are more involved. We shall therefore describe the implementation in more detail than that given in §5.5. We shall also be using the scale-invariant time stepping scheme (5.47), so we replace any reference to the time variable t with the new scaled time variable s in this section.

The similarity solution for the fourth-order problem (2.3)–(2.5) with $n = 1$ is a quartic function (Smyth and Hill (1988)). This results in the exact values of $q(x;s)$ and $v(x;s)$ being quadratic and linear respectively. We shall therefore choose our finite-dimensional subspaces $S_E^1(s)$, $S^1(s)$ & $S^{-1}(s)$ in this particular implementation such that $U(x;s)$ is piecewise quartic, $Q(x;s)$ piecewise quadratic and $V(x;s)$ piecewise linear. Combined with a suitable choice of piecewise quartic, quadratic and linear basis function in each weak form and scale-invariant time stepping, the method can be shown to possess the S Property in the L^2 norm.

5.6.1 Discretisation

The first step is to discretise the domain defined by the support of the solution $U(x;s)$ at a given time level s^k into elements, using a number of nodes. If there are N nodes in the domain this implies that there are $N - 1$ elements to consider. The nodal positions are given by the vector $X^k = \{X_i^k\}$, where X_i^k is the position of node i, $(i = 1;\ldots;N)$ at time level s^k.

As stated in §5.5, we shall require that $X_i^k < X_{i+1}^k$, at any given time s^k, for $i = 1;\ldots;N-1$, otherwise the nodes are considered to have tangled (with the consequences regarding resulting loss of consistency between local and global mass conservation and loss of positivity of the solution applying).

5.6.2 Basis Functions

We next define the basis functions to be used throughout this implementation of the moving mesh method for the fourth order problem (2.3)–(2.5) with $n = 1$. We choose the basis functions $w(x,s)$ to be Lagrange polynomials of either first, second or fourth order, as follows:

For functions lying in the space of piecewise quartics we set $w(x,s)$ to be a fourth order Lagrange polynomial denoted by $\phi_i(x,s)$, for $i = 1,\ldots,4N-3$. In a given element we require five different $\phi_i(x,s)$ to specify a unique quartic function, resulting in a total of $4N-3$ basis functions $\phi_i(x,s)$ over the mesh.

For the space of piecewise quadratics we choose $w(x,s)$ to be a quadratic Lagrange polynomial denoted by $\psi_i(x,s)$ for $i = 1,\ldots,2N-1$. There are three different $\psi_i(x,s)$ required to specify a unique quadratic function over a given element, resulting in $2N-1$ basis functions $\psi_i(x,s)$ over the mesh.

Finally, we define $w(x,s)$ to be a linear Lagrange polynomial, denoted by $\theta_i(x,s)$, $i = 1,\ldots,N$, chosen for functions lying in the space of piecewise linears. There are two different $\theta_i(x,s)$ required to define a unique linear function over a given element, resulting in a total of N basis functions $\theta_i(x,s)$ over the mesh.

In Figure 5.7 we plot examples of these functions over a single element, with $\theta_i(x,s)$ given in the left plot, $\psi_i(x,s)$ in the central plot and $\phi_i(x,s)$ in the right plot.

The test functions have a time dependence due to the fact that the nodal positions move with the velocity $v(x,s)$.

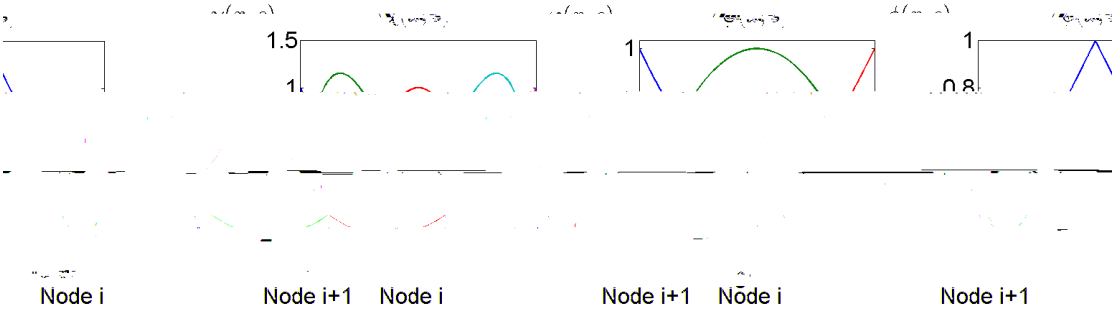

Figure 5.7: Plot of the various choices of Lagrange polynomial used in this implementation of the Finite Element Method, plotted over a single element.

5.6.3 Finite Dimensional Subspaces

Throughout x5.6, our function approximations are chosen to lie in nite dimensional subspaces of the test space given by

$$S_E^1(s) = \text{span}\{\ _j(x;s)\}; \qquad \text{for } j = 2;\dots;\hat{N} \quad 1;$$

$$S^1(s) = \text{span}\{'_j(x;s)\}; \qquad \text{for } j = 1;\dots;\bar{N};$$

$$S^1(s) = \text{span}\{\ _j(x;s)\}; \qquad \text{for } j = 1;\dots N;$$

$$S^1(s) = \text{span}\{\ _j(x;s)\}; \qquad \text{for } j = 1;\dots;\hat{N}:$$

where we have introduced $\hat{N} = 4N \quad 3$ and $\bar{N} = 2N \quad 1$ for ease of notation. The reduced span of $S_E^1(s)$ is due to the boundary conditions $u = 0$ and therefore there are no test functions attributed to the boundary points in this case.

5.6.4 Obtaining the Q(x;s) Approximation

Given $U(x;s)$, we can then nd the approximation $Q(x;s)$ to $q(x;s)$ from (5.14).

We choose our test function $w(x;s)$ in (5.14) to be one of the $'_i(x;s)$ $(i = 1;\dots;\bar{N})$.

Due to the properties of the $\phi_i(x;s)$ functions this produces the system of equations

$$\int_{a(s)}^{b(s)} \phi_1(x;s)Q(x;s)\,dx = \int_{a(s)}^{b(s)} \phi_1^0(x;s)U^0(x;s)\,dx; \qquad \text{for } i = 1;$$

$$\int_{a(s)}^{b(s)} \phi_i(x;s)Q(x;s)\,dx = \int_{a(s)}^{b(s)} \phi_i^0(x;s)U^0(x;s)\,dx; \qquad \text{for } i = 2;\ldots;N^{\tau}-1;$$

$$\int_{a(s)}^{b(s)} \phi_{N^{\tau}}(x;s)Q(x;s)\,dx = \int_{a(s)}^{b(s)} \phi_{N^{\tau}}^0(x;s)U^0(x;s)\,dx; \qquad \text{for } i = N^{\tau}.$$

Writing $Q(x;s)$ as a linear combination of the $\phi_j(x;s)$,

$$Q(x;s) = \sum_{j=1}^{N^{\tau}} \phi_j(x;s)Q_j(s); \tag{5.32}$$

and $U(x;s)$ as a linear combination of the $\phi_j^0(x;s)$

The time-dependent entries of M are given by

$$M_{ij}(s^k) = \begin{cases} \displaystyle\int_{x_1^k}^{x_2^k} \varphi_1(x;s)\varphi_j(x;s)\,dx & \text{for } i = 1; \\ \displaystyle\int_{x_{i-1}^k}^{x_{i+1}^k} \varphi_i(x;s)\varphi_j(x;s)\,dx & \text{for } i = 2;\dots;\bar{N}-1; \\ \displaystyle\int_{x_{\bar{N}-1}^k}^{x_{\bar{N}}^k} \varphi_{\bar{N}}(x;s)\varphi_j(x;s)\,dx & \text{for } i = \bar{N}; \end{cases}$$

for $j = 1;\dots;\bar{N}$, while time-dependent entries of K are given by

$$K_{ij}(s^k) = \begin{cases} \displaystyle\int_{1}^{\varphi_N} \cdots ; \quad x^k_i(x;s)x \quad \text{for } i\,)\,dx \\ \displaystyle\int_{x^k} \cdots \quad \varphi_1(x;s) \quad ; \end{cases}$$

We then make use of the finite element approximation (5.23), which we repeat here for clarity (in terms of s): Find $Z(x;s) \in S^1(s)$ such that

$$\int_{a(s)}^{b(s)} U(x;s)\left(\frac{\partial w}{\partial x}\frac{\partial Z}{\partial x} + \frac{\partial Q}{\partial x}\right) dx = 0; \qquad \forall w \in S^1(s); \tag{5.37}$$

which is used instead of (5.15).

Choosing w in (5.37) to be one of the $\psi_i(x;s)$ functions, writing $Q(x;s)$ using (5.32), and writing $Z(x;s)$ as the linear combination

$$Z(x;s) = \sum_{j=1}^{N_T} \psi_j(x;s) Z_j(s); \tag{5.38}$$

we can obtain the coefficients $Z_j(s)$ by solving the matrix system

$$K^e(U)Z = -K^e(U)Q; \tag{5.39}$$

where $K^e(U)$ is of dimension $N_T \times N_T$ and has time-dependent entries given by

$$K^e(U)_{ij}(s^k) = \begin{cases} \displaystyle\int_{x_2^k}^{x_1^k} U(x;s)\,\psi_1'(x;s)\,\psi_j'(x;s)\,dx & \text{for } i = 1; \\ \displaystyle\int_{x_{i+1}^k}^{x_i^k} U(x;s)\,\psi_i'(x;s)\,\psi_j'(x;s)\,dx & \text{for } i = 2;\ldots;N-1; \\ \displaystyle\int_{x_N^k}^{x_{N-1}^k} U(x;s)\,\psi_N'(x;s)\,\psi_j'(x;s)\,dx & \text{for } i = N; \end{cases} \tag{5.40}$$

for $j = 1;\ldots;N_T$.

The $K^e(U)$ matrix is singular and hence non-invertible (see Remark 5.1). We may obtain a unique Z through the imposition of a suitable constraint, such as (5.24)

$$\sum_{j=1}^{N_T} Z_j(s) = 0;$$

thereby putting a constraint on $S^1(s)$. There is no loss of generality since $z(x;t)$ is a potential with a free parameter.

5.6.6 Obtaining the Velocity $V(x;s)$

The velocity $V(x;s)$ can be obtained from $Z(x;s)$ through the weak form

$$\int_{a(s)}^{b(s)} w(x;s)\left(v(x;s) - \frac{\partial z}{\partial x}\right) dx = 0; \qquad \forall w \in W; \tag{5.41}$$

from which we seek the nite element approximation $V(x;s)$ of $v(x;s)$ given $Z(x;s) \in S^1(s)$ in the form: Find $V \in S^{h1}(s)$ such that

$$\int_{a(s)}^{b(s)} w(x;s)V(x;s)\,dx = \int_{a(s)}^{b(s)} w(x;s)\frac{@Z}{@x}dx; \qquad 8w \in S^{h1}(s): \tag{5.42}$$

We select $w(x;s)$ in (5.42) to be one of the $_i(x;s)$ functions ($i = 1;:::;N$), resulting in the system of equations

$$\begin{aligned}
\int_{a(s)}^{b(s)} {}_1(x;s)V(x;s)\,dx &= \int_{a(s)}^{b(s)} {}_1(x;s)\frac{@Z}{@x}dx; && \text{for } i = 1;\\
\int_{a(s)}^{b(s)} {}_i(x;s)V(x;s)\,dx &= \int_{a(s)}^{b(s)} {}_i(x;s)\frac{@Z}{@x}dx; && \text{for } i = 2;:::;N \quad 1;\\
\int_{a(s)}^{b(s)} {}_N(x;s)V(x;s)\,dx &= \int_{a(s)}^{b(s)} {}_N(x;s)\frac{@Z}{@x}dx; && \text{for } i = N:
\end{aligned}$$

Expanding $V(x;s)$ as the linear combination

$$V(x;s) = \sum_{j=1}^{N} {}_j(x;s)V_j(s); \tag{5.44}$$

and using the expansion for $Z(x;s)$ given by (5.38), we obtain the modi ed system of equations

$$\sum_{j=1}^{N} V_j(s)\int_{a(s)}^{b(s)} {}_1(x;s)\ {}_j(x;s)\,dx = \sum_{j=1}^{N} Z_j(s)\int_{a(s)}^{b(s)} {}_1(x;s)'\ {}^0_j(x;s)\,dx;$$

$$\text{for } i = 1;$$

$$\sum_{j=1}^{N} V_j(s)\int_{a(s)}^{b(s)} {}_i(x;s)\ {}_j(x;s)\,dx = \sum_{j=1}^{N} Z_j(s)\int_{a(s)}^{b(s)} {}_i(x;s)'\ {}^0_j(x;s)\,dx;$$

$$\text{for } i = 2;:::;N \quad 1;$$

$$\sum_{j=1}^{N} V_j(s)\int_{a(s)}^{b(s)} {}_N(x;s)\ {}_j(x;s)\,dx = \sum_{j=1}^{N} Z_j(s)\int_{a(s)}^{b(s)} {}_i(x;s)'\ {}^0_j(x;s)\,dx;$$

$$\text{for } i = N:$$

The coe cients of $Z(x;t)$ can be obtained by solving the matrix system

$$MV = BZ; \tag{5.46}$$

where the M matrix is as given in (5.19), and the B matrix is of dimension $N \times N_T$ and has time-dependent entries given by

$$B_{ij}(s^k) = \begin{cases} \int_{x_1^k}^{x_2^k} {}_1(x;s)\,{}_j^{0}(x;s)\,dx & \text{for } i \end{cases}$$

Since the $_{i}$

with $U_1^0 = U_{\hat{N}}^0 = 0$ from the zero boundary conditions. Once substituted, the partitioned form (5.16) becomes equivalent to solving the system of equations

$$\sum_{j=1}^{\hat{N}-2} U_j^0 \int_{a(s^0)}^{b(s^0)} \tilde{\varphi}_1(x;s^0)\,\varphi_j(x;s^0)\,dx = \tilde{\alpha}_1; \qquad \text{for } i = 1;$$

$$\sum_{j=1}^{\hat{N}-2} U_j^0 \int_{a(s^0)}^{b(s^0)} \tilde{\varphi}_i(x;s^0)\,\varphi_j(x;s^0)\,dx = \tilde{\alpha}_i; \qquad \text{for } i = 2;\ldots;\hat{N}-3;$$

$$\sum_{j=1}^{\hat{N}-2} U_j^0 \int_{a(s^0)}^{b(s^0)} \tilde{\varphi}_{\hat{N}-2}(x;s^0)\,\varphi_j(x;s^0)\,dx = \tilde{\alpha}_{\hat{N}-2}; \qquad \text{for } i = \hat{N}-2;$$

which gives the best L^2 fit to the initial condition in the space $S_E^1(s^0)$. These equations can be written in the matrix form

$$\tilde{A}^0 U^0 = \tilde{\alpha}; \tag{5.51}$$

where $\tilde{A}$ is a mass matrix of dimension $\hat{N}-2 \times \hat{N}-2$ with time-dependent entries given by

principle (5.16), the system of equations becomes

$$\sum_{j=1}^{N} U_j(s) \int_{a(s)}^{b(s)} \tilde{\alpha}_1(x;s)\, \tilde{\alpha}_j(x;s)\, dx = \tilde{\alpha}_1; \qquad \text{for } i = 1;$$

$$\sum_{j=1}^{N} U_j(s) \int_{a(s)}^{b(s)} \tilde{\alpha}$$

where the superscript 0 indicates the initial values of the quantity considered.

Remark. Since (5.54) is derived from

$$\int_{a(s^0)}^{b(s^0)} \gamma_i (U^0(x) - u^0(x))\, dx = 0; \qquad \text{for } i = 1; \ldots; \hat{N} - 2;$$

using the γ_i values, it follows that the initial approximation $U^0(x)$ is identical to the similarity solution at time s^0 to within rounding error since the self similar solution $u^S(x; s) \in S_E^1(s)$, the space of quartic functions.

Obtaining $Q(x; s^0)$

Given the initial approximation $U^0(x)$, we obtain the coefficients Q_j^0 in (5.32) using equation (5.35):

$$M^0 Q^0 = K^0 U^0.$$

Remark. The exact $q(x; s)$ function is a quadratic in the case when $n = 1$. The approximation $Q^0(x)$ to $q^0(x)$ calculated in §5.6.4 is also quadratic. Since the initial function $U^0(x)$ used in calculating $Q^0(x)$ is identical to the similarity solution to within rounding error, $Q^0(x)$ is also exact within rounding error.

Obtaining $Z(x; s^0)$

Given $Q^0(x)$ and $U^0(x)$, we obtain $Z^0(x)$ by solving the matrix system (5.39):

$$\mathcal{K}^e(U^0) Z^0 = \mathcal{K}^e(U^0) Q^0;$$

giving the coefficients Z_j^0 in (5.38).

Remark. The exact velocity potential $z(x; s)$ is quadratic and the approximation $Z^0(x)$ calculated in §5.6.5 is piecewise quadratic. Hence the approximate velocity potential $Z^0(x; s^0)$ will be identical to the exact velocity potential to within rounding error at time s^0.

Calculating the Nodal Velocities

The velocity $V^0(x)$ is calculated using the previously calculated $Z^0(x)$ by solving the matrix system (5.46),

$$M V^0 = B^0 Z^0;$$

giving the coe cients V_j^0

which implies from (5.55) that

$$\int_{a(s^1)}^{b(s^1)} \tilde{\gamma}_i(x;s^1)(U(x;s^1) - u^S(x;s^1))\,dx = 0; \qquad \text{for } i = 1;\ldots;\hat{N} - 2;$$

and hence that $U(x;s^1)$ is the L^2 best fit to the similarity solution over one time step.

Remark. In practice the best fit of the solution is preserved only to within rounding error. The arguments in this section are not a proof that this implementation of the FEMPCM possesses the S Property, but by choosing our approximations to lie in the appropriate space matching the order of the function we wish to approximate, the method should compute an approximation to within rounding error over a single time step. 2x

Subsequent time steps

Once this result is established at $s = s^1$ the argument can be repeated for any subsequent time step.

5.7 Numerical Results

We now present numerical results obtained from the FEMPCM described in x5.3{x5.6, which support the arguments made in x5.6.9. The FEMPCM was implemented based on the steps in this chapter.

The initial condition $u^0(x)$ for the method is chosen to be the similarity solution from (2.28),

$$u^S(x;s) = \frac{1}{18l73a0\ -43.339\ Td\ [(57256.3\ Td\ [(000\ Td1[())-".9701\ Tf\ 6.245\ 4.505\ T}$$

5.7.1 Single Time Step

We run the implementation of the FEMPCM for a single time step in order to demonstrate that the method possesses the S Property in the L^2 norm.

We run the method on an initially equally spaced mesh of $N = 21$ nodes. We use the similarity solution (4.20) to determine the initial approximation U^0 at the initial time $s^0 = 1$, such that

$$u^0(x$$

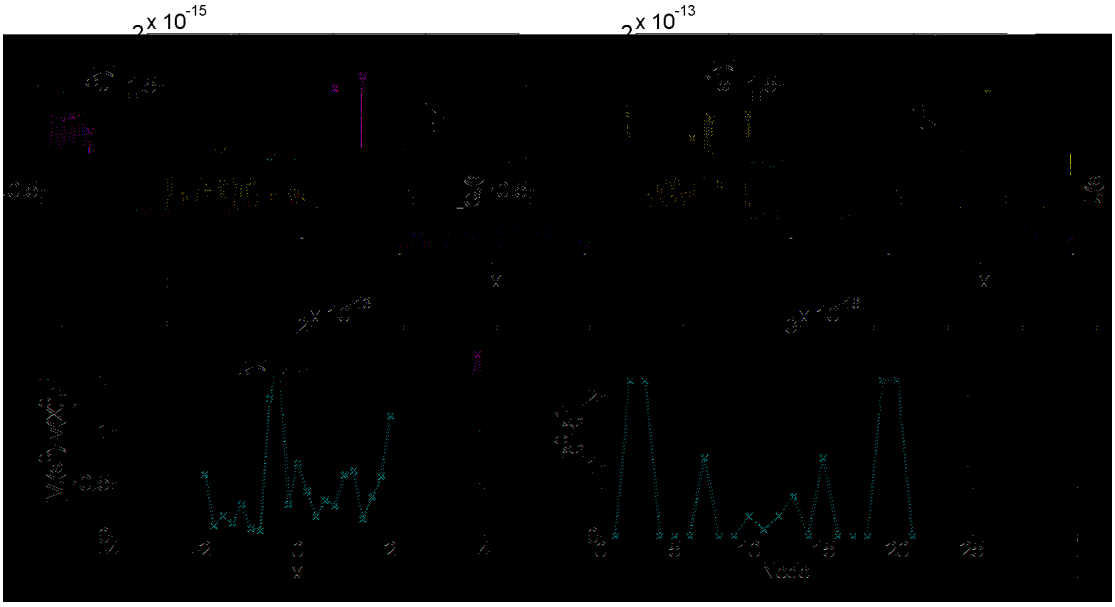

Figure 5.8: Absolute error in the FEMPCM over a single time step. The top left window contains the absolute error in U^1 ($O(10^{-15})$), the top right window the error in Q^0 ($O(10^{-13})$), bottom left the error in V^0 ($O(10^{-13})$) and bottom right the error in the mesh positions ($O(10^{-16})$).

Note that if we were to continue running the FEMPCM over a longer time window we would expect this behaviour to continue. The solution pro le would become shallower as the boundary expands outwards, in keeping with the conservation of mass principle.

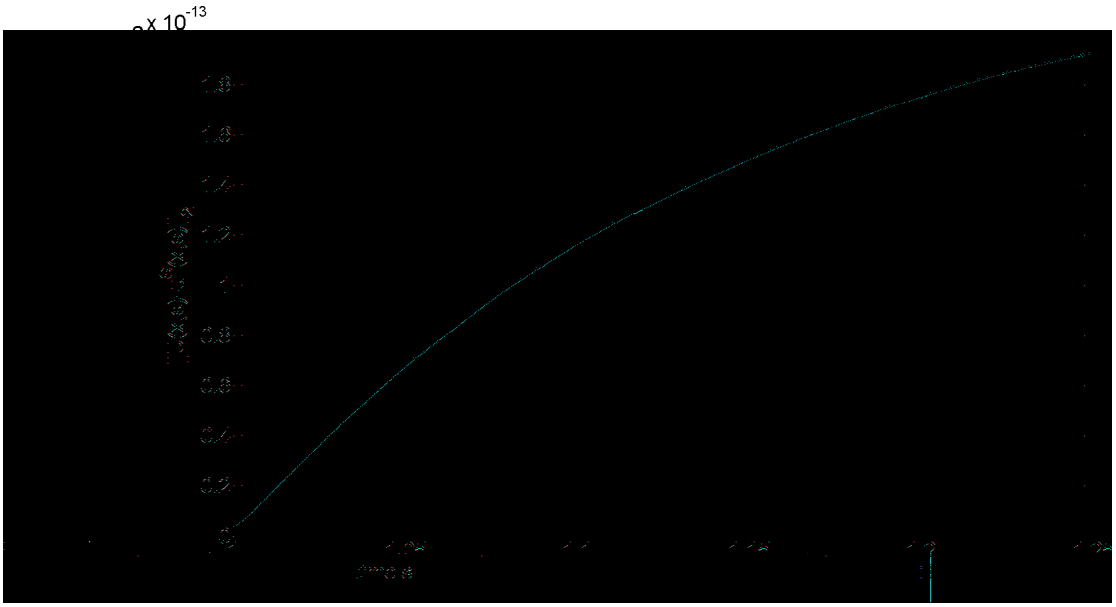

Figure 5.10: L^2 error in the solution for the quartic implementation of the FEMPCM, plotting over the time window $s \in [1; 1.25]$ on a mesh of 21 Nodes. Scale of y-axis on plot is 10^{-13}.

We next observe the error incurred by the method in calculating the solution $U(x; s)$ shown in Figure 5.9. Figure 5.10 shows how the L^2 error in the solution evolves over the

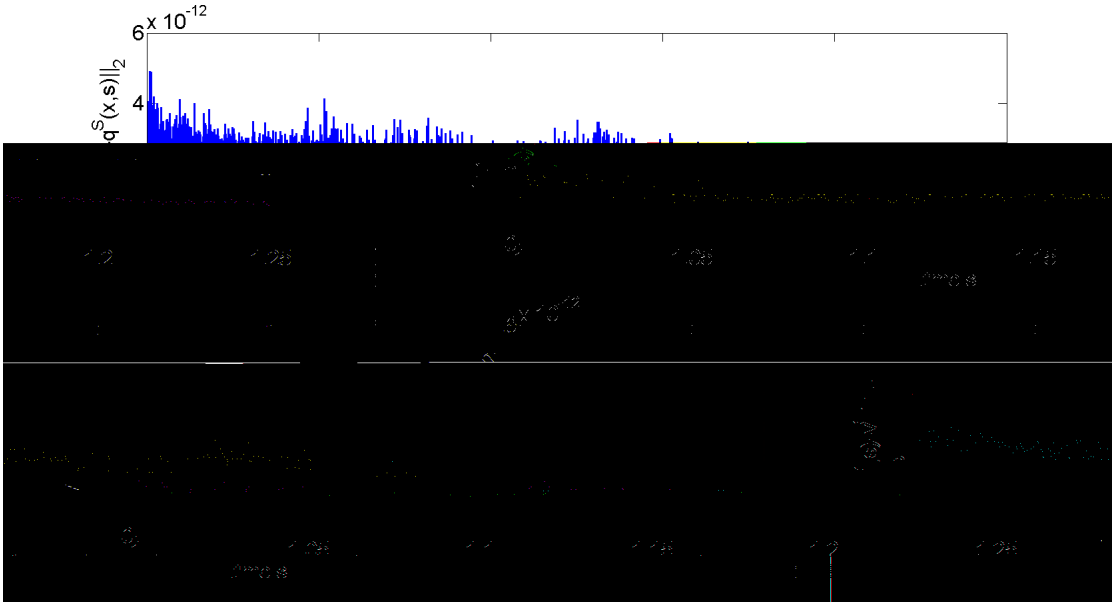

Figure 5.11: L^2 error in $Q(x;s)$ (top plot) and $V(x;s)$ (bottom plot) for the quartic implementation of the FEMPCM, plotting over the time window $s \in [1;1.25]$ on a mesh of 21 Nodes. Scale of y-axis on both plots is 10^{-12}.

approximations is higher than that of $U(x;s)$. Over the course of the time window however, this error does not increase in the same manner as observed in Figure 5.10. We see that the error is approximately 10^{-12} throughout the time window, with the variance decreasing as time progresses. The high oscillation observed in these errors, along with their magnitude suggests that this error is likely numerical and not from the method, which suggests that the S Property is approximately maintained over multiple time steps.

5.8 The FEMPCM in Second/Sixth-Order Problems

In chapter 4 the MPCM was shown to be able to possess the S Property in the l^1 norm, not only for the fourth-order problem (2.3){(2.5) but also for the second (2.7){(2.9) and sixth-order (2.10){(2.12) problems. It is also possible to produce an implementation of the FEMPCM for second-order and sixth-order problems which possess the S Property in the L^2 norm for similarity solutions as for the fourth-order problem (2.3){(2.5).

In order to try and avoid unnecessary repetition in this section, the steps used in the implementation for each problem will be omitted as the majority of the steps are outlined in detail in x5.6. We shall instead outline the di erences between the implementation given in x5.6 and those required for the second or sixth order problems. The main di erences occur in the choice of basis function used in each step of the method and the matrix systems which must be solved to obtain the function approximations. As in x5.6, all references to the time variable shall be made in terms of s rather than t due to the use of a scale-invariant time stepping scheme.

5.8.1 An Implementation of the FEMPCM for the Second-Order Problem for any n

The implementation for the second-order problem (2.7){(2.9) for any n makes use of the piecewise quadratic $\varphi_i(x;s)$ $(i = 1;\ldots;N-1)$ and piecewise linear $\psi_i(x;s)$ $(i = 1;\ldots;N)$ basis functions given in x5.6.2 (written here in terms of s).

The biggest difference between the second and fourth-order implementations is that there is no requirement for the weak form (5.11), since $q = u$ everywhere in the second order case. We therefore do not require the finite dimensional subspaces $S^1(s)$ in this implementation.

We choose our function approximations to lie in the test spaces given by

$$S_E^1(s) = \text{span}\{\phi_j(x;s)\}; \qquad \text{for } j = 2,\ldots,N_r - 1,$$

$$S_e^1(s) = \text{span}\{\phi_j(x;s)\}; \qquad \text{for } j = 1,\ldots,N_r,$$

$$S_h^1(s) = \text{span}\{\psi_j(x;s)\}; \qquad \text{for } j = 1,\ldots,N,$$

and expand $Z(x;s)$, $V(x;s)$ and $U(x;s)$ as the following linear combinations:

$$Z(x;s) = \sum_{j=1}^{N_r} \phi_j(x;s) Z_j(s);$$

$$V(x;s) = \sum_{j=1}^{N} \psi_j(x;s) V_j(s);$$

$$U(x;s) = \sum_{j=2}^{N_r-1} \phi_j(x;s) U_j(s).$$

Obtaining the Velocity Potential $Z(x;s)$

The coefficients $Z_j(s)$ of the velocity potential $Z(x;s)$ are obtained by solving the matrix system

$$K_e(U) Z = K_e(U^n) Q;$$

where $K_e(U)$ is of dimension $N_r - N_r$ and has entries given by (5.40). The matrix $K_e(U^n)$ has entries given by (5.40), with U replaced by U^n. A constraint such as (5.24) is required to uniquely solve the matrix system, as $K_e(U)$ is a singular matrix (see Remark 5.1).

Obtaining the Velocity $V(x;s)$

The coefficients of $Z(x;t)$ can be obtained by solving the matrix system

$$M V = B Z;$$

which is the same system as described in §5.6 for obtaining V from Z.

Time Stepping

Time stepping is performed using the scale-invariant time stepping scheme (5.47),

$$X_i^{k+1} = X_i^k + \frac{1}{\alpha} s^{1/\alpha - 1} s V_i^k, \qquad \text{for } i = 1, \ldots, N,$$

where $\alpha = (n+2)^{-1}$ in the second-order problem (2.7)–(2.9).

Recovery of $U(x,s)$ on the Updated Mesh

Solution recovery for the second-order problem (2.7)–(2.9) involves the construction of modified basis functions $\phi_i(x,s)$ and $\psi_i(x,s)$, both for $i = 1, \ldots, N-2$. These quantities are constructed in the same manner as the ϕ_i and ψ_i described in §5.6.8.

The coefficients $U_j(s)$ for $j = 2, \ldots, N-1$ are then found by solving the matrix system

$$\hat{A} U = \tilde{\gamma},$$

where $\hat{A}$ is a mass matrix of dimension $N-2$ by $N-2$ with time-dependent entries given by

$$\hat{A}_{ij}(s^k) = \int_{a(s)}^{b(s)} \phi_i(x,s)\phi_j(x,s)\,dx, \qquad \text{for } i,j = 1, \ldots, N-2.$$

and $\tilde{\gamma} = \{\gamma_i\}$, for $i = 1, \ldots, N-2$.

Implications

By implementing the method described above, the method can be shown to possess the S Property in the L^2 norm for any value of n.

5.8.2 An Implementation of the FEMPCM for the Sixth-Order Problem with $n = 1$

The process of producing an implementation of the FEMPCM for the sixth-order problem (2.10)–(2.12) is more complex than that of the fourth-order problem (2.3)–(2.5) due

to the addition of the $p(x;t)$ function and the need to use higher order spaces in the method. Many of the steps are similar to those detailed in §5.6, but using different basis functions in the various finite element approximations.

Weak Forms

We shall make use of the weak forms (5.13), (5.12), (5.36), (5.41) and the weighted conservation principle (5.6), all of which we state here for clarity:

$$\int_{a(s)}^{b(s)} w(x;s)p(x;s)\,dx = \int_{a(s)}^{b(s)} \frac{\partial w}{\partial x}\frac{\partial u}{\partial x}\,dx; \qquad \forall w \in W;$$

$$\int_{a(s)}^{b(s)} w(x;s)q(x;s)\,dx = \int_{a(s)}^{b(s)} \frac{\partial w}{\partial x}\frac{\partial p}{\partial x}\,dx; \qquad \forall w \in W;$$

$$\int_{a(s)}^{b(s)} u\left(\frac{\partial w}{\partial x}\frac{\partial z}{\partial x} + \frac{\partial q}{\partial x}\right)dx = 0; \qquad \forall w \in W;$$

$$\int_{a(s)}^{b(s)} w(x;s)\left(v(x;s) - \frac{\partial z}{\partial x}\right)dx = 0; \qquad \forall w \in W;$$

$$\int_{a(s)}^{b(s)} w(x;s)u(x;s)\,dx = \ell(w); \qquad \forall w \in W.$$

Finite Dimensional Subspaces

We next select the finite dimensional subspaces $S_E^1(s), S^1(s), S^{h1}(s), S^1(s), S^{e1}(s) \subset W$. The basis functions for use in this implementation will be the piecewise linear $\phi_i(x;s)$ $(i = 1;\ldots;N)$, piecewise quadratic $\psi_i(x;s)$ $(i = 1;\ldots;N^-)$, piecewise quartic $\theta_i(x;s)$ $(i = 1;\ldots;N^h)$ and we introduce piecewise sextic $\eta_i(x;s)$ $(i = 1;\ldots;N)$ functions for the sixth-order problem (2.10)–(2.12), where we have introduced $N = 6N - 5$, $N^h = 4N - 3$ and $N^- = 2N - 1$ for ease of notation.

We then set

$$S_E^1(s) = \text{span}\{\phi_j(x;s)\}, \qquad \text{for } j = 2,\ldots,N-1,$$

$$S^1(s) = \text{span}\{\phi_j(x;s)\}, \qquad \text{for } j = 1,\ldots,N,$$

$$\hat{S}^1(s) = \text{span}\{\phi_j(x;s)\}, \qquad \text{for } j = 1,\ldots,N,$$

$$S^1(s) = \text{span}\{\phi'_j(x;s)\}, \qquad \text{for } j = 2,\ldots,N-1,$$

$$\hat{S}^1(s) = \text{span}\{\phi_j(x;s)\}, \qquad \text{for } j = 1,\ldots,N.$$

Finite Element Approximations

From the weak forms above, we obtain the following finite element approximations:

For a given $U(x;s) \in S^1(s)$ we seek the finite element approximation $P(x;s)$ in the form: Find $P \in \hat{S}^1(s)$ such that

$$\int_{a(s)}^{b(s)} w(x;s)P(x;s)\,dx = \int_{a(s)}^{b(s)} \frac{\partial w}{\partial x}\frac{\partial U}{\partial x}dx, \qquad \forall w \in \hat{S}^1(s). \tag{5.61}$$

We then seek the finite element approximation $Q(x;s)$ for a given $P(x;s) \in \hat{S}^1(s)$ in the form: Find $Q(x;s) \in S^1(s)$ such that

$$\int_{a(s)}^{b(s)} w(x;s)Q(x;s)\,dx = \int_{a(s)}^{b(s)} \frac{\partial w}{\partial x}\frac{\partial P}{\partial x}dx \qquad \forall w \in S^1(s). \tag{5.62}$$

The finite element approximation $Z(x;s)$ for a given $Q(x;s) \in S^1(s)$, $U(x;s) \in S^1(s)$ is: Find $Z(x;s) \in S^1(s)$ such that

$$\int_{a(s)}^{b(s)} U(x;s)\frac{\partial w}{\partial x}Z(x;s) + \frac{\partial Q}{\partial x}\,dx = 0; \qquad \forall w \in S^1(s).$$

The finite element approximation $V(x;s)$ for a given $Z(x;s) \in S^1(s)$ is of the form: Find $V(x;s) \in \hat{S}^1(s)$ such that

$$\int_{a(s)}^{b(s)} w(x;s)\ V(x;s)\ \frac{\partial Z}{\partial x}\,dx = 0; \qquad \forall w \in \hat{S}^1(s).$$

Once the mesh points have been updated using the scale-invariant time stepping scheme (4.13), we recover $U(x;s)$ on the updated mesh by seeking $U \in S_E^1(s)$ such that

$$\int_{a(s)}^{b(s)} w(x;s)U(x;s)\,dx = \ (w) \qquad \forall w \in S^1(s).$$

Obtaining $P(x;s)$

For a given $U(x;s)$, we obtain the approximation $P(x;s)$. We choose the $w(x;s)$ in (5.61) to be one of the $_i(x;s)$ functions ($i = 1;:::;N$). This leads to a system of N equations

$$\int_{a(s)}^{b(s)} {}_1(x;s)P(x;s)\,dx = \int_{a(s)}^{b(s)} {}_1^0(x;s)U^0(x;s)\,dx; \qquad \text{for } i = 1;$$

$$\int_{a(s)}^{b(s)} {}_i(x;s)P(x;s)\,dx = \int_{a(s)}^{b(s)} {}_i^0(x;s)U^0(x;s)\,dx; \qquad \text{for } i = 2;:::;N \quad 1;$$

$$\int_{a(s)}^{b(s)} {}_N(x;s)P(x;s)\,dx = \int_{a(s)}^{b(s)} {}_N^0(x;s)U^0(x;s)\,dx; \qquad \text{for } i = N:$$

Writing $P(x;s)$ as the linear combination of the $_j(x;s)$,

$$P(x;s) = \sum_{j=1}^{N} {}_j(x;s)P_j(s);$$

and $U(x;s)$ as the linear combination of the $_j(x;s)$,

$$U(x;s) = \sum_{j=2}^{N \quad 1} {}_j(x;s)U_j(s);$$

we obtain a system of equations from which the coe cients $P_j(s)$ can be obtained through solving the matrix system

$$M\,P = K\,U;$$

where M is a tridiagonal matrix of dimension $N \quad N$ and K is a trid5atrix of dimension

system of N equations

$$\int_{a(s)}^{b(s)} \varphi_1(x;s)Q(x;s)\,dx = \int_{a(s)}^{b(s)} \varphi_1^{0}(x;s)P^{0}(x;s)\,dx; \qquad \text{for } i = 1;$$

Updating Mesh Positions

The mesh positions are then updated the new positions X^{k+1} using the scale-invariant time stepping scheme,

$$X_i^{k+1} = X_i^k + 7s^6 \quad sV_i^k; \qquad \text{for } i = 1;:::;N;$$

which di ers from (5.47) since $= 1=7$ in the sixth-order problem (2.10){(2.12) for n = 1.

Recovery of $U(x;s)$

We next recover $U(x;s)$ on the updated mesh positions using the distributed conservation of mass principle.

We introduce modi ed basis functions $\tilde{}_i(x;s)$ and modi ed constants $\tilde{}_i$ (both for $i = 1;:::;N \quad 2$), obtained using the same method as in x5.6.8. The coe cients $U_j(s)$ may then be obtained by solving the matrix system

$$\tilde{A}U = \tilde{};$$

where $\tilde{A}$ is a matrix of dimension $N \quad 2$

Chapter 6

The MPCM in the Fourth-Order Problem with $n > 1$

6.1 Aims of this Chapter

In this chapter we shall examine the fourth-order problem in cases where $n > 1$. In particular we shall examine the initial and small-time behaviour of the boundary of the domain for various initial conditions and choices of n, since there is a wide range of behaviours which can be exhibited.

We shall investigate whether the MPCM as described in Chapters 3 and 4 is able to model these various behaviours and in cases where it fails explore alternatives. The presence of singularities at the boundary of the domain will be shown to have a drastic effect on the success of the MPCM as described in previous chapters. We also explore whether the difficulties relating to the implementation of the MPCM are present in the FEMPCM.

A hybrid numerical method is then proposed, consisting of a fixed-mesh method

of expressing the velocity, in which the fourth-order PDE is not written as two second-order equations, for use in a modi ed version of the MPCM. This alternative velocity avoids the need to approximate the $q(x;t)$ function, as this function is a major cause of the numerical issues in the MPCM.

6.2 The 4th Order Problem with $n > 1$

In this chapter we shall mainly focus on the fourth-order problem with $n > 1$ in (2.3a), written as two second-order equations, which we restate here for clarity. We seek a solution $u(x;t)$ to the 1D fourth-order nonlinear di usion equation with $n > 1$,

$$\begin{aligned} \frac{@u}{@t} &= \frac{@}{@x}\left(u^n \frac{@q}{@x}\right); \\ q &= \frac{@^2u}{@x^2}; \end{aligned} \tag{6.1}$$

for $x \in \Omega_T$, subject to an initial condition at time $t = t^0$ given by

$$u(x;t_0) = u^0(x); \qquad \text{for } x \in \Omega(t^0);$$

and boundary conditions

$$\begin{aligned} u &= 0; \\ \frac{@u}{@x} &= 0; \\ uv + u^n \frac{@q}{@x} &= 0; \end{aligned}$$

at the moving boundaries $x = a(t)$ and $x = b(t)$, for $t > t^0$.

As we are investigating the fourth-order problem with $n > 1$ and with no explicit similarity solutions available, we shall focus on the qualitative behaviour of the approximate solutions generated by the MPCM. In chapter 3 we observed that in the case where $n = 1$ and the initial condition corresponds to a similarity solution, the velocity $v(x;t)$ is a linear function in x given by

$$v(x;t) = \frac{x}{t};$$

6.4 Initial Boundary Velocities

In this section we shall focus on the initial time $t = t^0$ and examine the initial boundary velocity for $n > 1$. We consider a particular family of initial conditions and determine under what circumstances the limit (6.3) exists.

We shall consider initial conditions of the form

$$u^0(x) = c_1(!^2 \quad x^2) \ ; \tag{6.4}$$

for some constant > 0, where c_1 and $!$ are arbitrary constants. In this case the initial boundary positions are located at the points $x = \ !$, so $a(t^0) = \ !$ and $b(t^0) = !$, giving (together with the symmetry of the PDE (6.1)) symmetric solutions about the point $x = 0$.

We wish to consider the di erent possible asymptotic behaviours of the moving boundaries for di erent choices of $$ in (6.4) and n in (6.1), as outlined in King (2001) and Blowey et al. (2007). From this point on we shall focus our discussion on the asymptotic behaviour at the right-hand boundary at $x = !$, although the same arguments apply for the left-hand boundary with a little adjustment.

For $x \ !$ we have, from (6.4),

$$u^0(x) \quad c_1(2!\ )\ (! \quad x) \ : \tag{6.5}$$

Using (6.4) and (6.2) we can calculate the initial velocity $v^0(x)$ to be

$$v^0(x) = 12\,c_1^n\ (\quad 1)x(!^2 \quad x^2)^{n \quad 2} \quad 8c_1^n\ (\quad 1)(\quad 2)x^3(!^2 \quad x^2)^{n \quad 3};$$

$$c_2\ (\quad 1)(! \quad x)^{n \quad 2} \quad c_3\ (\quad 1)(\text{rf 9c[(u)]52r 10.9091 Tf 10.909 0 Td [(1)()]TJ/F25 10.909}$$

6.4.1 n = 3

If n and are such that n = 3, then the limit on the right-hand side of (6.8) is nite, with the limiting velocity

$$v_b^0 = \quad c_3 \; (\quad 1)(\quad 2):$$

Hence the limiting boundary velocity (using c_3 from (6.7) at x = !) is

$$v_b^0 = \quad c_1^n (2! \;)^n \quad (\quad 1)(\quad 2); \qquad (6.9)$$

which is non-zero if $\neq$ 1 ; 2. The direction of motion of the moving boundary is then dependent on the sign of c_1 and the value of . For example, if $c_1 > 0$ in (6.4) then from (6.9) we have a positive velocity for 1 < < 2 and a negative velocity for < 1, > 2.

It is worth noting that the values = 1 and = 2 correspond to points at which the nite velocity (6.9) changes sign from positive to negative or vice versa.

6.4.2 n > 3

In the case where n and are such that n > 3, the limit on the right-hand side of (6.8) (and therefore v_b) is zero as x ! ! . In this case the boundary is initially stationary.

6.4.3 n < 3

In this case the limit on the right-hand side of (6.8) does not exist. In this instance the boundary will move with an initially unbounded velocity. It is known however, that the velocity is only unbounded instantaneously at the initial time, after which the nite speed of propagation property possessed by the PDE results in a nite velocity (see Bernis (1996b) and Bernis (1996a)).

This result is supported by asymptotic results presented in

6.5.2 Evolution of the Solution Close to the Boundary

Once the initial behaviour has occurred, the local behaviour of the solution for $t > t^0$ is dependent upon the value of n and is discussed in King (2001). When $n \geq 3$ the local behaviour of the solution as $x \to \beta$ (see also (2.6)) takes the form

$$u \sim \begin{cases} \dfrac{n^3 \dot{b}}{3(3-n)(2n-3)}(b(t)-x)^3 \Big)^{\frac{1}{n}}; & \text{for } \dfrac{3}{2} < n < 3; \\ \dfrac{3}{4}\dot{b}(b(t)-x)^3 \ln\left(\dfrac{1}{(b(t)-x)}\right)^{\frac{2}{3}}; & \text{for } n = \dfrac{3}{2}; \\ B(t)(b(t)-x)^2; & \text{for } n < \dfrac{3}{2}; \end{cases} \tag{6.10}$$

where $\dot{b}$ denotes the velocity of the boundary $b(t)$ and $B(t)$ must be determined as part of the solution. Here $b(t)$ is the position of the moving boundary, so at time $t = t^0$ we have $b(t^0) = \beta$.

If $n > 3$ the solution only takes the form (6.10) once any waiting-time has ceased. In some cases the waiting-time ceases due to a shock forming as the gradient of the solution becomes unbounded close to the boundary. The solution may also decrease such that

$$u \sim \left(\frac{n^3(\beta-x)^4}{8(4-n)(2-n)(n+4)\,t}\right)^{1/n}; \qquad \text{as } x \to \beta;\ t^0 < t < t^w;$$

experiencing various types of decay based upon n and α (see Blowey et al. (2007) for more details).

In the regime $3 < n < 4$, with $n > 3/2$ the boundary velocity tends to zero as $t \to t^0$, with the boundary waiting. We anticipate that between t^0 and the end of the waiting-time, α decreases from the initial value (greater than $3-n$) until such time as it becomes equal to $3-n$. At this point the solution will take the form (6.10) and the boundary begins to move.

6.6 Initial Boundary Velocities using the MPCM

From the results in §6.4 we expect the MPCM to be able to track the boundaries of the domain only in cases where $n \geq 3$, since the velocity in these cases is finite or zero.

The case where the boundary velocity is initially unbounded needs special consideration (and is therefore not considered further in this section).

6.6.1 Numerical Verification of Boundary Velocity Issues

We shall now perform experiments using the implementation of the MPCM from Chapter 4 to verify that the issues described in the previous section will occur.

We shall calculate the initial velocity V^0 using (3.24) (see §3.3.4 for more details) using the MPCM described in Chapter 4 on meshes of an increasing number of nodes. We use the initial condition

$$u^0(x) = \frac{1}{10}\left(1 - x^2\right); \tag{6.11}$$

for various $$ in order to estimate the correct boundary velocities for the finite advance, finite retreat and waiting-time cases described in §6.4.

We perform experiments on meshes of $N = 21, 41, 81, \ldots, 1281$ nodes, using the following choices of n in (6.1) and $$ in (6.11):

Finite Advance

For the finite advance case we choose $n = 8/5$, $ = 15/8$, giving $n + = 3$ with $1 < < 2$. In this case we should observe a positive boundary velocity, corresponding to a finite advance of the boundary. The exact value of the initial boundary velocity in this case is equal to

$$v_b^0 = \frac{105}{64}\left(\frac{1}{10}\right)$$

.28 Td [(b)]TJ/F15 v .10.90- 7.3[(F)83(8F26 7.9701 Tf -0.394

N	n = 8=5, = 15=8	n = 2, = 3=2
21	0.0994	-0.0025
41	0.0930	-0.0036
81	0.0971	-0.0039
161	0.0999	-0.0041
321	0.1015	-0.0042
641	0.1023	-0.0042
1281	0.1027	-0.0042
v_b^0	0:04	0.03

N	$n = 5/2$, $= 5/2$	$n = 2$, $= 2$
21	-0.0076	-0.0076
41	-0.0012	-0.0012
81	$1.63 \quad 10^{-4}$	$1.63 \quad 10^{-4}$
161	$2.15 \quad 10^{-5}$	$2.15 \quad 10^{-5}$
321	$2.75 \quad 10^{-6}$	$2.75 \quad 10^{-6}$
641	$3.48 \quad 10^{-7}$	$3.47 \quad 10^{-7}$
1281	$4.36 \quad 10^{-8}$	$4.37 \quad 10^{-8}$
v_b^0	0	0

Table 6.3: Initial boundary velocities for the waiting-time experiments in §6.6.1. The expected boundary velocity is given in the final row for each case.

cases we have no conclusive evidence of convergence and we cannot be confident that the method is producing satisfactory results.

The finite retreat cases (Table 6.2) do have the correct sign, but there appears little evidence that the velocity is converging towards the expected value at an acceptable rate as N increases. In both cases the calculated boundary velocity is larger than the expected value for all values of N.

Of the three cases investigated the waiting-time case (Table 6.3) is most encouraging, as the computed velocity does appear to be decreasing towards zero as N increases. Since the MPCM is a numerical method we would not expect the velocity to be exactly zero, but a decrease of one order of magnitude as N is doubled suggests that it will eventually reach rounding error. The values shown in Table 6.3 are much higher than rounding error however, so the number of nodes required for the velocity to approach rounding error is likely to be prohibitively high.

In an attempt to understand why the initial velocities are not as expected, we investigate the computed boundary velocity for one of the experiments. Choosing the $n = 6/5$, $= 5/2$ finite retreat case, we plot the computed initial velocity for the final 20 mesh points of the mesh. This computed velocity is given in Figure 6.1.

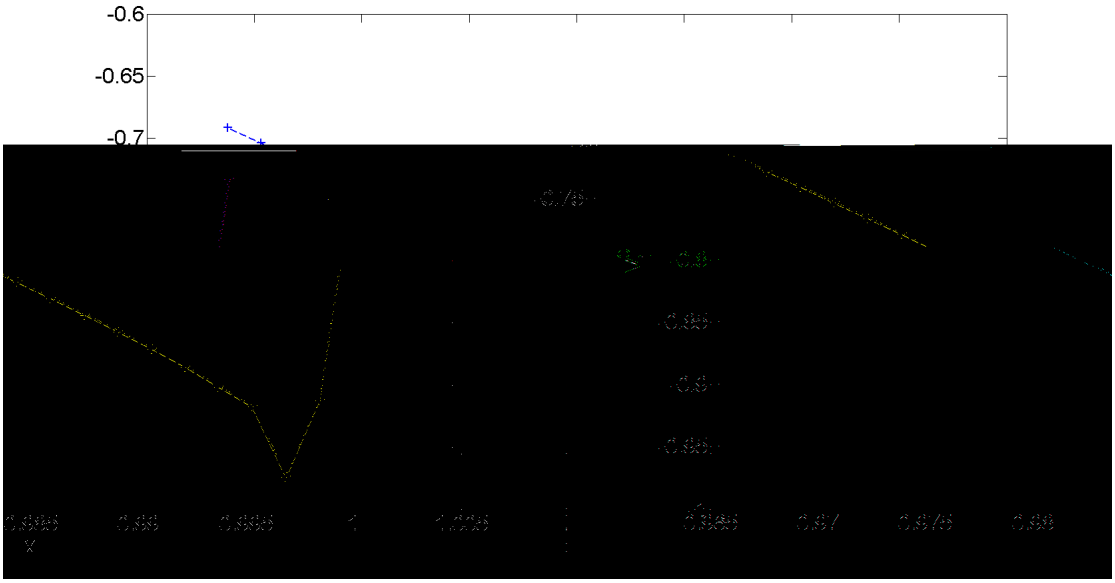

Figure 6.1: The computed velocity V_i^0 plotted over the final 20 mesh points for the $n = 6=5$, $= 5=2$ finite retreat case.

We see that oscillations have appeared at points close to the boundary, which then cause the quadratic extrapolation (used in the evaluation of the boundary velocity) to be performed inaccurately. Examining the velocity for the other cases (not shown), we also observe the appearance of oscillations close to the boundary which cause the computed boundary velocity to be different than the expected value.

In an attempt to explain these inaccurate boundary velocities, we must examine more than the product of terms in (6.8). The restriction on the values of n and $$ is not the only consideration that must be made when considering the limit in (6.8). The function $q(x;t)$ in (6.1) and its derivative $\frac{@q}{@x}$ may also be problematic, which we shall examine next.

6.6.2 Initial Boundary Values of $q(x;t)$

The implementation of the MPCM described in Chapters 3 and 4 involves an approximation of the $q(x;t)$ function as well as the velocity $v(x;t)$ (which from (6.2) consists of

a pointwise multiplication of $u^{n\ 1}$ and $^{@q}$

From (6.4), we can calculate the initial approximation to $\frac{@q}{@x}$ asymptotically at points close to the boundary. Using (6.4),

$$\frac{@q}{@x} = 8c_1 \quad (\quad 1)(\quad 2)x^3(! \ ^2 \quad x^2) \ ^3 \quad 12c_1 \quad (\quad 1)x(! \ ^2 \quad x^2) \ ^2;$$

which, for $x \quad !$ is given by

$$\frac{@q}{@x} \quad 2 \ c_1 \quad (\quad 1)(\quad 2)! \quad (! \quad x) \ ^3 \quad 3(2 \ )c_1 \quad (\quad 1)! \ ^1(! \quad x) \ ^2: \quad (6.13)$$

It is clear from (6.13) that for $< \ 3$, $\frac{@q}{@x}$ becomes unbounded as $x \ ! \ !$. We cannot therefore expect the MPCM to be able to accurately approximate $\frac{@q}{@x}$ at the boundary points using extrapolation (or indeed any numerical method) for values of $< \ 3$.

This information provides further insight into the possible boundary behaviours from (6.8). For $< \ 3$ we have that $\frac{@q}{@x}$ is unbounded at the boundary of the domain and hence the limit (6.3) (if it exists) will involve a multiplication of the form \0 1 " for values of $n > 1$ (and may therefore be indet4 4.504 rminat.504).

weak forms and resulting finite element approximations are integral forms and it can be argued that the FEMPCM will be able to calculate the velocity at points close to the boundary without the issues previously experienced due to the singularity present at the boundary.

In this section we explore the initial behaviour and convergence of the piecewise linear implementation of the FEMPCM, described in §5.5 (which includes the calculation of the approximation $Q(x;t)$). We shall use this implementation, as opposed to the higher order implementation which possesses the $\mathcal{S}$ Property described in §5.6, as the higher order implementation is only appropriate for the $n = 1$ case.

We conduct numerical experiments in which we hope to observe the various behaviours of the boundaries (finite advance, finite retreat and waiting-time) outlined in §6.4 for values of n and α such that $n \geq 3$. It is hoped that since the functions are approximated throughout the domain (using L^2 projections) rather than just point-wise at the mesh points, the velocity will be non-oscillatory close to the boundary and therefore produce convergent boundary trajectories.

The implementation used involves the calculation of $Q(x;t)$, which we have shown to be problematic in the implementation of the MPCM for point-wise approximations. It is hoped that the integral forms of the finite element implementation will alleviate the issues associated with $Q(x;t)$ and $V(x;t)$ close to the boundary.

6.7.1 Finite Advance

We set $n = 2$ in (6.1), and use the initial condition

$$u^0(x) = \frac{1}{10}(1 - x^2)^{3/2};$$

in the finite element implementation, giving a value of $n = 3$ in which $n = 2$, $\alpha = 3/2$ so that $1 < \alpha < 2$. We therefore expect the right-hand boundary of the domain to move outwards with a finite velocity. In this experiment a time window of $t \in [1; 1.25]$ is used in order to provide information on the behaviour of the boundary for times beyond the initial time.

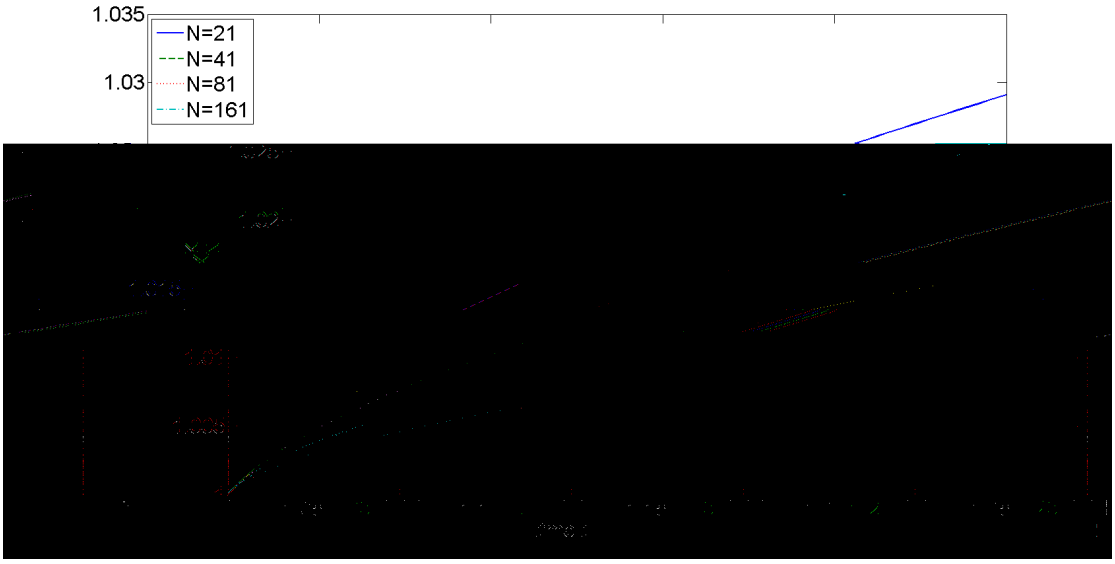

Figure 6.2: Boundary trajectories for the nite advance experiment. Shown here are trajectories for meshes of $N = 21$ (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes.

Figure 6.2 details the boundary trajectories for the nite advance experiment. We see that the boundary has moved further for the $N = 21$ case than all others, with the maximum distance moved by the boundary decreasing each time N is increased. There is no conclusive evidence of convergence of the boundary trajectories.

We next examine the nodal values of the solution at the end of the time window on mesh points which we would expect to coincide over all of the meshes used. We test values at the central point of the mesh $x = 0$, the right-hand boundary X_N and an `Interior Point' lying between the central point and the right hand boundary.

If we use the $N = 161$ solution as a reference solution (U_{ref}, say), we can determine the absolute di erence between the solution values and mesh locations for the other meshes. The absolute di erences are given in Table 6.4, with the solution values at the boundary omitted due to the zero boundary conditions making such calculations redundant. We also omit the di erence in mesh position at the central point as the symmetric initial conditions cause the central point to remain xed at $x = 0$ throughout

the time window.

	Central Point		Interior Point				Boundary Point
N	$\lvert U_{ref} \rvert$	$\lvert U \rvert$	$\lvert U_{ref} \rvert$	$\lvert U \rvert$	$\lvert X_{ref} \rvert$	$\lvert X \rvert$	$\lvert X_{ref} \rvert$ $\lvert X \rvert$
21	3:49	10^{-5}	3:59	10^{-6}	2:07	10^{-4}	0.0158
41	3:55	10^{-6}	1:05	10^{-5}	6:23	10^{-5}	0.0086
81	8:09	10^{-8}	4:22	10^{-6}	1:46	10^{-5}	0.0034

Table 6.4: Absolute difference of mesh points and solution values for the finite advance experiment at coinciding points

Table 6.4 shows some evidence of slow convergence between the solution values of various meshes. The rate at which this convergence is being achieved is not very quick however, indicating a possible need to increase the number of mesh points further. Some evidence of convergence has been observed for the interior and boundary mesh points, but the results are unconvincing.

6.7.2 Finite Retreat

In the finite retreat experiment we set $n = 1:2$ in (6.1) and use the initial condition

$$u^0(x) = \frac{1}{10}\left(1 - x^2\right)^{5=2};$$

in the finite element implementation. For these choices of n and we therefore expect an initially retreating right-hand boundary with a finite velocity. We run the method over the time window $t \in [1; 1:025]$.

Figure 6.3 points,

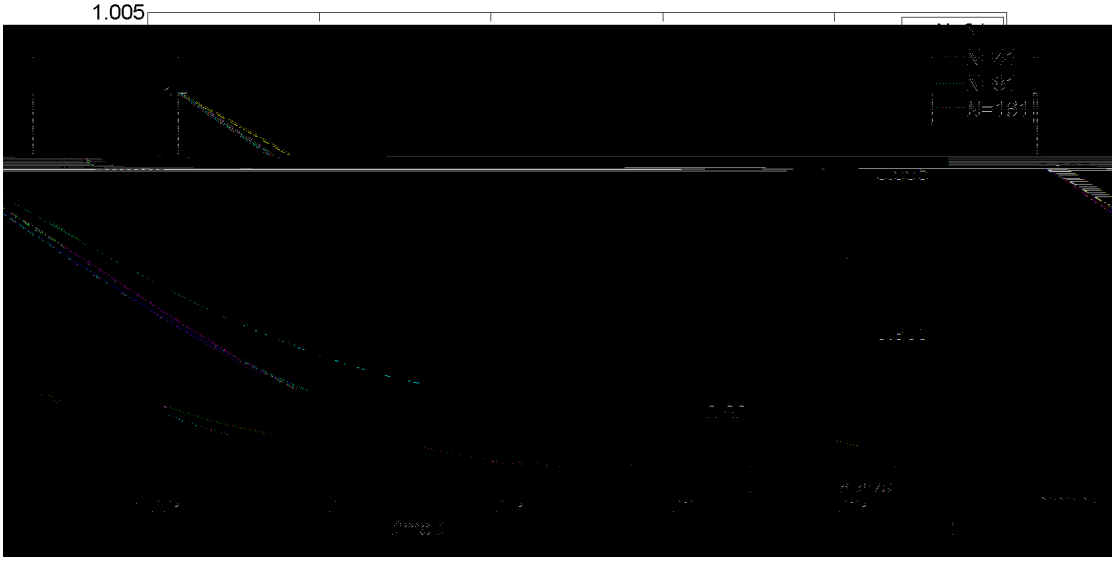

Figure 6.3: Boundary trajectories for the nite retreat experiment. Shown here are trajectories for meshes ofN = 21 (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes.

mesh further, we might expect the trajectories to become closer together. As discussed in the nite di erence case however, the rate of convergence is not quick enough to propose that a reference trajectory has been reached.

If we use theN = 161 solution as a reference solution (ψ_{ref} , say), we can determine the absolute di erence between the solution values and mesh locations for the other meshes. The absolute di erences are given in Table 6.5.

Table 6.5 shows that there is some evidence of convergence between the various meshes. The rate of convergence is slow however, indicating a possible need to increase the number of mesh points further.

6.7.3 Waiting-Time

The waiting-time experiment uses value ofn = 2 and = 2, giving an initial condition

$$u^0(x) = \frac{1}{10}(1 \quad x^2)^2$$

	Central Point		Interior Point				Boundary Point	
N	$\lvert U_{ref} - U \rvert$		$\lvert U_{ref} - U \rvert$		$\lvert X_{ref} - X \rvert$		$\lvert X_{ref} - X \rvert$	
21	8:70	10^{-6}	8:59	10^{-6}	1:99	10^{-4}	0.0043	
41	2:62	10^{-6}	1:85	10^{-6}	5:06	10^{-5}	0.0013	
81	5:45	10^{-7}	3:85	10^{-7}	9:89	10^{-6}	2:76	10^{-4}

Table 6.5: Absolute difference of mesh points and solution values for the finite retreat experiment at coinciding points

and an initial boundary velocity of zero. The time window used in this experiment is $t \in [1; 1{:}025]$, as in the finite retreat case.

Figure 6.4 shows the boundary trajectories for the various meshes, which we would expect to remain within rounding error of $x = 1$ if the boundary is experiencing waiting-time behaviour. We see that the $N = 21$ trajectory has moved outwards more than

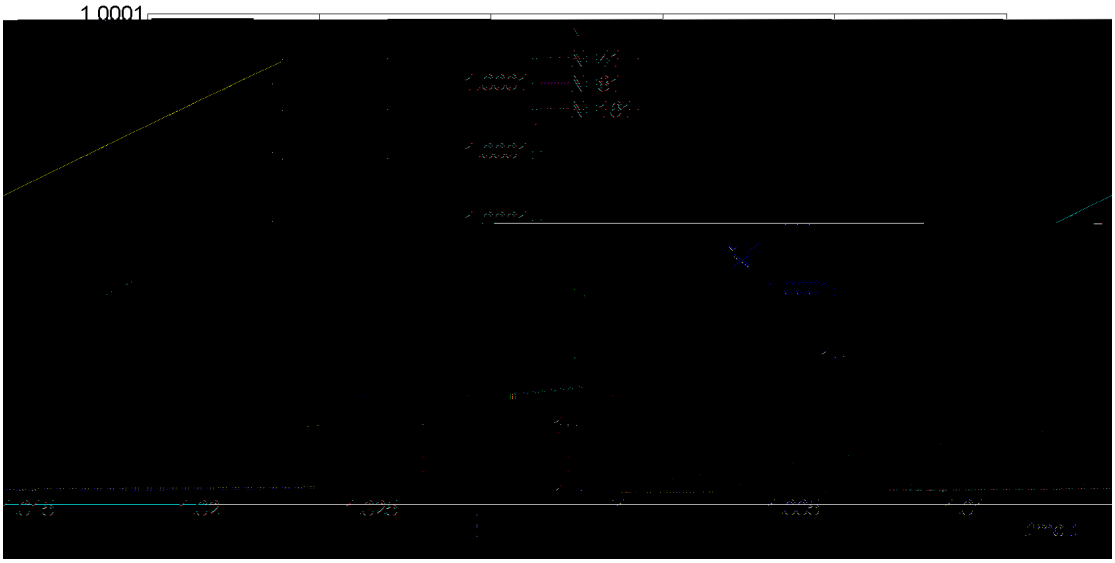

Figure 6.4: Boundary trajectories for the waiting-time experiment. Shown here are trajectories for meshes ofN = 21 (blue solid line), 41 (green dashed line), 81 (red dotted line) and 161 (cyan dash-dotted line) nodes.

	Central Point		Interior Point				Boundary Point	
N	jU_{ref}	Uj	jU_{ref}	Uj	jX_{ref}	Xj	jX_{ref}	Xj
21	5:03	10^{7}	7:22	10^{6}	3:21	10^{5}	1:25	10^{4}
41	7:83	10^{8}	1:64	10^{6}	7:92	10^{5}	2:95	10^{5}
81	1:38	10^{8}	3:29	10^{7}	1:59	10^{6}	5:87	10^{6}

Table 6.6: Absolute di erence of mesh points and solution values for the waiting-time experiment at coinciding points

x6.7.1{x6.7.3 for values ofN = 21 to N = 1281, with Remark 6.1 about v_b^0 from x6.6.1 valid here.

N	Advance	Retreat	Wait
21	0.2597	-1.1248	0.0054
41	0.2629	-1.3433	0.0014
81	0.2643	-1.4534	3:39 10^{-4}
161	0.2650	-1.5085	8:46 10^{-5}
321	0.2653	-1.5362	2:12 10^{-5}
641	0.2655	-1.5500	5:29 10^{-6}
1281	0.2655	-1.5569	1:33 10^{-6}
v_b^0	0.03	0:95	0

Table 6.7: Initial Boundary Velocities for the experiments in x6.7.1{6.7.3. The expected

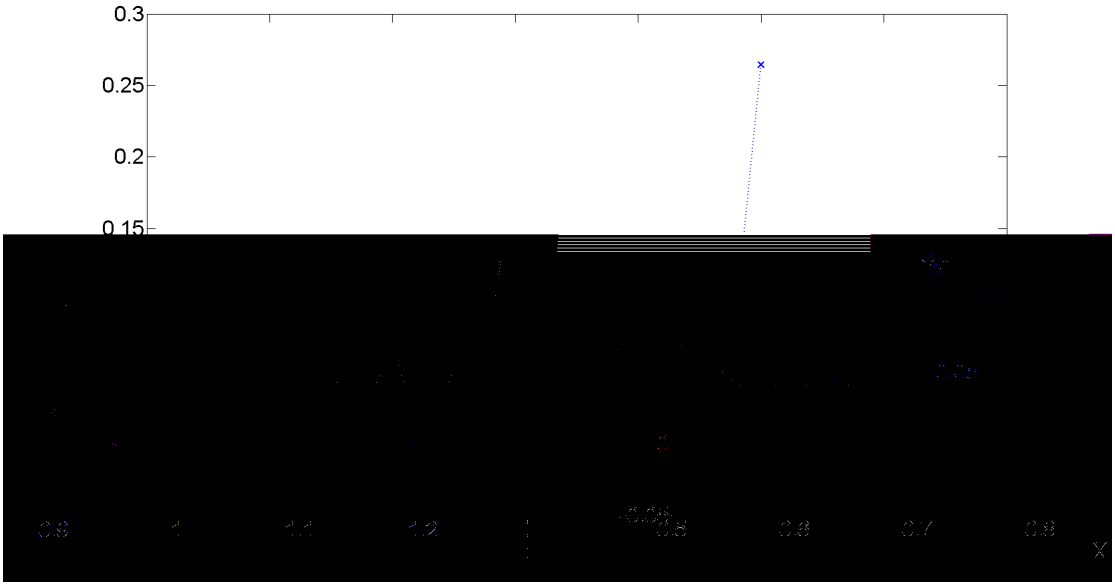

Figure 6.5: Computed velocity in the first time step for the finite advance experiment with $N = 81$, at points near the right-hand boundary.

close to the boundary, which has caused the boundary velocity to be larger than the expected value.

6.7.5 Critique of the FEMPCM with $n > 1$

Since the oscillations are observable in both the MPCM (see §6.6.1) and the FEMPCM (§6.7) , the issue with convergence that we are observing at the boundaries of the domain is not restricted to the implementation being used.

The need to calculate an approximation to the $q(x;t)$ function and its first derivative, even in weak form, appears to be the main cause of issues when considering the implementation of the FEMPCM (as discussed in §6.6.2–§6.6.3 for the MPCM). When considering initial conditions of the form (6.4), the range of α for which either $q(x;t)$ or $\frac{\partial q}{\partial x}$ becomes unbounded at the boundary is restrictively large.

Despite the integral forms present in the FEMPCM, allowing for a weak form representation present throughout the whole domain (as opposed to point-wise values in

the nite di erence implementation), the implementation appears to be unable to adequately model the velocities expected for the various values of tested in this section.

6.8 Mesh Point Distribution

In the discussion of the numerical results for the MPCM and FEMPCM presented in this thesis, the distribution of the mesh points has thus far been overlooked. In this section we shall explore the initial distribution of the mesh points and their evolution at later times, with a view to how the MPCM and FEMPCM could bene t from careful selection of the initial mesh.

6.8.1 Uniform Initial Mesh

All the numerical results presented in this thesis so far have been generated using a mesh which was chosen to be initially equally spaced. In the cases where the MPCM or FEMPCM possesses the Property (i.e. the results given in Chapters 4 and 5), this uniform initial mesh remains uniform over time due to the linear velocity present in these cases (see equations (3.14) and (3.31) for = 1).

As an example, Figure 6.6 illustrates the mesh evolution for the multiple time step results given in x4.3.6. In this case the initial mesh spacing was taken to be $X = 0:2$, which has increased to $X = 0:5$ at time $s = 2:5$.

For the examples presented in this chapter, an initially uniform mesh (driven by a velocity which is not necessarily linear in terms of) may not remain uniform. A uniform mesh may also not provide adequate resolution at points of interest (such as near the boundary for the types of problems studied in this chapter).

To demonstrate this we examine the cases explored in x6.6.1, for the various choices of and used. In the results presented only the initial boundary velocity was discussed. Here we run the MPCM for the various ; values over 10,000 time steps and plot the evolution of the nal 11 mesh points from a mesh of $N = 81$ points. These plots are given in Figure 6.7.

We see that the most interesting behaviour is taking place in the nite retreat cases ($n = 6=5$; $= 5=2$ and $n = 15=11$; $= 11=5$), with the points closest to the boundary being of particular interest. Here the mesh spacing decreases as the boundary retreats,

interfaces.

6.9.1 The Fixed-Mesh Method

The use of a fixed-mesh method requires us to recast the original fourth-order problem. We shall seek a solution $u(x;t)$ to the fourth-order PDE

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u^n \frac{\partial q}{\partial x}\right); \qquad \text{in} \qquad (t^0;T)$$

$$q = \frac{\partial^2 u}{\partial x^2}; \qquad \text{in} \qquad (t^0;T)$$

for $x \in \Omega$, where Ω is a finite-space interval $(X_a;X_b)$ and no longer time-dependent.

We seek solutions subject to an initial condition

$$u(x;t_0) = u^0(x); \qquad \text{at } t = t^0;$$

and boundary conditions

$$\frac{\partial u}{\partial x} = 0;$$

$$uv + u^n \frac{\partial q}{\partial x} = 0;$$

at the moving interfaces $x = a(t)$ and $x = b(t)$, for $t > t^0$.

Compared with the MPCM, the fixed-mesh part of the hybrid numerical method operates on a static mesh of points, with a fixed distance Δx between points. Since the mesh is now fixed, the domain Ω will include points outside of the support of $u(x;t)$, to allow the interfaces of the support to potentially expand over time. For all points in Ω outside of the interfaces at $x = a(t)$, $x = b(t)$ we have that $u = 0$. We shall choose the two points X_a, X_b far enough away from the support of $u(x;t)$ that the interfaces will not reach these points by the end of the time window.

By including mesh points outside of the support of the solution we are able to use a numerical scheme in the approximation of $u(x;t)$ which has an effect on the movement of the free boundary over a given time-step. This feature of the numerical scheme will allow the hybrid method to be able to model the case where the boundary moves instantaneously with an unbounded velocity, which was not previously possible in the MPCM.

Given a discretisation of the domain $x \in [X_a; X_b] \subset \mathbb{R}$ in the form of a mesh of N nodes, we can approximate the function $u(x;t)$ by nodal values U_i $(i = 1; \ldots; N)$ using a finite difference scheme.

Due to the presence of the u^n factor in (6.1) we choose the semi-implicit finite difference scheme

$$U_i^{k+1} = U_i^k + \frac{\Delta t}{\Delta x^2} \left(U_{i+1/2}^k\right)^n \left(\frac{\left(-U_{i+2}^{k+1} + 2U_{i+1}^{k+1} - U_i^{k+1}\right)}{\Delta x^2} - \frac{\left(-U_{i+1}^{k+1} + 2U_i^{k+1} - U_{i-1}^{k+1}\right)}{\Delta x^2} \right) + \left(U_{i-1/2}^k\right)^n \left(\frac{\left(-U_{i+1}^{k+1} + 2U_i^{k+1} - U_{i-1}^{k+1}\right)}{\Delta x^2} - \frac{\left(-U_i^{k+1} + 2U_{i-1}^{k+1} - U_{i-2}^{k+1}\right)}{\Delta x^2} \right)!; \tag{6.14}$$

for $i = 2; \ldots; N - 2$.

The values $U_{i+1/2}^k$ and $U_{i-1/2}^k$ in (6.14) are obtained by the midpoint approximations

$$U_{i+1/2}^k = \frac{U_{i+1}^k + U_i^k}{2}; \qquad U_{i-1/2}^k = \frac{U_i^k + U_{i-1}^k}{2}; \tag{6.15}$$

for $i = 1; \ldots; N - 1$.

We enforce boundary conditions by setting $U_0^k = U_1^k = U_{N-1}^k = U_N^k = 0$ for all values of k, as we have chosen the boundaries of the domain such that the numerical free boundary does not reach the boundary of over the course of the time window. Hence we do not expect the solution at the nodes outside the support to take any value other than zero.

We can rearrange (6.14) and the boundary conditions in order to obtain a linear system of N equations of the form

$$A U^{k+1} = F^k;$$

for the solution vector $U^{k+1} = \{U_i^{k+1}\}$, where A is a $N \times N$ pentadiagonal matrix and F^k is a vector of length N.

The first two rows and last two rows of A have non-zero entries only on the main diagonal, with entries $A_{0;0} = A_{1;1} = A_{N-1;N-1} = A_{N;N} = 1$. The remaining rows have

the following form:

$$
\begin{aligned}
A_{i;i\ \ 2} &= \ \ [(U_i^k{}_{\ 1=2})^n]; \\
A_{i;i\ \ 1} &= \ \ \ [(U_{i+1=2}^k)^n + 3(U_i^k{}_{\ 1=2})^n]; \\
A_{i;i} &= 1 + 3\ \ [(U_i^k{}_{\ 1=2})^n + (U_{i+1=2}^k)^n]; \\
A_{i;i+1} &= \ \ \ [3(U_{i+1=2}^k)^n + (U_i^k{}_{\ 1=2})^n]; \\
A_{i;i+2} &= \ \ [(U_{i+1=2}^k)^n];
\end{aligned}
$$

where $= \ t=(\ x)^4$, for $i = 2;:::;N \ \ 2$. The vector F^k has entries

$$F^k = (0;0;U_2^k;U_3^k;:::;U_{N\ \ 3}^k;U_{N\ \ 2}^k;0;0)^T:$$

6.9.2 De ning the Numerical Interface

Since the approximate solution U_i^k is only de ned at discrete points and the position of the moving interface (at which $u(x;t) = 0$) may lie between two mesh points we need a method of de ning an approximation to the position of the moving interface at the end of the time window $t\ 2\ [t^0;T^0]$, since the position of the interfaces will need to be passed to the MPCM.

From (6.14) and (6.15), we can see that for points outside of the support of $u(x;t)$ we have

$$U_{i\ \ 1}^k = U_i^k = U_{i+1}^k = 0\ \ )\ \ U_i^{k+1} = 0; \qquad (6.17)$$

and as a result the free boundary can advance by at most one mesh point over a single time step. A consequence of (6.17) is that the time step size t must be chosen such that $\ t = O(\ x^2)$ (Barrett et al. (1998)).

Given this feature of the numerical scheme, it is possible to introduce a means of determining the position of the numerical boundary. Since the solution is expected to remain nonnegative if the initial condition is nonnegative (Bernis and Friedman (1990)), we may de ne the position of the numerical interface to coincide with the rst mesh point at which the solution is zero outside of the support of the approximate solution. This approach is awed if oscillations appear in the solutions as a result of the numerical

scheme however, which makes determining such a point extremely problematic. A non-negativity preserving scheme (such as that found in Blowey et al. (2007)) should ensure that the interface can be obtained in this manner, but this has not been included in the implementation presented here.

6.9.4 Use of the MPCM

Once the solution obtained by the fixed-mesh method described in §6.9.1 has been shown to converge, the mesh is reduced to only mesh points containing the support of U, with at most one zero-valued mesh point at each interface.

From the fixed mesh solution, we select a representation of mesh points throughout the support of U, using the selected points as the initial approximation U^0 in the MPCM. The MPCM is much more computationally expensive to run when the number of nodes in the mesh is high, which is likely to be the case when the fixed-mesh method has converged, hence the need to reduce the number of points used in this part of the hybrid method.

This solution is then used as the initial condition in the MPCM, at time $t = T^0$. The MPCM method is then run over the time window $t \in [T^0; T]$, where T is the final time the method is run to, as described in Chapters 3 or 4.

6.9.5 Critique of the Hybrid Method

The hybrid method described in this section has the potential to be able to model the case where the initial velocity of the boundary of the domain is instantaneously unbounded, as well as modelling the other expected behaviours discussed in §6.4. However, there are a number of factors to consider in order to enable the method to work correctly, which have so far prevented the method being implemented satisfactorily:

The value of T^0 is important, since we are interested in using the MPCM part of

In the xed-mesh portion of the hybrid method there is a further potential issue which may arise if the moving interface lies on or close to a mesh point, in cases where $q(x;t)$ or $\frac{@q}{@x}$ are unbounded at the interface. In this instance we may not produce accurate results as the nite di erence scheme used would not be able to approximate these unbounded quantities.

Early simulations of the hybrid numerical method have shown that the xed-mesh portion may develop unwanted oscillations at points close to the boundary, which makes estimation of the position of the numerical interface problematic. Although much e ort was made in attempting to reduce these oscillations or improve the means of estimating the interface position, the outcome was unsatisfactory.

6.10 An Alternative Expression for the Velocity

In this section we shall consider the fourth-order problem written as a single fourth-order PDE rather than two second-order equations (and therefore without reference to the $q(x;t)$ function). Setting $m = 1$ in (2.1), we therefore seek solutions $u(x;t)$ to

$$\frac{@u}{@t} = \frac{@}{@x}\left(u^n \frac{@^3u}{@x^3}\right); \qquad \text{in } _T;$$

subject to the initial condition at time $t = t^0$ given by

$$u(x;t^0) = u^0(x); \qquad \text{for } x \; 2 \; (\; t^0);$$

and the boundary conditions

$$\begin{aligned} u &= 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^{\,0};\\ \frac{@u}{@x} &= 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^{\,0};\\ uv \quad u^n \frac{@^3u}{@x^3} &= 0; \qquad \text{at } x = a(t); \quad x = b(t); \quad t > t^{\,0}: \end{aligned}$$

Through conservation of mass the velocity at points in the interior of the domain is

$$v = u^{n\ 1}\frac{@^3u}{@x^3}; \tag{6.19}$$

avoiding any reference toq(x; t).

We now introduce an identity which allows us to rewrite (6.19) as

$$v = u^{n-1}\frac{\partial^3 u}{\partial x^3} - \frac{1}{n}\frac{\partial^3(u^n)}{\partial x^3} - \frac{6(n-1)}{n^2}\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(\frac{\partial(u^{n-2})}{\partial x}\right)^{2}\right) + \frac{27(n-1)(n-2)}{2n^3}\left(\frac{\partial(u^{n-3})}{\partial x}\right)^{3}; \tag{6.20}$$

the proof of which is as follows.

Proof. We shall being by making use of the identity

$$\frac{\partial^3}{\partial x^3}(u^n) - n\frac{\partial^2}{\partial x^2}\left(u^{n-1}\frac{\partial u}{\partial x}\right); \tag{6.21}$$

from which we may write

$$n\frac{\partial^2}{\partial x^2}\left(u^{n-1}\frac{\partial u}{\partial x}\right) - nu^{n-1}\frac{\partial^3 u}{\partial x^3} + 2n\frac{\partial(u^{n-1})}{\partial x}\frac{\partial^2 u}{\partial x^2} + n\frac{\partial u}{\partial x}\frac{\partial^2}{\partial x^2}(u^{n-1});$$
$$nu^{n-1}\frac{\partial^3 u}{\partial x^3} + 3n(n-1)u^{n-2}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + n(n-1)(n-2)u^{n-3}\left(\frac{\partial u}{\partial x}\right)^3: \tag{6.22}$$

Dividing (6.22) through by n

which we may rearrange to obtain

$$(n-1)u^{n-2}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\frac{1}{2}\frac{\partial}{\partial x}\frac{\partial (u^{n-1})}{\partial x}+\frac{1}{2}(n-1)(n-2)u^{n-3}\left(\frac{\partial u}{\partial x}\right)^3;$$
$$\frac{1}{2}(n-1)\frac{\partial}{\partial x}\left[u^{n-2}\left(\frac{\partial u}{\partial x}\right)^2\right]+\frac{1}{2}(n-1)(n-2)u^{n-3}\left(\frac{\partial u}{\partial x}\right)^3. \quad (6.25)$$

Substituting (6.25) into (6.23),

$$u^{n-1}\frac{\partial^3 u}{\partial x^3}+\frac{1}{n}\frac{\partial^3 u}{\partial x^3}+\frac{3}{2}(n-1)\frac{\partial}{\partial x}\left[u^{n-2}\left(\frac{\partial u}{\partial x}\right)^2\right]+\frac{1}{2}(n-1)(n-2)u^{n-3}\left(\frac{\partial u}{\partial x}\right)^3; \quad (6.26)$$

which is almost in the required form.

We may write (6.26) in the required form using the identities

$$u^{n-2}\left(\frac{\partial u}{\partial x}\right)^2 \quad u^{n=2}\frac{\partial u}{\partial x}^2+\frac{4}{n^2}\left(\frac{\partial (u^{n=2})}{\partial x}\right)^2;$$

and

$$u^{n-3}\left(\frac{\partial u}{\partial x}\right)^3 \quad u^{n=3}\frac{\partial u}{\partial x}^3+\frac{27}{n^3}\left(\frac{\partial (u^{n=3})}{\partial x}\right)^3;$$

which results in

$$u^{n-1}\frac{\partial^3 u}{\partial x^3}+\frac{1}{n}\frac{\partial^3 u}{\partial x^3}+\frac{6(n-1)}{n^2}\left(\frac{\partial (u^{n=2})}{\partial x}\right)^2+\frac{27}{2n^3}(n-1)(n-2)\left(\frac{\partial (u^{n=3})}{\partial x}\right)^3;$$

as required. □

Remark. A corresponding identity holds for the second-order problem and in principle exists for the sixth-order problem when the sixth-order PDE is considered as a single equation as opposed to three second-order equations. The details of these corresponding identities are omitted here as the focus is on the fourth-order problem.

In a similar manner to that discussed in §6.4, we can examine the behaviour of the initial alternative velocity at the boundary at points $x = \pm$, when the initial condition is of the form (6.4). For points close to $x = \pm$, we have that $u^0(x)$ is of the form (6.5) and using this we can examine the various terms in (6.20).

The first term on the RHS of (6.20) is given by

$$\frac{1}{n}\frac{\partial^3(u^n)}{\partial x^3} = 12c_1^n\ (n-1)x(\beta^2 - x^2)^{n-2} - 8c_1^n\ (n-1)(n-2)x^3(\beta^2 - x^2)^{n-3};$$

which, for $x \to \beta$,

$$\frac{1}{n}\frac{\partial^3(u^n)}{\partial x^3} \to c_1^n\ (n-1)\ 3(2\beta)^{n-1}(\beta - x)^{n-2} - (n-2)(2\beta)^n\ (\beta - x)^{n-3}.$$

Similarly, the second term is given by

$$\frac{6(n-1)}{n^2}\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(\frac{\partial(u^{n-2})}{\partial x}\right)^2\right) = 12c_1^n(n-1)^2 x(\beta^2 - x^2)^{n-2} - (n-2)x^3(\beta^2 - x^2)^{n-3};$$

which for $x \to \beta$ has the form

$$\frac{6(n-1)}{n^2}\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\left(\frac{\partial(u^{n-2})}{\partial x}\right)^2\right) \to c_1^n(n-1)^2\, 3(2\beta)^{n-1}(\beta - x)^{n-2} - \frac{3}{2}(n-2)(2\beta)^n\ (\beta - x)^{n-3}.$$

Finally, the third term is given by

$$\frac{27(n-1)(n-2)}{2n^3}\left(\frac{\partial(u^{n-3})}{\partial x}\right)^3 = \frac{8}{2}c_1^{n-3}(n-1)(n-2)x^3(\beta^2 - x^2)^{n-3};$$

which for $x \to \beta$,

$$\frac{27(n-1)(n-2)}{2n^3}\left(\frac{\partial(u^{n-3})}{\partial x}\right)^3 \to \frac{1}{2}c_1^{n-3}(n-1)(n-2)(2\beta)^n\ (\beta - x)^{n-3}.$$

The velocity (6.20) at points $x \to \beta$ is therefore

$$\begin{aligned} v \to\ & 3c_1^n(2\beta)^{n-1}\ (n-1)\ (n-1)^2\ (\beta - x)^{n-2} \\ & (c_1)^n(2\beta)^n\ (n-1)(n-2)\ \frac{3}{2}^2(n-1)(n-2) \\ & + \frac{1}{2}^3(n-1)(n-2)\ (\beta - x)^{n-3}. \end{aligned} \tag{6.27}$$

Examining (6.27), we see that the velocity consists entirely of terms which are smooth for choices of $n \geq 3$, indicating that they should possess a limit at the boundary in these cases. For cases where $n < 3$ the velocity becomes unbounded as $x \to \beta$ as expected.

6.10.1 Approximating the Alternative Velocity

We can readily approximate the terms in (6.20) using nite di erences to obtain an

N	n = 8=5, = 15=8	n = 2, = 3=2
21	0.0502	0.0289
41	0.0445	0.0311
81	0.0460	0.0321
161	0.0468	0.0324
321	0.0470	0.0325
641	0.0471	0.0325
1281	0.0471	0.0325
v_b^0	0:04	0.03

Table 6.8: Initial boundary velocities for the nite advance experiments using the alternative velocity. The expected boundary velocity is given in the nal row for each case.

N	n = 6=5, = 5=2	n = 15=11, = 11=5
21	-0.7990	-0.1323
41	-0.9314	-0.1755
81	-0.9397	-0.1762
161	-0.9386	-0.1750
321	-0.9380	-0.1746
641	-0.9378	-0.1745
1281	-0.9377	-0.1744
v_b^0	0:95	0:18

Table 6.9: Initial boundary velocities for the nite retreat experiments using the alternative velocity. The expected boundary velocity is given in the nal row for each case.

N	n = 5=2, = 5=2	n = 2, = 2
21	-0.0207	-0.0079
41	5:53 10 4	6:75 10 5
81	1:84 10 4	9:22 10 4
161	7:77 10 6	3:96 10 4
321	7:12 10 7	1:23 10 4
641	1:87 10 7	3:41 10 5
1281	2:65 10 8	8:94 10 6
v_b^0	0	0

Table 6.10: Initial boundary velocities for the waiting-time experiments using the alternative velocity. The expected boundary velocity is given in the nal row for each case.

In an attempt to understand this di erence, we investigate the computed boundary velocity for one of the experiments to determine whether oscillations are still appearing close to the boundary. As in x6.6.1, we select the nite retreat case (with n = 6=5, = 15=8 and n = 3) and plot the computed initial velocity for the nal ve mesh points in the N = 81 case and the nal 20 mesh points of the mesh in the N = 1281 case. These computed velocities are given in Figure 6.8, along with the exact $v^0(x)$ calculated using (6.20).

From Figure 6.8 it is clear that there are still oscillations forming close to the boundary, which a ects the nal computed boundary velocity. These oscillations are much smaller in magnitude than those seen in Figure 6.1 however, which explains why the computed velocities are much closer to the expected values. We also observe that the oscillations are larger for the N = 1281 mesh, which can explain why the computed boundary velocity is slowly moving away from the expected values.

It is likely that the oscillations observed here may arise from the numerical approximation of the terms in (6.20) as opposed to the presence of the singularity in the original experiments in (x6.6.1). The nite di erence approximation of the derivatives for mesh

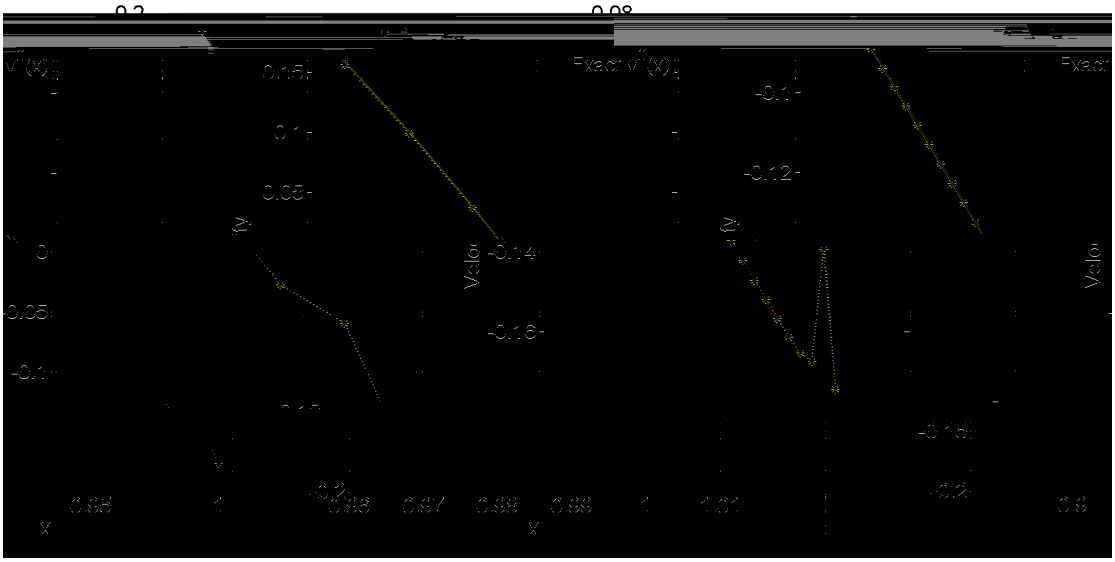

Figure 6.8: The computed alternative velocity V_i^0 for the n = 6=5, = 15=8 nite retreat case. The left plot shows the nal ve mesh points for the N = 81 mesh, while the right plot shows the N = 1281 mesh plotted over the nal 20 mesh points. The green line in each plot is the exact$v^0(x)$ calculated from (6.20)

term. It may be possible to modify the scheme used to approximate this term such that the MPCMb possesses the S Property over a single time step (if scale-invariant time

original MPCM.

In the nal chapter we shall provide a summary of the thesis, including conclusions drawn throughout. We shall also provide a discussion on potential future directions of the work.

Chapter 7

Conclusions and Further Work

In this final chapter we shall summarise the work in this thesis and present conclusions that can be drawn from the progress made. We then provide an outline of future directions in which the subject can be taken, either related directly to the work covered in the thesis or natural extensions to the work which have not been previously covered.

7.1 Summary

In Chapter 1 we outlined the main aims of the thesis. These were to:

> Extend the finite difference work of Parker (2010) for a particular second-order problem to higher orders (fourth and sixth) as well as to more general second-order problems using a finite difference implementation of the velocity-based conservation method. These implementations, which use a scale-invariant time stepping scheme, are theoretically able to propagate similarity solutions forward in time to essentially within rounding error of the exact solution over a single time step. We term this property of the numerical method the S Property.
>
> Produce an implementation of the BHJ method using higher order basis functions and scale-invariant time stepping in order to propagate similarity solutions forward in time to within rounding error. In principle, this finite element implementation

The majority of Chapter 3 focused on introducing the MPCM for the various orders of problem considered. The conservation method uses local mass conservation in order to establish the velocities of interior mesh points (with extrapolation used to obtain the velocity at the boundary points), which is then used to move the mesh in time using a suitable time stepping method. The local mass conservation is then used to recover the solution algebraically on the updated mesh at the next time step. The chapter finishes with a definition of the S Property, which we demonstrate in later chapters that the MPCM possesses in certain circumstances.

In Chapter 4 the MPCM was modified for the fourth order problem (2.3)–(2.5) with $n = 1$ in order to demonstrate that the method possesses the S Property when the initial condition coincides with a similarity solution and a scale-invariant time stepping scheme is used. By modifying the schemes used, the truncation error of each step of the method was shown to be equal to zero over a single time step. This property was verified by numerical results, although multiple time steps highlighted a build up of rounding error incurred by the time stepping method. This build up of error was investigated and bounded.

Chapter 4 also provided insight into how the MPCM possesses the S Property for the second-order problem (2.7)–(2.9) for any choice of n. For the sixth-order problem (2.10)–(2.12) with $n = 1$, the required modifications to the schemes in the MPCM are presented. Numerical verification is not provided for the sixth-order problem due to a lack of a working program code.

A departure from finite difference implementations took place in Chapter 5, with the description of the FEMPCM. This required the introduction of weak forms of the various equations in the different problems considered. We described the BHJ method (from which the FEMPCM originated) with a few minor alterations for the fourth-order problem (2.3)–(2.5), using piecewise linear basis functions throughout. We then demonstrated that by choosing the basis functions in the FEMPCM to be of the same order as the exact solution, the method for the fourth-order problem (2.3)–(2.5) with $n = 1$ could be shown to possess the S Property. Numerical results were presented which

verify the possession of the S Property, with a build up of rounding error present as in the MPCM.

We then described the steps required to modify the FEMPCM for the second-order problem (2.7){(2.9) with any

For problems in which the method would be expected to possess this Property, this uniform mesh is suitable for producing results of the required accuracy. In the case of the problems examined in Chapter 6, the initial mesh may bene t from being focused closer to the boundary (with alternative means of initially distributing the mesh points are discussed).

A possible means of alleviating the issue of modelling an initially unbounded boundary velocity was proposed in the form of a hybrid numerical method, combining a xed-mesh method with the MPCM. At present this method is unveri ed, but a description of how this could be implemented is given in x6.9, along with a critique of the issues to be addressed in the implementation.

Finally, a formulation with an alternative expression for the velocity was proposed which, when implemented numerically, alleviates the issues experienced with approximating the $q(x;t)$ function (for $n \quad 3$). This alternative velocity does not contain any of the products of terms which were shown to be problematic in the previous implementation. Numerical results show that this alternative velocity produces much improved initial boundary velocities and is therefore a potential solution to the problems experienced previously.

7.2 Conclusions

In conclusion, we have demonstrated both advantages and disadvantages in the use of the MPCM for solving the type of nonlinear di usion problems discussed in this thesis. The ability of the MPCM to propagate similarity solutions forward in time to the accuracy demonstrated in Chapter 4 is noteworthy and the manner in which it can apply to di erent orders of problem is also of interest. The problems highlighted in

forward in time to within rounding error if a scale-invariant time stepping scheme is used) for problems of various orders (second/fourth/sixth) when the numerical schemes are carefully modified

In chapter 4 the numerical schemes of the MPCM were modified from the schemes originally described in Chapter 3. These modified schemes were shown to have zero local truncation error over a single time step when a similarity solution is used as an initial condition and through the use of a scale-invariant time stepping scheme. In practice we observed a build up of rounding error which was shown to be due to the time stepping part of the method and this error was bounded. This level of accuracy is present even in a small number of nodes, which reduces the computational cost of the method.

The FEMPCM can be implemented through careful choice of approximation spaces so that it can also possess the S Property, producing a solution valid over the whole of the domain at an increased computational cost (when compared to the MPCM which is only valid at the mesh points).

Chapter 5 demonstrated that an implementation of the FEMPCM can also be constructed which possesses the S Property for the same problems as previously mentioned. As in the MPCM, a build up of rounding error is observed in the numerical results, which is attributable to the time stepping method.

There are a range of situations in which the MPCM (and FEMPCM) is unable to accurately model the behaviour of the moving boundary of the fourth-order problem (2.3){(2.5), either due to the $q(x;t)$ function or its derivative becoming unbounded at the boundary or due to the boundary being expected to move with an initially unbounded velocity.

In Chapter 6 we explored the applications of the MPCM to the fourth-order problem (2.3){(2.5) for more general cases, since the ability of the method to produce solutions for general n and for more general initial conditions is of interest to the wider community. It was noted that for initial conditions of the form (6.4) there

are choices of initial condition in which either the velocity or the functions

The second alternative involves avoiding the approximation of the $q(x;t)$ function

Implementation of the hybrid numerical method is also a possible future research avenue, as discussed in §6.9. The main issue with the method is the time point at which the method switches from a fixed-mesh to a moving-mesh method, as this is not obvious a-priori. This value may also be dependent upon the value of n and the initial condition used, since it is intended that the fixed-mesh part of the method be run until the boundary velocity is able to be approximated accurately by the moving-mesh portion.

There is also more work to be achieved on the current implementations of the MPCM and FEMPCM, particularly with regard to the sixth-order problem (2.10)–(2.12). Numerical results were omitted from this thesis for the MPCM as a working program code was not available. The method as described in Chapter 4 is expected to possess the S_6 Property, but currently the program code is unable to approximate the quartic $p(x;s)$ function to the required accuracy.

Finally, a natural extension to the work in this thesis is to move into higher spatial dimensions. A logical extension into two spatial dimensions would be an implemen-

Bibliography

D.G. Aronson. The porous media equation. In A. Fasano and M. Primicerio, editors,

order nonlinear degenerate parabolic equation. Numerische Mathematik, 96:401{434, 2004.

T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. Computer methods in applied mechanics and engineering, 139(1):3{47, 1996.

E. Beretta, M. Bertsch, and R. Dal Passo. Nonnegative solutions of a fourth-order nonlinear degenerate parabolic equation. Arch. Rational Mech. Anal., 129:175{200, 1995.

F. Bernis. Finite speed of propagation for thin viscous ows when 2 n 3. C.R. Acad. Sci. Paris Ser. I Math , 322:1169{1174, 1996a.

F. Bernis. Finite speed of propagation and continuity of interface for thin viscous ows.

J.F. Blowey, J.R. King, and S. Langdon. Small and waiting time behaviour of the thin lm equation. SIAM J. Appl. Math. , 67:1776{1807, 2007.

P. Bochev, G. Liao, and G. dela Pena. Analysis and computation of adaptive moving grids by deformation. Numerical Methods for Partial Di erential Equations , 12(4): 489{506, 1996.

J. Bouwe Van Den Berg, M. Bowen, J.R. King, and M.M.A. El-Sheikh. The self-similar solution for draining in the thin lm equation. Euro. Jnl of Applied Mathematics, 15: 329{346, 2004.

M. Bowen and J.R. King. Asymptotic behaviour of the thin lm equation in bounded domains. Euro. J. Appl. Math. , 12:135{157, 2001.

M. Bowen, J. Hulshof, and J.R. King. Anomalous exponents and dipole solutions for the thin lm equation. SIAM J. Appl. Math. , 62:149{179, 2001.

C.J.W. Breward, H.M. Byrne, and C.E. Lewis. The role of cell-cell interactions in a two-phase model for avascular tumour growth. J. Math. Biol. , 45:125{152, 2002.

J. Buckmaster. Viscous sheets advancing over dry beds. J. Fluid. Mech., 81:735{756, 1977.

C.J. Budd and M.D. Piggott. The geometric integration of scale-invariant ordinary and partial di erential equations. Journal of Computational and Applied Mathematics, 128:399{422, 2001.

C.J. Budd and J.F. Williams. Moving mesh generation using the parabolic monge-ampere equation. SIAM Journal on Scienti c Computing , 31(5):3438{3465, 2009.

C.J. Budd, W. Huang, and R.D. Russell. Moving mesh methods for problems with blow-up. SIAM Journal on Scienti c Computing , 17(2):305{327, 1996.

C.J. Budd, G.J. Collins, W.Z. Huang, and R.D. Russell. Self-similar numerical solutions of the porous medium equation using moving mesh methods. Philosophical

Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 357:1047{1077, 1999.

C.J. Budd, B. Leimkuhler, and M.D. Piggott. Scaling invariance and adaptivity. Applied numerical mathematics, 39(3):261{288, 2001.

C.J. Budd, W. Huang, and R.D. Russell. Adaptivity with moving grids. Acta Numerica, 18:111{241, 2009.

C.J. Budd, M.J.P. Cullen, and E.J. Walsh. Monge{amp`ere based moving mesh methods

E.B. Dussan and S.H. Davis. On the motion of a fluid-fluid interface along solid surface. J. Fluid Mech., 65:71–95, 1974.

J.D. Evans, M. Vynnycky, and S.P. Ferro. Oxidation-induced stresses in the isolation oxidation of silicon. Journal of engineering mathematics, 38(2):191–218, 2000.

J.D. Evans, V.A. Galaktionov, and J.R. King. Source-type solutions of the fourth-order unstable thin film equation. Euro. Jnl of Applied Mathematics, 18:273–321, 2007a.

J.D. Evans, V.A. Galaktionov, and J.R. King. Unstable sixth-order thin film equation: I. blow-up similarity solutions. Nonlinearity, 20:1799–1841, 2007b.

J.D. Evans, V.A. Galaktionov, and J.R. King. Unstable sixth-order thin film equation: Ii. global similarity patterns. Nonlinearity, 20:1843–1881, 2007c.

J.C. Flitton and J.R. King. Moving-boundary and fixed-domain problems for a sixth-order thin-film equation. Euro. Jnl of Applied Mathematics, 15:713–754, 2004.

L. Giacomelli, H. Knüpfer, and F. Otto. Smooth zero-contact-angle solutions to a thin-film equation around the steady state. J. Differential Equations, 245:1454–1506, 2008.

H.P. Greenspan. On the motion of a small viscous droplet that wets a surface. J. Fluid Mech., 84:125–143, 1978.

G. Grün. On the convergence of entropy consistent schemes for lubrication type equations in multiple space dimensions. Mathematics of Computation, 72:1251–1279, 2003.

G. Grün and M. Rumpf. Nonnegativity preserving convergent schemes for the thin film equation. Numerische Mathematik, 87:113–152, 2000.

G. Grün and M. Rumpf. Simulation of singularities and instabilities arising in thin film flow. European J. Appl. Math., 12:293–320, 2001.

D. Huang, Q. Yang, and S. Zhou. Group properties and invariant solutions of a sixth-order thin film equation in viscous fluid. Journal of Mathematical Physics, 54:013510, 2013.

W. Huang and R.D. Russell. A high dimensional moving mesh strategy. Applied Numerical Mathematics, 26(1):63{76, 1997.

W. Huang and R.D. Russell. Moving mesh strategy based on a gradient ow equation for two-dimensional problems. SIAM Journal on Scienti c Computing , 20(3):998{1015, 1999.

W. Huang and R.D. Russell. Adaptive Moving Mesh Methods Springer, New York, 2011.

W. Huang, Y. Ren, and R.D. Russell. Moving Mesh Partial Di erential Equations (MMPDES) Based on the Equidistribution Principle. SIAM J. Numer. Anal. , 31: 709{730, 1994.

M. Hubbard, M.J. Baines, and P.K. Jimack. Consistent dirichlet boundary conditions for numerical solution of moving boundary problems. Applied Numerical Mathematics, 59:1337{1353, 2009.

J. Hulshof and A.E. Shishkov. The thin lm equation with 2 n < 3: Finite speed of propogation in terms of the l^1-norm. Adv. Di erential Equations , 3:625{642, 1998.

H. Jiang and N. Wei-Ming. On steady states of van der waals force driven thin lm equations. Euro. J. Appl. Math. , 18:153{180, 2007.

A. Jungel and R. Pinnau. A positivity-preserving numerical scheme for a nonlinear fourth order parabolic system. SIAM J. Numer. Anal. , 39:385{406, 2001.

W.L. Kath and D.S. Cohen. Waiting-time behavior in a nonlinear di usion equation. Studies in Applied Mathematics, 67:79{105, 1982.

J.R. King. Mathematical aspects of semiconductor process modelling PhD thesis, University of Oxford, 1986.

J.R. King. The isolation oxidation of silicon. SIAM J. Appl. Math. , 49:264{280, 1989a.

J.R. King. The isolation oxidation of silicon: the reaction-controlled case. SIAM J. Appl. Math., 49:1064{1080, 1989b.

J.R. King. Two generations of the thin lm equation. Mathematical and Computer Modelling, 34:737{756, 2001.

J.R. King and M. Bowen. Moving boundary problems and non-uniqueness for the thin lm equation. Euro. J. Appl. Math., 12:321{356, 2001.

J.R. King and R.M. Taranets. Asymmetric travelling waves for the thin lm equation. J. Math. Anal. Appl., 404:399{419, 2013.

B.F. Knerr. The porous medium equation in one dimension. Trans. Amer. Math. Soc., 234:381{415, 1977.

K. Miller and R.N. Miller. Moving nite elements. i. SIAM Journal on Numerical Analysis

N.F. Smyth and J.M. Hill. High-Order Nonlinear Di usion. IMA Journal of Applied Mathematics, 40:73{86, 1988.

E.A. Socolovsky. On the numerical approximation of nite speed di usion problems. Numerische Mathematik, 53:97{105, 1988.

J.L. Vasquez. The Porous Medium Equation: Mathematical Theory. Oxford University Press, 2007.

L. Zhornitskaya and A.L. Bertozzi. Positivity-preserving numerical schemes for lubrication-type equations. SIAM J. Numer. Anal. , 37:523{555, 2000.