A Moving Collocation Scheme for Fourth Order Evolutionary Partial Differential Equations

by

Xiangmin XU

B.Sc., Xin Yang Normal University, 1995

M.Sc. Institute of Systems Science, Chinese Academy of Sciences, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN THE DEPARTMENT
OF
MATHEMATICS

SIMON FRASER UNIVERSITY

October 2003

APPROVAL

Name: Xiangmin XU

Degree: Master of Science

Title of thesis: A Moving Collocation Scheme for Fourth Order Evolutionary Partial Differential Equations

Examining Committee: Dr. Steven Ruuth
Chair

Dr. Robert Russell
Senior Supervisor

Dr. Manfred Trummer
Supervisor

Dr. Ralf Wittenberg
Supervisor

Dr. David Muraki
Internal/External Examiner

Date Approved: October 30, 2003

PARTIAL COPYRIGHT LICENCE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

A Moving Collocation Scheme for Fourth Order Evolutionary Partial Differential Equations

Author:

(signature)

XIANGMIN XU

(name)

Dec 8, 2003

(date)

Abstract

Fourth order model equations arise in many physical applications such as thin film theory, lubrication theory, convection-explosion theory, flame and wave propagation, phase transition at critical Lipschitz points, bi-stable systems and so on. In recent years, degenerate parabolic equations of fourth order became important for modelling diffusion processes in Physics and Material Sciences. Indeed, fourth order terms are increasingly recognized as being significant in many physical models, and this has led to the burgeoning literature.

The aim of this thesis is to study the moving collocation method for solving fourth order evolutionary partial differential equations. We start by presenting several types of fourth order evolutionary partial differential equations. Some difficulties we will encounter when numerically solving these fourth order evolutionary partial differential equations are addressed. A high resolution moving collocation scheme which uses septic Hermite splines is then introduced. MC4, a subroutine written in FORTRAN which implements the moving collocation method has been developed. The code is basically based on MOVCOL by Huang and Russell in 1996. Some new features of the code include a large selection of monitor functions and a conservation checking mechanism. Numerical examples are included to show the efficacy of the method.

Acknowledgments

I would like to give my sincere thanks to my supervisor Dr. Robert D. Russell for all his patience, guidance, encouragement and constant support during my study at Simon Fraser University.

Special thanks to Dr. Jeff Williams from the University of Bath for all the comments and test problems he sent in.

Special thanks to my colleagues in the department of mathematics, Simon Fraser University for all their support.

Contents

List of Figures

Chapter 1

Introduction

Well-chosen model equations have always played an important role in the study of complex patterns in physical, mechanical, chemical and biological systems. Classical model equations have typically been second-order partial differential equations. Very familiar examples are the heat equation, the wave equation, and the Laplace equation. They are designed to shed light on specific physical processes, such as diffusion, dispersion, absorption or wave propagation and their mutual interaction, and in some cases they also give a quantitative description of such processes.

While these second order model equations capture the essential features of specific problems, they are often much simpler than the full equations describing the physical processes. They are limited in their ability to describe complex features. When complex patterns such as pulses that consist of a series of spikes or other localized patterns with nontrivial structure develop, the limitation of these model equations becomes acute. To gain insight into the dynamics of the complex spatial and temporal patterns in a wide range of physical and mechanical problems, higher order equations or systems of equations have been proposed of which fourth order model equations commonly arise.

Fourth order model equations arise in many physical applications such as thin film theory, lubrication theory, convection-explosion theory, flame and wave propagation, phase transition at critical Lipschitz points, bi-stable systems and so on. In recent

years, degenerate parabolic equations of fourth order have become important for modelling diffusion processes in physics and material sciences. Indeed, fourth order terms are increasingly recognized as being significant in many physical models and this has led to the burgeoning literature including the recent book [43] which lists a number of models and references.

The investigation of higher order parabolic equations began almost at the same time as that of second order equations; however, the study of higher order equations, especially of quasi-linear equations, is far from complete compared to that of the second order case. The main reason is that many effective methods used in treating second order equations, such as those based on maximum principles, are no longer effective for higher order equ ations. In addition, the occurrence of degeneracy makes things even more difficult. The complex structure of fourth order partial differential equations also poses a challenging problem for the design of numerical methods for solving these problems.

Difficulties and challenges involved in numerically solving a fourth order evolutionary partial differential equations include preserving the non-negativity of the solution in the thin film equation, preserving the total mass within the Cahn-Hilliard equation, handling the case where blow-up and fast-decay are present, removing the singularities in the degenerate fourth order partial differential equations, as well as capturing the complex patterns in equations of gradient type.

The aim of this thesis is to study the theory and computation of a moving collocation method for solving fourth order evolutionary partial differential equations. We start by presenting several types of fourth order time-dependent partial differential equations and briefly addressing several challenges one normally will encounter when numerically solving them. Then, in view of the nature and breadth of the problems considered, a moving collocation method for solving fourth order evolutionary partial differential equations is introduced. This method is basically an extension of the moving collocation method for solving second order evolutionary partial differential equations. MC4, a subroutine written in FORTRAN which implements the moving collocation method has been developed. The code is basically based on MOVCOL by Huang and Russell in 1996 [31].

Numerical experiments have been done successfully for the thin film equation, the interface fluctuation equation, the Cahn-Hilliard equation and some other problems. Numerical results show that the moving collocation method introduced in this thesis handles singularities effectively. For the thin film models within which the moving contact lines are present, the positivity of the solution is preserved without special treatment, while in many articles great efforts have been taken to design a code to preserve the positivity of the solution to this equation. Mass conservation is also checked and the implication in solving equations with finite time blow-up or fast decay are discussed.

The layout of this thesis is as follows. In Part II, we present a number of fourth order time-dependent partial differential equations, the basic concepts of self-similar solutions and scaling invariant PDEs, and some issues related to numerically solving fourth order evolutionary partial differential equations. In Part III, we describe the MMPDE approach and introduce the moving collocation method for solving fourth order evolutionary partial differential equations. Conclusions are made in part IV, and some aspects worthy of further investigation are addressed.

It should be noted that it is Dr. Jeff Williams from University of Bath who first extended MOVCOL—the moving collocation method for solving second order evolutionary partial differential equation systems, to MC4—a moving collocation method for solving fourth order evolutionary partial differential equations.

Chapter 2

Evolutionary PDEs of Fourth Order

2.1 Several Types of Evolutionary PDEs of Fourth Order

The theory of fourth order parabolic equations presents many interesting mathematical problems. We summarize those we have been investigating as follows:

Type I: Thin film equations

$$u_t = -(u^n u_{xxx})_x, \quad x \in \mathbb{R}, t > 0, n > 0. \tag{2.1}$$

Thin film equations of the above form are derived from a lubrication approximation. They model the surface tension dominated motion of thin viscous films and spreading droplets [37]. The different values of n are related to the boundary conditions at the bottom used at the Navier-Stokes level. The most important cases in applications are $n = 3$, which corresponds to no-slip boundary conditions, and $n = 1$, which corresponds to the so-called Hele-Shaw cell [4]

One of the remarkable features of equation (2.1) is that the nonlinearity guarantees the nonnegativity preserving property of the solution [14].

Solution to (2.1) are also known to preserve mass [2], i.e.

$$\int_{\mathbb{R}} u(x,t)dx = \int_{\mathbb{R}} u_0(x)dx = M \tag{2.2}$$

It is shown in [4], [18] that the support of the solution has finite speed of propagation and continuous flux $u^n u_{xxx}$. They also show a rapid convergence of the solution onto the similarity solution before the merging of the support of the solution.

Recent studies of singularities in which $u = 0$ at a point, describing rupture of the fluid layer, show that such equations exhibit complex dynamics which can be difficult to simulate accurately. In particular, one must ensure that the numerical approximation of the interface does not show a false premature rupture. Generic finite difference schemes have the potential to manifest such instabilities especially when under-resolved [54].

The thin film equations of the following form [15],

$$u_t = -(uu_{xxx})_x - (u^m u_x)_x \tag{2.3}$$

with $m \geq 3$, are long-wave unstable: flat profiles can't remain linearly stable with respect to infinitesimal perturbations which have sufficiently large wave numbers. The fourth order term is linearly stabilizing and the second order term is linearly destabilizing. When $m \geq 3$, equation (2.3) blows up in finite time. For the critical case $m = 3$ and for the supercritical case $m > 3$, they have self-similar solutions with remarkably different properties.

A family of equations of the form:

$$u_t = -(f(u)u_{xxx})_x - (g(u)u_x)_x \tag{2.4}$$

is considered in [15]. These equations are long wave unstable if f and g perturb around a positive constant steady state, hence f and g are nonnegative.

It is conjectured in [15] that the large-u behavior of $g(u)/f(u)$ determines the presence or absence of a finite-time blow-up:

$$\lim_{y\to\infty} \frac{g(y)}{y^2 f(y)} = \begin{cases} \infty & : \ u \to \infty \text{ in finite time for some solutions} \\ \text{finite} & : \ \text{marginal case} \\ 0 & : \ \text{globally stable solutions.} \end{cases}$$

Another type of thin film equation is the thin film equation with absorption [26]:

$$u_t = -u_{xxxx} + |u|^p, x \in \mathbb{R}, t > 0, p > 1. \tag{2.5}$$

It is known that $p_F = 3$ is the critical Fujita exponent for (2.5):

1) If $p \in (1, p_F]$, then blow-up occurs for arbitrarily small initial data with non-negative first Fourier coefficient. Different initial data will have different blow-up regimes.

2) If $p > p_F$, then solutions are global in time for any sufficiently small initial data u_0.

Type II Interface fluctuations equation

$$u_t = -(u(\log u)_{xx})_{xx} \tag{2.6}$$

with non flux boundary condition

$$u_x(0,t) = u_x(1,t) = 0, \quad u_{xxx}(0,t) = u_{xxx}(1,t) = 0$$

or with periodic boundary condition

$$u(0,t) = u(1,t) = 1, \quad u_x(0,t) = u_x(1,t) = 0.$$

This equation arises as a scaling limit in the study of interface fluctuations in a certain spin system [23] and also models the electron concentration in a quantum semiconductor device with zero temperature and negligible electric field [35].

It is remarked that

1) Equation (2.6) is a particular case of a class of fourth order diffusion equations which admit self-similar solutions.

2) The solution shows an algebraic decay in the L^1 norm towards the corresponding self-similar profile.

3) The solution converges to the constant steady state $\int_{[0,1]} u_0$ very fast, where u_0 is strictly positive initial data.

Type III Cahn-Hilliard Equation

$$u_t + \gamma u_{xxxx} = (F'(u))_{xx} \ . \tag{2.7}$$

where $\gamma > 0$. This equation was derived by Cahn and Hilliard in 1958 to model the phenomenon of first order phase separation in binary alloys. Other than phase separation in binary alloys, diffusive processes such as the growth and dispersal in population can also be described by the Cahn-Hilliard equation.

In their book [27], V.A. Galaktionov and J.L. Vazquez addressed the concern about the asymptotic blow-up behavior of the following Cahn-Hilliard equation with blow-up

$$u_t + \gamma u_{xxxx} = (-u^3 + \gamma_1 u^2 - u)_{xx} \ . \tag{2.8}$$

Type IV: Equations of gradient type [17]

$$u_t = -\gamma u_{xxxx} + \beta u_{xx} - F'(u) \tag{2.9}$$

with $\gamma > 0$, and $F(u)$ is a potential.

Equations of gradient type occur in many physical models. Two typical examples are the Swift-Hohenberg equation and the extended Fisher-Kolmogorov equation.

For different models, different boundary conditions are adopted. For example, the Neumann Boundary conditions

$$u_x(0,t) = u_{xxx}(0,t) = 0, u_x(L,t) = u_{xxx}(L,t) = 0$$

are often used in the Swift-Hohenberg equation and the extended Fisher-Kolmogorov equation.

There are other boundary conditions such as Navier boundary conditions

$$u(0,t) = u(L,t) = 0, u_{xx}(0,t) = u_{xx}(L,t) = 0$$

and the general Dirichlet boundary conditions

$$u(0,t) = y_1(t); \quad u_x(0,t) = y_2(t); \quad u(L,t) = z_1(t); \quad u_x(L,t) = z_2(t)$$

where $y_1(t), y_2(t), z_1(t), z_2(t)$ are any suitable values that u and u_x may attain at $x = 0$ and $x = L$, respectively.

These types of equations usually display a multitude of stable stationary states.

Type V: Modified KS equation with blow-up

$$u_t = -u_{xxxx} - u_{xx} + (1-\lambda)(u_x)^2 \pm \lambda(u_{xx})^2, \lambda \in (0,1). \tag{2.10}$$

This equation admits interesting blow-up patterns — see [43] for a systematic study on stable patterns of these equations.

There are other types of fourth order equations such as the beam equations arising in the study of an elastic bar with non-convex stored energy density

$$u_{tt} + a^2 u_{xxxx} = g(x,t),$$

the suspension bridge equation arising in the study of the traveling waves in a suspension bridge

$$u_{tt} + u_{xxxx} + |u-1| - 1 = 0,$$

and the Boussinesq equations arising in the study of the effects of weak dispersion and nonlinearity in a shallow water

$$u_{tt} = (u_{xx} + \frac{1}{2}u^2)_{xx}. \tag{2.11}$$

For the time being, we only solve problems of the form

$$\begin{cases} F(x,t,u,u_x,u_{xx},u_{xxx},u_t,u_{xt},u_{xxt},u_{xxxt}) = G_x(x,t,u,u_x,u_{xx},u_{xxx},u_t,u_{xt},u_{xxt},u_{xxxt}) \\ u(x,t_a) = u_0(x) \\ B_1^L(t,x^L,x_t^L,u^L,u_x^L,u_{xx}^L,u_{xxx}^L,u_{xxxx}^L,u_t^L) = 0 \\ B_2^L(t,x^L,x_t^L,u^L,u_x^L,u_{xx}^L,u_{xxx}^L,u_{xxxx}^L,u_t^L) = 0 \\ B_1^R(t,x^R,x_t^R,u^R,u_x^R,u_{xx}^R,u_{xxx}^R,u_{xxxx}^R,u_t^R) = 0 \\ B_2^R(t,x^R,x_t^R,u^R,u_x^R,u_{xx}^R,u_{xxx}^R,u_{xxxx}^R,u_t^R) = 0 \end{cases} \tag{2.12}$$

where the superscripts L and R denote the value at the left and right boundary, respectively, e.g., $u_{xx}^R = u_{xx}(x^R,t)$.

2.2 Similarity Solutions of Fourth Order Equations

Finite time blow-up is a common phenomenon in many physical applications, particularly in combustion theory and nonlinear optics. To gain a fundamental insight

into systems which develop singularities in finite time, scaling and self-similarity were introduced in the 1930's and have been an important tool since then.

The nonexistence of self-similar solutions for second-order semi-linear parabolic equations has been known for at least twenty years. Of increasing interest in applications are equations with higher order spatial derivatives. With a fourth order model from explosion-convection theory, it was shown in [8] that nonexistence of self-similar solutions is not necessary the case for higher order equations.

To be more specific, consider a general partial differential equation which is invariant under the scaling

$$t' = \lambda t, \quad x' = \lambda^m x, \quad u' = \lambda^n u \tag{2.13}$$

for any positive λ. A similarity solution to a PDE is any solution which is invariant under this scaling. Many interesting fourth order PDEs, including problems leading to blow-up, are scaling invariant.

For example, the equation

$$u_t = -u_{xxxx} + |u|^p \tag{2.14}$$

is invariant under the scaling

$$t' = \lambda t, \quad x' = \lambda^{1/4} x, \quad u' = \lambda^{\frac{1}{1-p}} u. \tag{2.15}$$

For $p = 2$, this problem blows up at a certain time, denoted by T. Setting $\lambda = T - t$, we can easily show that this equation has a self-similar solution of the form

$$u(x,t) = \frac{f(x/(T-t)^{1/4})}{T-t}.$$

Similarly, the equation

$$u_t = -(u^3 u_x + u u_{xxx})_x \tag{2.16}$$

is scaling invariant under the scaling

$$t' = \lambda t, \quad x' = \lambda^{1/5} x, \quad u' = \lambda^{-\frac{1}{5}} u.$$

From this, it can be easily shown that equation (2.16) has a self-similar solution of the form

$$u(x,t) = \frac{f(x/(T-t)^{1/5})}{(T-t)^{1/5}}$$

for a positive constant T such that $T - t > 0$ for suitable range of t, say $t \in (t_a, t_b)$.

Generally, if a partial differential equation is invariant under the scaling (2.13) for any positive λ, then this equation has a self-similar solution of the form

$$u(x,t) = (T-t)^n f(x/(T-t)^m). \tag{2.17}$$

2.3 Other Issues Related to Numerically Solving a Fourth Order PDE

2.3.1 Compute Solutions near Singularity Time

Central to the singularity formation phenomenon for fourth order equations is the concept of finite time blow-up, where the solution of the Cauchy problem with uniformly bounded initial data $u_0(x)$ becomes unbounded at some finite time T in the sense that $u(x,t)$ exists and is classical on any time-interval $[0,T']$ with $T' \in (0,T)$ and

$$\sup_{x\in\mathbb{R}} |u(x,t)| \to \infty, \text{as } t \to T^-.$$

Finite time blow-up for fourth order semilinear and quasilinear parabolic equations has been well known since the 1970s [25]. It involves a delicate balance between the spatial and temporal derivatives and the reaction terms driving the blow-up.

Lubrication equations are fourth order degenerate diffusion equations of the form $h_t + \nabla \cdot (f(h)\nabla\Delta h) = 0$, describing thin films or liquid layers driven by surface tension. Recent studies of singularities in which $h \to 0$ at a point, describing rupture of the fluid layer, show that such equations exhibit complex dynamics which can be difficult to simulate accurately. In particular, one must ensure that the numerical approximation of the interface does not show a false premature rupture. Generic finite difference schemes have the potential to manifest such instabilities especially when under-resolved.

It is a challenging problem to compute solutions towards singularity time T. Even with implicit methods, one needs certain grid adaptation so as to have a more refined mesh near blowup points, and a coarse grid elsewhere.

2.3.2 Preserve the Positivity of a Solution

One of the main differences between the fourth order parabolic equations and the second order equations is that for the fourth order equations the nonnegativity of the initial data does not imply the same property of solutions at any time. For instance, the linear equation

$$u_t + u_{xxxx} = 0 \tag{2.18}$$

belongs to a large class of fourth order (linear and nonlinear) equations which do not preserve the sign of the initial data [1], [20].

However, this doesn't mean that positive solutions are impossible. For example, during the last ten years, it has been established that the thin film equation for

$$u_t + (u^n u_{xxx})_x = 0$$

for $n > 0$, unlike (2.18), preserves the nonnegativity of the initial data [7], [6].

The preservation of nonnegativity or positivity is not only challenging from an analytical point of view, but also the derivation of sign-preserving numerical schemes for fourth order equations is a field of intensive research. Even for strictly positive analytical solutions, if not carefully designed, a discretization scheme may still give negative solutions, causing unwanted numerical instabilities [4], [35].

2.3.3 Is the Mass Conserved?

Many fourth order evolutionary partial differential equations we have encountered so far evolve with patterns that preserve the mass. For instance, the thin film equations (2.1), the interface fluctuation equation, and the Cahn-Hilliard equation with the non-flux boundary condition

$$u_x = u_{xxx} = 0, \quad \text{at } x = 0, 1$$

all share the property of mass conservation.

As a matter of fact, when solving fourth order partial differential equations using the moving collocation method described in section 3.2, we always rewrite the equation in a conservative form as follows (even for the non-conservative case),

$$F(t, x, u, u_x, u_{xx}, u_{xxx}, u_t, u_{xt}, u_{xxt}, u_{xxxt}) = G_x(x, t, u, u_x, u_{xx}, u_{xxx}, u_t, u_{xt}, u_{xxt}, u_{xxxt}) \tag{2.19}$$

which satisfies the generalized conservation property

$$\int_a^b F dx = G|_{x=b} - G|_{x=a} \ .$$

Notice that the fourth order derivative doesn't appear as an argument in equation (2.19). An advantage of this conservative collocation lies in that it avoids computing the fourth order derivative directly for a fourth order problem. In general, for problems with singularity formation, each successive spatial derivative is considerably larger than the previous one. Thus the fourth order term will be represented with the worst error, and for the finite element method, it is only piecewise continuous. By reformulating the main equation into the conservative form, we avoid computing the fourth order derivative directly, and thus avoid the error caused by evaluating the fourth order derivatives.

Now a question arises naturally. How does this conservative form work? Since we are solving evolutionary partial differential equations on a moving grid, is the mass conserved for problems with mass conservation? Usually, an equation can be rewritten into different conservative forms. Will it affect the solution, and hence the mass?

For a problem which conserves the mass within the system and admits finite time blow-up solution, the answer to this question is especially important, since any small error in the computed mass may lead to considerably larger error in the solution and in turn affect the patterns we observe.

2.4 Summary

In the previous sections, we have presented several types of evolutionary partial differential equations of fourth order. Each of these problems involves some difficulties

and challenges when numerically solving these equations.

To summarize, these difficulties and challenges include preserving the nonnegativity of the solution in the thin film equation, preserving the total mass within the Cahn-Hilliard equation, handling the case where blow-up and fast-decay are present, removing the singularities in the degenerate fourth order partial differential equations as well as capturing the complex patterns in equations of gradient type. Many papers working on these issues can be found, and a list of models and references can be found in the recent book [43].

It is part of the aim of this thesis to address some of these issues and numerically check some of the theoretical results from a selection of papers.

Chapter 3

Moving Collocation Scheme for 4th Order PDEs

3.1 MMPDE Approach

3.1.1 Motivation

One of the most important considerations when solving partial differential equations having nontrivial solutions is the decision of how to automatically and stably chose a non-uniform mesh that suitably adapts to the solution behavior.

Adaptive mesh methods have been widely used for solving differential equations that involve large solution variations, such as shock waves, boundary layers, and contact surfaces [29]. It has been demonstrated that significant improvements in accuracy and efficiency can be gained by adapting mesh points so that they are concentrated about areas of large solution variation.

For time-dependent differential equations, adaptive methods can be divided into two categories, static methods and dynamic methods.

Static methods: The discrete solution and equation are initially defined on a given mesh. During the calculation, based on properties of a certain function that measures the goodness of the approximation, a new mesh that might have a different number of nodes from the old mesh is constructed. The solution is then interpolated from

the old mesh to the new mesh, and a new discrete approximation to the solution is defined on the new mesh. The redistribution of the old nodes, the addition of new nodes, and the interpolation of the dependent variables from the old mesh to the new mesh are done at a fixed time.

Dynamic methods (often called moving mesh methods): A mesh equation that involves nodes speed is employed to move a mesh having a fixed number of nodes in such a way that the nodes remain concentrated in regions of rapid variation of the solution. The mesh equation and the original differential equation are often solved simultaneously for the physical solution and the mesh. Unlike static methods, interpolation of dependent variables from the old mesh to the new mesh is unnecessary.

Generally, static methods are robust for problems where regions of rapid variation move with time; however, the continual readjustment usually tends to slow down the computation, making these methods inefficient. For the dynamic methods, while it turns out to be surprisingly difficult to derive consistently reliable moving mesh equations, if carefully chosen, very often, they can lead to efficient methods.

A satisfactory mesh equation should be simple, easy to program, and reasonably insensitive to the choice of its adjustable parameters [33]. Among moving mesh methods, the moving finite difference method of Dorfi and Drury [22] and the moving finite element method of K. Miller [39], [38] have aroused considerable interest. While in the moving finite difference method the moving mesh equation is obtained directly from an equi-distribution principle, in the moving finite element method the solution and mesh are obtained by a process closely associated with equi-distribution of one measure: the residual of the original equation written in finite element form.

In [34], several moving mesh partial differential equations (MMPDEs) based on the equidistribution principle are derived and studied both theoretically and numerically. Some of them are new while some of them are related to methods developed in [22] and other references therein. On the whole, these moving mesh equations are all related to the equidistribution idea described in subsection 3.1.2.

3.1.2 Equidistribution Principle and MMPDEs

The equidistribution idea, introduced by de Boor [5] and Dodson [21], is based upon the simple idea that if some measure of the error $M(x)$ is available, then a good choice for a mesh would be one in which the contributions to the error over the subintervals are equalized (or "distributed equally").

Mathematically, the goal of finding mesh functions [46] $X_i(t)_{i=2}^{N-1}$ or moving meshes

$$\Pi := \{x^L = X_0 < X_1(t) < \cdots < X_{N-1}(t) < X_N = x^R\} \tag{3.1}$$

which are equidistributing for all values of t means that

$$\int_{X_i(t)}^{X_{i+1}(t)} M(x,t)dx = \frac{1}{N}\int_{x^L}^{x^R} M(x,t)dx =: \frac{1}{N}\theta(t),\ i = 2,3,\cdots,N. \tag{3.2}$$

Define the error measure as

$$W_i(t) = \int_{X_i(t)}^{X_{i+1}(t)} M(x,t)dx \tag{3.3}$$

A node is said to attract other nodes when a measure of the truncation error at this point is larger than average, and a node is said to repel other nodes if the error measure is smaller than average.

Obviously, if all the nodes of a moving mesh attract or repel their neighboring nodes, the moving mesh can satisfy our needs to capture the large solution variations based on error equi-distribution.

Let x and ξ denote the physical and computational coordinates over the unit interval [0,1], respectively. we can express (3.2) in an integral form as

$$\int_0^{x(\xi,t)} M(\tilde{x},t)d\tilde{x} = \xi\theta(t) \tag{3.4}$$

where

$$\theta(t) = \int_0^1 M(\tilde{x},t)d\tilde{x}. \tag{3.5}$$

Differentiating (3.4) with respect to ξ once and twice will give us two differential forms of the equidistribution principle

$$M(x(\xi,t),t)\frac{\partial}{\partial\xi}x(\xi,t) = \theta(t) \tag{3.6}$$

and

$$\frac{\partial}{\partial \xi}\left\{M(x(\xi,t),t)\frac{\partial}{\partial \xi}x(\xi,t)\right\}=0. \tag{3.7}$$

Related to (3.6) and (3.7), various MMPDEs are derived in [34]. In MC4, we use the following two of them: MMPDE4, MMPDE6, and a smooth version of MMPDE4 derived in [32]:

$$\frac{\partial}{\partial \xi}\left(M\frac{\partial \dot{x}}{\partial \xi}\right)=-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right) \tag{3.8}$$

$$\frac{\partial^2 \dot{x}}{\partial \xi^2}=-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right) \tag{3.9}$$

$$\frac{\partial}{\partial \xi}\left\{\frac{\partial \dot{x}}{\partial \xi}\tilde{M}\right\}=-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left\{\frac{\partial x}{\partial \xi}\tilde{M}\right\} \tag{3.10}$$

where λ is a positive number and $\tilde{M}$ is defined to satisfy:

$$\tilde{M}-\lambda^{-2}\frac{\partial^2}{\partial \xi^2}\tilde{M}=M \tag{3.11}$$

and the boundary condition

$$\frac{\partial \tilde{M}}{\partial \xi}(0,t)=\frac{\partial \tilde{M}}{\partial \xi}(1,t). \tag{3.12}$$

Their discretizations are given, respectively, by [32]

$$\frac{M_{i+1}+M_i}{(1/N)^2}(\dot{x}_{i+1}-\dot{x}_i)-\frac{M_i+M_{i-1}}{(1/N)^2}(\dot{x}_i-\dot{x}_{i-1})=-\frac{E_i}{\tau} \tag{3.13}$$

$$\frac{1}{(1/N)^2}[\dot{x}_{i+1}-2\dot{x}_i+\dot{x}_{i-1}]=-\frac{E_i}{\tau} \tag{3.14}$$

and

$$\frac{z_{i+1/2}}{M_{i+1/2}}=\frac{z_{i-1/2}}{M_{i-1/2}} \tag{3.15}$$

where

$$E_i=\frac{M_{i+1}+M_i}{(1/N)^2}(x_{i+1}-x_i)-\frac{M_i+M_{i-1}}{(1/N)^2}(x_i-x_{i-1})$$

$$z_{i+1/2}=y_{i+1/2}-\frac{1}{\lambda^2 h^2}(y_{i+3/2}-2y_{i+1/2}+y_{i-1/2})$$

$$y_{i+1/2}=\frac{1/N}{\dot{x}_{i+1}-\dot{x}_i+\frac{1}{\tau}(x_{i+1}-x_i)}$$

$$M_{i+1/2} = M(x_{i+1/2}, t).$$

We also use

$$y_{-1/2} = y_{1/2}, \quad y_{N+1/2} = y_{N-1/2}, \quad x_0 = x_N = 0$$

to approximate the boundary conditions (3.12).

3.1.3 Monitor Functions

The key to the success of the described MMPDE approach of mesh movement is to define a proper monitor function. To perform mesh adaptation in the gradient direction of the physical solution $u(x,t)$, the most common choice is the arc-length monitor function

$$M = \sqrt{1 + u_x^2}.$$

Numerical experiments have shown that a monitor function based on the solution gradient is not always the best option and may fail in many cases [28]. In fact, depending on the distribution and magnitude of the solution gradient, thus on problems to be solved and numerical schemes used, the monitor function may over or under concentrate mesh points in regions of large solution gradient or errors. This will certainly make the underlying moving mesh method less robust.

A common remedy is to introduce a parameter (denoted by α) to control the intensity of the mesh adaptation [12], [13], [49]. Based on the equidistribution principle, Beckett and Mackenzie define the monitor function to be [12] :

$$M = 1 + \alpha |u_{xx}|^{1/m}, \quad \alpha = \frac{1}{\langle |u_{xx}|^{1/m} \rangle}, \tag{3.16}$$

where m is an integer and $\langle \cdot \rangle$ denotes the average over the domain. They apply an adaptive scheme with this monitor function to the finite difference solution of a singularity perturbed, two-point boundary value problem and obtain a uniform convergence rate.

It is worthwhile to mention that MC4 gives solutions that always have continuous third order derivatives, thus it becomes natural to choose a monitor function which

involves the second order derivative, while in MOVCOL the solution only has continuous first order derivative so such a choice may be misleading if the second order derivative in the monitor function is not carefully reevaluated.

For a partial differential equation with no blow-up, the monitor function defined in (3.16) works well. However, if the solution of a partial differential equation does blow up (which actually occurs very often among fourth order problems), it is almost impossible to control the error near the blow up points in a robust way with a-posteriori error control. In view of the self-similarity property of many partial differential equations of fourth order, it is very possible for us to preserve the geometrical structure. Under this circumstance, we choose a function of the solution and its derivatives instead of the error as the monitor function.

In order to preserve the geometrical structure of the dynamics, we will also have to use scale-invariant monitor functions. It is also why we use MMPDE6 instead of MMPDE4 when a scaling invariant function is desired.

If the PDE is invariant under the scaling

$$t' = \lambda t, \quad x' = \lambda^m x, \quad u' = \lambda^n u, \tag{3.17}$$

then for MMPDE6 to be also invariant under (3.17), we require

$$M(\lambda^m x, \lambda^n u, \lambda^{n-m} u_x, \lambda^{n-2m} u_{xx}, \lambda^{n-3m} u_{xxx}) = \lambda^{-1} M(x, u, u, u_{xx}, u_{xxx}). \tag{3.18}$$

For example, for equation (2.14) which is scaling invariant under the scaling (2.15), it has been shown that the solution of this equation blows up at finite time when $p = 2$. Suppose we are to choose a monitor function which involves only the solution u. It can be easily seen that for equation (2.14) with $p = 2$, the monitor function should be $M(u) = |u|$. To make it strictly positive, we may choose

$$M(u) = 1 + |u|. \tag{3.19}$$

Hence, we get an asymptotically scaling invariant monitor function. We call it asymptotically scaling invariant because the solution is large at the blow-up area, and we have $1 + |u| \approx |u|$.

Similarly, if we are going to choose a monitor function which involves u_x, we will have to choose the monitor function

$$M(u) = |u_x|^{4/5}.$$

Once again, we make it strictly positive by adding some constant and making it asymptotically scaling invariant, for example, we can set the monitor function to be

$$M(u) = 1 + |u_x|^{4/5}. \tag{3.20}$$

For blow-up problems, especially for problems with finite time blow-up at a small region, the above scaling invariant monitor function may also fail due to the mesh over-concentration on this small region.

One way to overcome this is to combine the scaling invariant idea with the idea of Beckett and Mackenzie when they define the monitor function as (3.16) so that we have almost half of the mesh points within the blow-up region and half of the mesh points outside the blow-up region. Thus instead of (3.19), we can use

$$M(u) = 1 + \frac{|u|}{\langle |u| \rangle}, \tag{3.21}$$

or instead of (3.20), we can use

$$M(u) = 1 + \frac{|u_x|^{4/5}}{\langle |u_x|^{4/5} \rangle}. \tag{3.22}$$

3.1.4 Smoothing

For most problems which involves large solution variations, the monitor function is fairly non-smooth in space. This usually will lead to an abruptly varying mesh and ultimately cause a deterioration in the convergence rate and an increase in the error [40]; thus, some kind of smoothing should be employed in the MMPDEs in order to make the mesh move more smoothly.

In both MOVCOL and MC4, the monitor smoothing technique is applied. The values of the smoothed monitor function $\tilde{M}$ at node i are defined by

$$\tilde{M}_i = \sqrt{\frac{\sum_{k=i-p}^{i+p} (M_k)^2 \left(\frac{\gamma}{1+\gamma}\right)^{|k-i|}}{\sum_{k=i-p}^{i+p} \left(\frac{\gamma}{1+\gamma}\right)^{|k-i|}}} \tag{3.23}$$

where p determines the range of smoothing. For example, the non-smooth case has $p = 0$, and the three-point smoothing has $p = 1$. The three-point smoothing is commonly used in adaptive methods and will result in a five-block-diagonal algebraic system. For a general p, the resulting algebraic system is (3+2p)-block-diagonal; thus the higher the value of p, the more cost needed to solve the nonlinear system, and the smoother the resulting mesh. Empirically, $p = 1, 2, 3$ usually give good results.

The smoothing parameter γ has been used by many authors and has a natural physical meaning – the "rigidity" of a grid [22]. It requires that the grid spacing should not change from one interval to the next by more than $\frac{\gamma}{1+\gamma}$ for the stability of the grid. The choice of γ is fairly insensitive [33], and generally it can be fixed. It is recommended that one choose γ to be $1 \leq \gamma \leq 2$.

One other most critical parameter in equi-distributing mesh moving strategies is the time smoothing parameter τ. It represents a timescale for forcing the mesh toward equidistribution and preventing the mesh from crossing [34]. By comparing (3.7) and (3.8) we can see that the term $-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial M}{\partial \xi}\right)$ is actually a correction term to make the mesh equidistribute the monitor function. If τ is very large, the grid will not move and cannot be adapted. If τ is extremely small and tends to zero, it is possible that we will have a crossing mesh. The optimal choice of τ is also related to the size and smoothness of the monitor function. This is especially true for explicit integration, where a small τ can result in mesh crossing and failures in integrations. For explicit integration methods, changing τ can result in a new evaluation of the Jacobian for the Newton iteration; hence τ should not change frequently and might be kept as a constant. The recommended range for τ in both MOVCOL and MC4 is $10^{-6} < \tau < 10^{-3}$.

3.2 Moving Collocation Code for Fourth Order PDEs

3.2.1 A Brief Description of the Code

The moving collocation code described in the following subsection 3.3.2 is basically based on the existing code MOVCOL. Developed by Huang and Russell in 1996, MOVCOL is primarily intended to solve system of one dimensional second order PDEs. It uses a method of lines approach, in which the physical PDEs are discretized in space with MOVing COLlocation method [31], the moving mesh points are computed based upon MMPDEs [34], and the resulting ODE system is integrated in time with the DAE solver developed by Linda Petzold [41].

Dr. Jeff Williams from the University of Bath first extended the idea of MOVCOL to solve system of one dimensional fourth order PDEs. He also wrote two codes in MATLAB, one using the FORTRAN code DDASSL for the time integration via a MEX interface, and the other the MATLAB routine ode15i.m.

Due to the nature and breadth of the problems considered, we are solving higher order problems on a non-uniform grid. The recent work of Saucez et al. shows that great care has to be taken to reliably solve for the high-derivatives on a non-uniform grid. In [50] they used a finite difference method and the same adaptive strategy we use in our approach and found out that large number of nodes (e.g. 1000) were required to solve the problems

$$u_t + (u^m)_{xx} + (u^n)_{xxx} = 0,$$

$$u_{tt} - u_{xx} + u_{xxxx} + (u^2)_{xx} = 0,$$

$$u_t + 10uu_{xxx} + 25u_xu_{xx} + 20u^2u_x + u_{xxxx} = 0.$$

When actually running the codes written in MATLAB to solve a fourth order evolutionary partial differential equation with a compact scheme which has a large number of mesh points, we found out that the codes run extremely slowly. Experience showed that only several simple equations with few mesh points (like 40) could be solved. The code got stuck on hard problems.

Designed as a tool for doing numerical computations with matrices and vectors, Matlab is also powerful in displaying information graphically. It is also easy to code a problem up in Matlab. However, it certainly slows the computation down especially for our purpose. It is also expensive for Matlab to come in and out of the Mex interface between Matlab and DDASSL. In view of these facts, I start to write a FORTRAN code based on MOVCOL with considerable assistance from Dr. Williams.

The main subroutine which implements the moving collocation method is called MC4. The code solves the fourth-order parabolic PDEs of the general divergence form

$$F(t,x,u,u_x,u_{xx},u_{xxx},u_t,u_{xt},u_{xxt},u_{xxxt}) = \frac{\partial}{\partial x}G(t,x,u,u_x,u_{xx},u_{xxx},u_{xt},u_{xxt},u_{xxxt}) \tag{3.24}$$

for $x^L < x < x^R$ and $t_a < t \leq t_b$, supplemented with the initial condition

$$u(x,t_a) = U(x), \quad x^L \leq x \leq x^R \tag{3.25}$$

and four suitable boundary conditions. For the time being, we only consider problems with two boundary conditions at the left end of the interval and two boundary conditions at the right.

The equation is assumed to be well-posed, that is, the existence and uniqueness of the solution is guaranteed by the user.

The extension of MOVCOL, MC4, also uses a method of lines approach. The PDEs are discretized in space with a moving collocation method similar to the method described in [31], and the moving mesh points are computed based on MMPDEs [34] exactly the same way as in MOVCOL. For most of the problems, we integrate the resulting ODE system in time with DDASSL. A new ODE solver DASPK.f developed by Shengtai Li and Linda Petzold in 2001 has also been tried.

It should be noted that, in theory, MOVCOL can also solve partial differential equations of fourth order by converting them into systems of equations of second order. However, there are some concerns such as the fact that we will always have continuous third order derivatives when solving the equation by MC4, but with MOVCOL we would not. Other concerns are related to efficiency. In general, we have encountered many fourth order problems which when converted into second order system of equations and solved with MOVCOL, the code can't start or only with great difficulty.

One guess for why relates to the following fact: Suppose we are trying to solve the fourth order evolutionary partial differential equation

$$u_t = u_{xxxx} + u^2. \tag{3.26}$$

By setting $v = u_{xx}$, we rewrite equation (3.26) into

$$\begin{aligned} u_t &= v_{xx} + u^2 \\ v_t &= (u_{xt})_x. \end{aligned} \tag{3.27}$$

Notice that the second equation in (3.27) is actually $v = u_{xx}$, which is an algebraic equation rather than a differential equation. For our experience, this makes the system approach more difficult to start and integrate.

3.2.2 Moving Collocation Method

Hermite Interpolating Polynomials

One criterion for measuring the smoothness of a curve is the continuity of its derivatives. Perceptually smooth motion typically has at least two continuous derivatives, often more. Cubic spline curves, defined to interpolate both the value of a function and the value of its first derivative at given data point, generally only give solutions continuous in their first derivatives.

To obtain continuous higher derivatives, we consider using a Hermite interpolant of higher order. Suppose that our aim is to develop an interpolating polynomial which interpolates the function and its derivatives up to pth order at $N+1$ data points. We require that

$$\begin{aligned} g(x_i) &= f_i, \quad i = 0, 1, \cdots, N \quad \texttt{N+1 constraints} \\ g^{(1)}(x_i) &= f_i^{(1)}, \quad i = 0, 1, \cdots, N \quad \texttt{N+1 constraints} \\ &\vdots \\ g^{(p)}(x_i) &= f_i^{(p)}, \quad i = 0, 1, \cdots, N \quad \texttt{N+1 constraints.} \end{aligned}$$

We have a total of $(p+1)(N+1)$ constraints. Since the number of constraints must equal the number of unknowns in the interpolating polynomial, we need to set up a

general polynomial which is of degree $(p+1)(N+1)-1$

$$g(x) = \sum_{j=0}^{(p+1)(N+1)-1} a_j x^j.$$

The procedure to develop Hermite interpolation can be summarized as:

✓Set up the form for the interpolating polynomial

✓Implement constraints

✓Solve for unknown coefficients, $a_i, i = 0, \cdots, (p+1)(N+1)-1$.

Suppose we want to develop the Hermite interpolating polynomial which interpolates the function and its first, second and third derivatives for the interval [0, 1]. We have $p = 3$ and $N = 2$, so $(p+1)(N+1)-1 = 7$. Let

$$g(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4 + a_5 s^5 + a_6 s^6 + a_7 s^7.$$

Applying the constraints, we have

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}}_{a} = \underbrace{\begin{pmatrix} f_0 \\ f_1 \\ f_0^{(1)} \\ f_1^{(1)} \\ f_0^{(2)} \\ f_1^{(2)} \\ f_0^{(3)} \\ f_1^{(3)} \end{pmatrix}}_{f}.$$

Denoting $S = (1 \; s \; s^2 \; s^3 \; s^4 \; s^5 \; s^6 \; s^7)$, then

$$\begin{aligned} g(s) &= Sa = SA^{-1}f \\ &= \Big(\phi_{0,0}(s) \;\; \phi_{1,0}(s) \;\; \phi_{0,1}(s) \;\; \phi_{1,1}(s) \;\; \phi_{0,2}(s) \;\; \phi_{1,2}(s) \;\; \phi_{0,3}(s) \;\; \phi_{1,3}(s)\Big) f \end{aligned} \tag{3.28}$$

where

$$\begin{aligned}
\phi_{0,0}(s) &= 1 - 35s^4 + 84s^5 - 70s^6 + 20s^7, \\
\phi_{0,1}(s) &= s - 20s^4 + 45s^5 - 36s^6 + 10s^7, \\
\phi_{0,2}(s) &= 1/2s^2 - 5s^4 + 10s^5 - 15/2s^6 + 2s^7, \\
\phi_{0,3}(s) &= 1/6s^3 - 2/3s^4 + s^5 - 2/3s^6 + 1/6s^7, \\
\phi_{1,0}(s) &= 35s^4 - 84s^5 + 70s^6 - 20s^7, \\
\phi_{1,1}(s) &= -15s^4 + 39s^5 - 34s^6 + 10s^7, \\
\phi_{1,2}(s) &= 5/2s^4 - 7s^5 + 13/2s^6 - 2s^7, \\
\phi_{1,3}(s) &= -1/6s^4 + 1/2s^5 - 1/2s^6 + 1/6s^7.
\end{aligned} \tag{3.29}$$

Each element $\phi_{i,j}(s)$ of the above matrix SA^{-1} is a polynomial of order 7. Those polynomials are called the septic Hermite basis functions.

The polynomial defined in (3.28) interpolate a function and its first, second and third derivatives; thus, the interpolating curves should have a continuous third derivative.

The collocation discretization of the physical PDEs then follows naturally. Suppose that at time $t \in [t_a, t_b]$, a mesh with a fixed number $N+1$ of nodes

$$x^L = X_0(t) < X_1(t) < \cdots < X_{N-1}(t) < X_N(t) = x^R \tag{3.30}$$

is given in the interval $[x^L, x^R]$.

According to (3.28), the physical solution $u(x,t)$ can be approximated on mesh (3.30) with the piecewise septic Hermite polynomials derived above by

$$\begin{aligned}
v(x,t) \;=\;& v_i(t)\phi_{0,0}(s^{(i)}) + v_{x,i}(t)H_i(t)\phi_{0,1}(s^{(i)}) \\
&+ v_{xx,i}(t)H_i^2(t)\phi_{0,2}(s^{(i)}) + v_{xxx,i}(t)H_i^3(t)\phi_{0,3}(s^{(i)}) \\
&+ v_{i+1}(t)\phi_{1,0}(s^{(i)}) + v_{x,i+1}(t)H_i(t)\phi_{1,1}(s^{(i)}) \\
&+ v_{xx,i+1}(t)H_i^2(t)\phi_{1,2}(s^{(i)}) + v_{xxx,i+1}(t)H_i^3(t)\phi_{1,3}(s^{(i)})
\end{aligned} \tag{3.31}$$

for $x \in [X_i(t), X_{i+1}(t)]$, $i = 1, 2, \cdots, N-1$, where $v_i(t)$, $v_{x,i}(t)$, $v_{xx,i}(t)$ and $v_{xxx,i}(t)$ denote the approximations to $u(X_i(t),t)$, $u_x(X_i(t),t)$, $u_{xx}(X_i(t),t)$ and $u_{xxx}(X_i(t),t)$ respectively. The local coordinate $s^{(i)}$ is defined by

$$s^{(i)} := (x - X_i(t))/H_i(t), \quad H_i(t) := X_{(i+1)}(t) - X_i(t). \tag{3.32}$$

For $x \in [X_i(t), X_{i+1}(t)]$, $i = 1, 2, \cdots, N-1$, we have

$$
\begin{aligned}
v_x(x,t) &= (1/H_i)\left(v_i \frac{d\phi_{0,0}}{ds} + v_{x,i} H_i \frac{d\phi_{0,1}}{ds} + v_{xx,i} H_i^2 \frac{d\phi_{0,2}}{ds} + v_{xxx,i} H_i^3 \frac{d\phi_{0,3}}{ds} \right. \\
&\quad \left. + v_{i+1} \frac{d\phi_{1,0}}{ds} + v_{x,i+1} H_i \frac{d\phi_{1,1}}{ds} + v_{xx,i} H_i^2 \frac{d\phi_{1,2}}{ds} + v_{xxx,i+1} H_i^3 \frac{d\phi_{1,3}}{ds} \right) \\
v_{xx}(x,t) &= (1/H_i^2)\left(v_i \frac{d^2\phi_{0,0}}{ds} + v_{x,i} H_i \frac{d^2\phi_{0,1}}{ds} + v_{xx,i} H_i^2 \frac{d^2\phi_{0,2}}{ds} + v_{xxx,i} H_i^3 \frac{d^2\phi_{0,3}}{ds} \right. \\
&\quad \left. + v_{i+1} \frac{d^2\phi_{1,0}}{ds} + v_{x,i+1} H_i \frac{d^2\phi_{1,1}}{ds} + v_{xx,i} H_i^2 \frac{d^2\phi_{1,2}}{ds} + v_{xxx,i+1} H_i^3 \frac{d^2\phi_{1,3}}{ds} \right) \\
v_{xxx}(x,t) &= (1/H_i^3)\left(v_i \frac{d^3\phi_{0,0}}{ds} + v_{x,i} H_i \frac{d^3\phi_{0,1}}{ds} + v_{xx,i} H_i^2 \frac{d^3\phi_{0,2}}{ds} + v_{xxx,i} H_i^3 \frac{d^3\phi_{0,3}}{ds} \right. \\
&\quad \left. + v_{i+1} \frac{d^3\phi_{1,0}}{ds} + v_{x,i+1} H_i \frac{d^3\phi_{1,1}}{ds} + v_{xx,i} H_i^2 \frac{d^3\phi_{1,2}}{ds} + v_{xxx,i+1} H_i^3 \frac{d^3\phi_{1,3}}{ds} \right) \\
v_{xxxx}(x,t) &= (1/H_i^4)\left(v_i \frac{d^4\phi_{0,0}}{ds} + v_{x,i} H_i \frac{d^4\phi_{0,1}}{ds} + v_{xx,i} H_i^2 \frac{d^4\phi_{0,2}}{ds} + v_{xxx,i} H_i^3 \frac{d^4\phi_{0,3}}{ds} \right. \\
&\quad \left. + v_{i+1} \frac{d^4\phi_{1,0}}{ds} + v_{x,i+1} H_i \frac{d^4\phi_{1,1}}{ds} + v_{xx,i} H_i^2 \frac{d^4\phi_{1,2}}{ds} + v_{xxx,i+1} H_i^3 \frac{d^4\phi_{1,3}}{ds} \right)
\end{aligned}
$$

$$
\begin{aligned}
v_t(x,t) = {} & \frac{dv_i}{dt}\phi_{0,0} + \left(\frac{dv_{x,i}}{dt} H_i + v_{x,i} \frac{dH_i}{dt} \right) \phi_{0,1} + \left(\frac{dv_{xx,i}}{dt} H_i^2 + 2 v_{xx,i} H_i \frac{dH_i}{dt} \right) \phi_{0,2} \\
& + \left(\frac{dv_{xxx,i}}{dt} H_i^3 + 3 v_{xxx,i} H_i^2 \frac{dH_i}{dt} \right) \phi_{0,3} + \frac{dv_{i+1}}{dt}\phi_{1,0} \\
& + \left(\frac{dv_{x,i+1}}{dt} H_i + v_{x,i+1} \frac{dH_i}{dt} \right) \phi_{1,1} + \left(\frac{dv_{xx,i+1}}{dt} H_i^2 + 2 v_{xx,i+1} H_i \frac{dH_i}{dt} \right) \phi_{1,2} \\
& + \left(\frac{dv_{xxx,i+1}}{dt} H_i^3 + 3 v_{xxx,i+1} H_i^2 \frac{dH_i}{dt} \right) \phi_{1,3} - v_x(x,t) \left(\frac{dX_i}{dt} + s^{(i)} \frac{dH_i}{dt} \right)
\end{aligned} \tag{3.33}
$$

where $\phi_{m,n}$, $d^k\phi_{m,n}/d^k s$, $m = 0, 1$; $n = 0, 1, 2, 3$; $k = 1, 2, 3, 4$, are functions of $s^{(i)}$.

With these approximations, we can now write down a system of ordinary differential equations for the unknowns $v_i(t)$, $v_{x,i}(t)$, $v_{xx,i}(t)$, $v_{xxx,i}(t)$.

Notice that the physical PDE in the divergence form (3.24) satisfies the conservation law

$$
\int_{x^L}^{x^R} F dx = G|_{x=x^R} - G|_{x=x^L}. \tag{3.34}
$$

It will be natural to require the collocation scheme to satisfy an analogous discrete conservation law.

An advantage of conservative collocation lies in that it avoids computing the fourth order derivative directly for a fourth order problem. This is especially desirable for

problems with singularity formation, which generally means each successive derivative is considerably larger than the previous one and thus may be represented with large error. Another consideration is that we are using septic Hermite polynomials, so in general, the fourth order derivative is only piecewise continuous when solving the problem on a moving grid, and this also generates errors.

We approximate F by its Lagrange interpolant

$$\begin{aligned} F = \; & F(X_{i1},t)\frac{(x-X_{i2})(x-X_{i3})(x-X_{i4})}{(X_{i1}-X_{i2})(X_{i1}-X_{i3})(X_{i1}-X_{i4})} \\ & +F(X_{i2},t)\frac{(x-X_{i1})(x-X_{i3})(x-X_{i4})}{(X_{i2}-X_{i1})(X_{i2}-X_{i3})(X_{i2}-X_{i4})} \\ & +F(X_{i3},t)\frac{(x-X_{i1})(x-X_{i2})(x-X_{i4})}{(X_{31}-X_{i1})(X_{i3}-X_{i2})(X_{i3}-X_{i4})} \\ & +F(X_{i4},t)\frac{(x-X_{i1})(x-X_{i2})(x-X_{i3})}{(X_{i4}-X_{i1})(X_{i4}-X_{i2})(X_{i4}-X_{i3})} \end{aligned} \tag{3.35}$$

where $X_{ij} = X_i + s_j H_i (j = 1,2,3,4)$,

$$\begin{aligned} & s_1 = (1+\lambda_1)/2, \quad s_2 = (1+\lambda_2)/2 \\ & s_3 = 1 - s_2, \qquad\quad s_4 = 1 - s_1 \end{aligned} \tag{3.36}$$

denote the four Gauss points on [0,1], and $\lambda_i, i = 1,2$ are the two roots of the fourth order Legendre polynomial in [0,1]

$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3). \tag{3.37}$$

The right-hand side of equation (3.24) is integrated between the five Lobatto points on each subinterval

$$\begin{aligned} I_1 &= \int_{H_i \times l_1}^{H_i \times l_2} F dx = G_2 - G_1 \\ I_2 &= \int_{H_i \times l_2}^{H_i \times l_3} F dx = G_3 - G_2 \\ I_3 &= \int_{H_i \times l_3}^{H_i \times l_4} F dx = G_4 - G_3 \\ I_4 &= \int_{H_i \times l_4}^{H_i \times l_5} F dx = G_5 - G_4 \end{aligned} \tag{3.38}$$

where

$$\begin{aligned} G_j &:= G(t,x,v,v_x,v_{xx},v_{xxx},v_t,v_{xt},v_{xxt},v_{xxxt})|_{x=L_j} \\ G_{j+1} &= G(t,x,v,v_x,v_{xx},v_{xxx},v_t,v_{xt},v_{xxt},v_{xxxt})|_{x=L_{j+1}} \\ L_j &= X_i + l_j \times H_i, \quad j = 1,2,3,4,5, \\ l_1 &= 0, \quad l_2 = (1+\gamma)/2, \quad l_3 = 1/2, \quad l_4 = 1-l_2, \quad l_5 = 1 \end{aligned} \tag{3.39}$$

denote the five Lobatto points on [0,1], and γ is the nonzero root of $P_4'(x)$ in [0,1]

$$P_4'(x) = \frac{1}{8}(4 \times 35x^3 - 2 \times 30x) = \frac{5}{2}(7x^3 - 3x). \tag{3.40}$$

Combining (3.35) and (3.38), we can obtain a system of equations for $F(X_{ij}(t))$ in terms of G_j for $i = 1,2,\cdots,N-1$. Since this system of equations have been obtained using a Galerkin approach, we can regard these equations as collocation equations for (3.24) but with special treatment for the term dG/dx. The resulting system and the method we use are referred as moving collocation.

For many problems, the initial solution has steep gradients which can't be resolved on a uniform mesh. In such a case, it is necessary to generate a corresponding initial equidistributed mesh. In this approach, such an equidistributed mesh is first generated by integrating the physical PDE

$$\begin{aligned} &\frac{\partial v}{\partial t} = U(x), 0 < t \leq 1 \\ &v(x,0) = 0 \end{aligned} \tag{3.41}$$

where $U(x)$ is the initial condition. The obtained mesh at $t = 1$ is then used as the initial mesh for solving the physical PDE (3.24)

3.3 Numerical Experiments

To demonstrate the efficacy of the moving collocation method described in section 3.2, we now proceed to present some numerical results for a selection of problems.

3.3.1 A Simple Test Problem with Exact Solution

We shall start with a simple test problem which has an exact solution. We take advantage of this fact and gain some intuitive idea about the equi-distribution principle

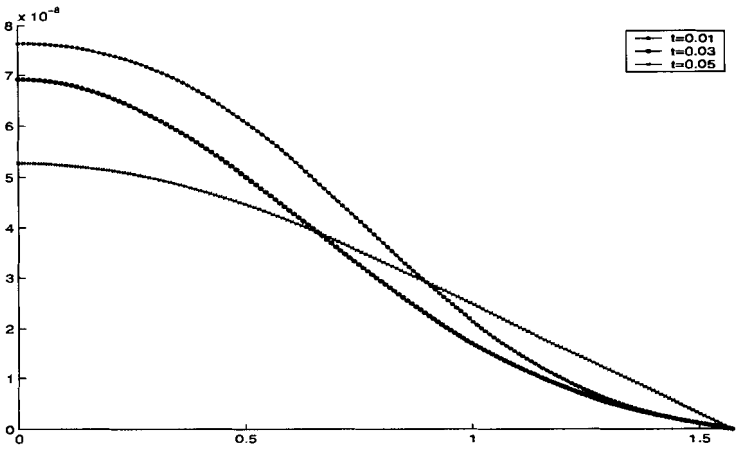

Figure 3.1: Problem I, Error in the solution: $|u - u_{\text{exact}}|$

$atol = 1 \times 10^{-7}$, $rtol = 1 \times 10^{-7}$, $npts = 160$, $MF = \sqrt{1 + u_x^2}$.

and property of the solution we obtained using MC4. We solve

$$\begin{cases} u_t = -u_{xxxx} - u \\ u(x,0) = \cos(x) \\ u_x(0;t) = u_{xxx}(0;t) = 0 \\ u(\pi/2;t) = u_{xx}(\pi/2;t) = 0. \end{cases} \tag{3.42}$$

The exact solution of this problem is $u(x,t) = e^{-2t}\cos(x)$.

Figure 3.1 shows that for atol=rtol=1×10^{-7}, the absolute error between the numerical solution and the exact solution is less than 8×10^{-8}. Thus the code gives the correct numerical solution.

In figure 3.1, *atol* and *rtol* stand for the absolute error tolerance and the relative error tolerance used in the ODE solver; *npts* stands for the number of points used when computed the solution and MF stands for the monitor function used in the moving mesh partial differential equation. We will use same notations for these parameters in other figures throughout this section.

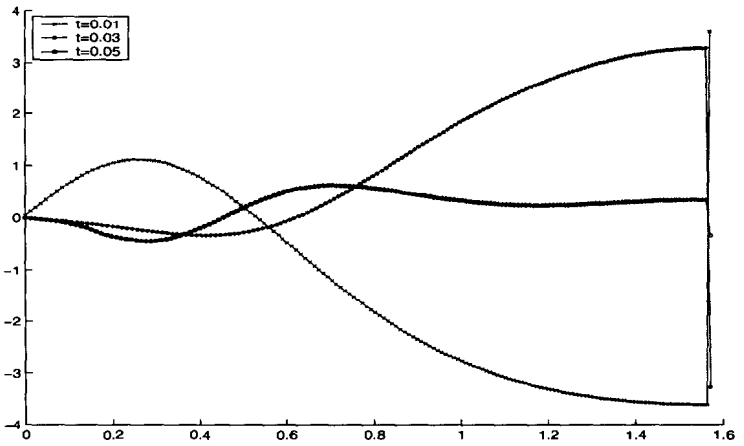

Figure 3.2: Problem I, The computed fourth derivative

$atol = 1 \times 10^{-7}$, $rtol = 1 \times 10^{-7}$, $npts = 160$, $MF = \sqrt{1 + u_x^2}$.

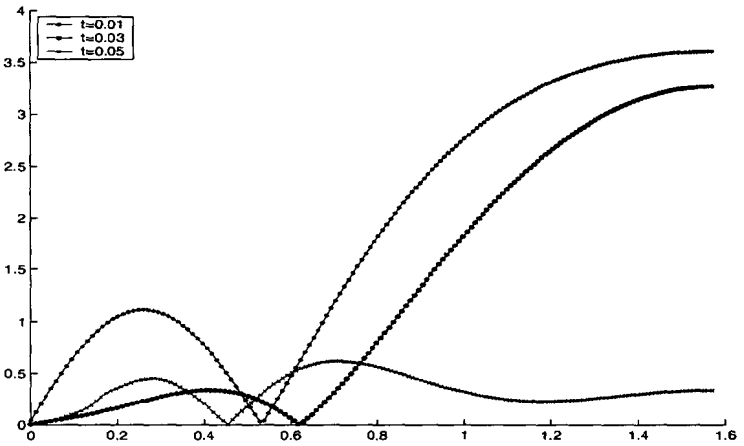

Figure 3.3: Problem I, Absolute error in the fourth derivative

$atol = 1 \times 10^{-7}$, $rtol = 1 \times 10^{-7}$, $npts = 160$, $MF = \sqrt{1 + u_x^2}$.

We have mentioned in section 3.1.3 that the solution always has third order continuous derivative. The fourth derivative is often only piecewise continuous. A natural question would be: how close is the approximation to the fourth order derivative?

Figure 3.2 shows the computed fourth order derivative. Here the fourth derivative is computed according to (3.33). Since the exact solution is $u = e^{-2t}\cos(x)$, the fourth derivative is simply $u_{xxxx} = e^{-2t}\cos(x)$. This is a continuous function and can be viewed as the function $\cos(x)$ shrinking as time goes on. However, from figure 3.3 we can see that the fourth derivative is only piecewise continuous. In particular, it has a jump discontinuity at the right boundary.

Figure 3.3 shows the absolute error in the computed fourth order derivative. As we can see, the error is quite large compared to the exact fourth derivative. It is also for this reason, in our collocation method, the fourth derivative doesn't appear explicitly. For problems in which finite singularities form, the fourth derivatives are very often extremely large. A wise choice would be avoiding evaluating this derivative.

3.3.2 A Coupled Equation System with Exact Solution

As in MOVCOL, MC4 aims at solving equation systems. Another test problem we used to test the performance of the code for equation systems is

$$\begin{cases} u_t = -v_{xxxx} \\ v_t = u_{xxxx} \\ u_x = u_{xxx} = 0, \quad x = 0, 2\pi \\ v_x = v_{xxx} = 0, \quad x = 0, 2\pi. \end{cases} \tag{3.43}$$

This coupled system has the exact solution

$$u = \cos(t+1)\cos(x), \quad v = \sin(t+1)\cos(x).$$

Figure 3.4 shows the solution to the coupled system. Figure 3.5 shows the error of the numerical solution. Once again, MC4 gives the correct solution.

Have seen that MC4 does solve the MMPDE and give correct answer, we now move on to present some of the interesting/challenging problems we have solved using MC4.

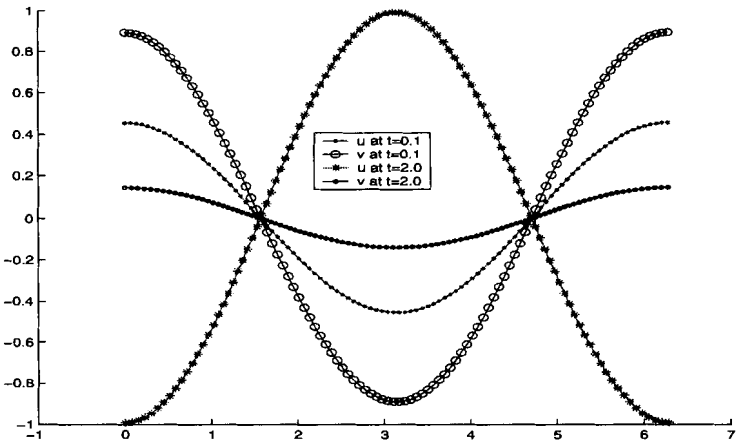

Figure 3.4: Problem II, Solution to the system

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 100$, $MF = \sqrt{1 + u_x^2}$.

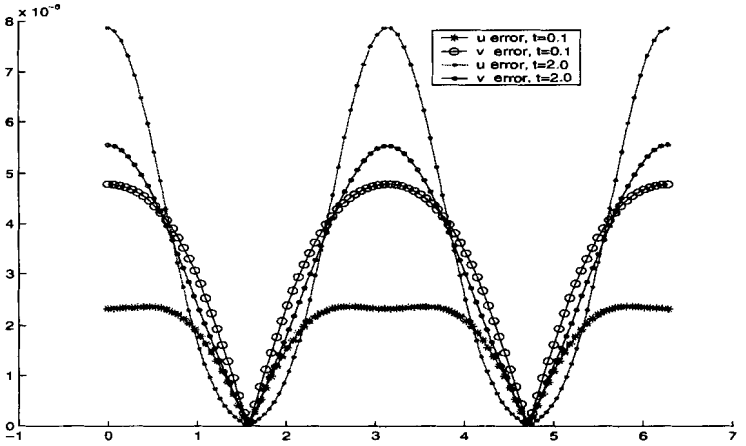

Figure 3.5: Problem II, Absolute error in the solution

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 100$, $MF = \sqrt{1 + u_x^2}$.

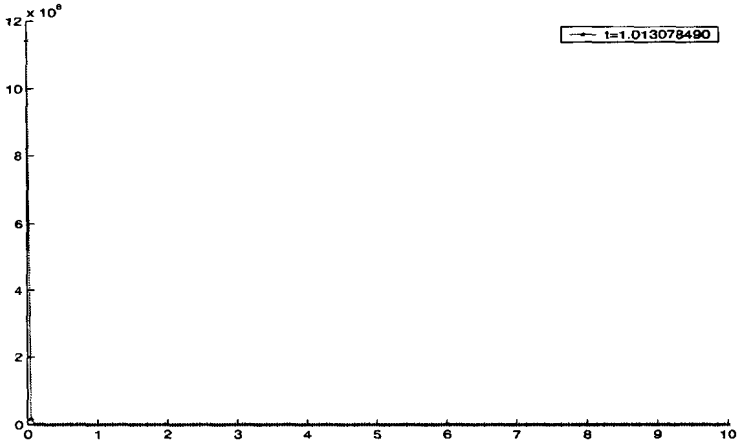

Figure 3.6: Problem III, Finite time single-point blow-up

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 200$, $MF = 1 + |u|/\langle u \rangle$.

3.3.3 A Problem with Finite Time Blow-Up

$$\begin{cases} u_t = u_{xxxx} + |u|^2 \\ u(0) = 2e^{-x^2} \\ u_x(0;t) = u_{xxx}(0;t) = 0 \\ u_x(10;t) = u_{xx}(10;t) = 0. \end{cases} \tag{3.44}$$

Theoretically, the solution to this equation blows up at the origin in a finite time $T \approx 1$. From section 2.2 we know that this equation has a self-similar solution of the form

$$u(x,t) = \frac{f(x/(T-t)^{1/4})}{T-t}.$$

It is shown in [8] that there exist at least 2 non-trivial self-similar solutions to equation (3.44) with the one having the simpler spatial shape corresponding to stable self-similar solution.

Figure 3.6 shows the solution at $t = 1.013078490$ where we can see clearly the single-point blow-up at the origin. From figure 3.6 we also observe that our code can

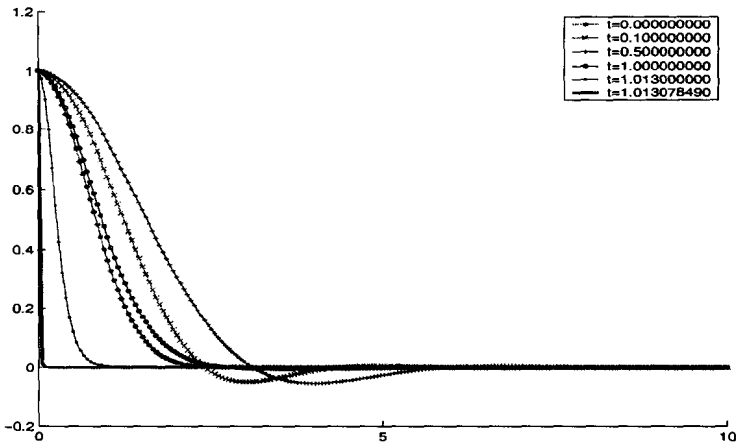

Figure 3.7: Problem III, Re-scaled solutions at different times

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 200$, $MF = 1 + |u|/\langle u \rangle$.

effectively handle sharp corners .

Figure 3.7 shows the re-scaled solutions at different times. In figure 3.7, we also observed two self-similar solutions. The one developed at earlier times is decreasing for small values of $x/(T-t)^{1/4}$ and increasing for larger values of $x/(T-t)^{1/4}$. The other one which appears to be a monotone decreasing function of $x/(T-t)^{1/4}$ corresponds to the stable self-similar solution obtained in [8]. The numerical solution confirmed theoretical results in [8].

3.3.4 A Problem with Finite Time Blow-up and Positivity

The following quasi-linear equation

$$u_t = u(-u_{xxxx} + u^2) \tag{3.45}$$

is studied in [52]. It is parabolic only for $u \geq 0$. This equation preserves the positivity of the solution.

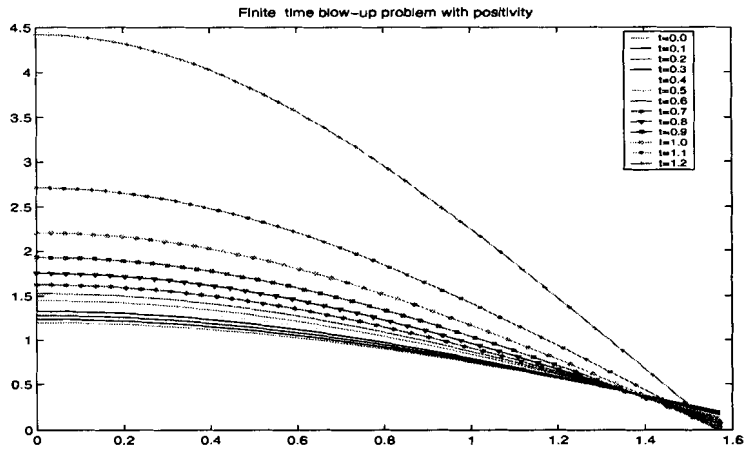

Figure 3.8: Problem IV, Solutions at different times, $a = 0.2$

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 40$, $MF = 1 + |u|$.

For our numerical experiments, we solve the equation

$$\begin{cases} u_t = u(-u_{xxxx} + u^2) \\ u(0) = a + \cos(x) \\ u_x(0;t) = u_{xxx}(0;t) = 0 \\ u_x(\pi/2;t) + u_{xxx}(\pi/2;t) = 0 \\ u_{xx}(\pi/2;t) = 0. \end{cases} \tag{3.46}$$

Theoretically, this problem may blow up or not according to the value of the constant a.

Figure 3.8 shows the solution of (3.46) when $a = 0.2$. Numerical results show that the solution starts to increase dramatically at about $t = 1$. The solution will be greater than 10^{10} before $t = 1.47$. Noticing that we are using the asymptotically scaling invariant monitor function $1 + |u|$, the mesh points concentrate on areas where the values of $|u|$ are large.

Figure 3.9 shows the solution of (3.46) when $a = 1$. Numerical results show that the solution starts to blow up at about $t = 0.8$. The solution will be greater than 10^{10}

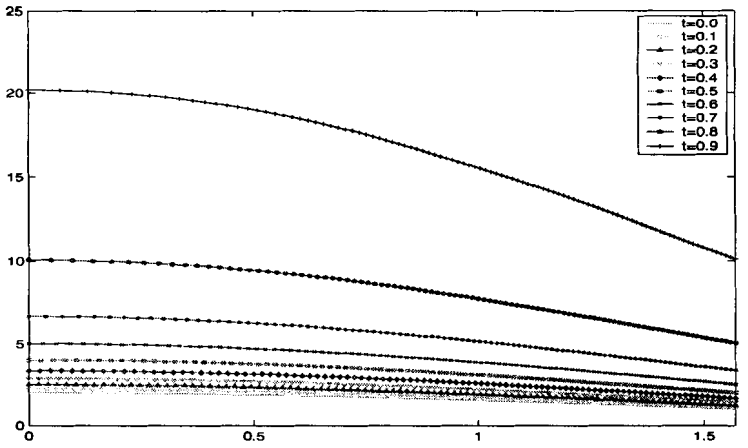

Figure 3.9: Problem IV, Solutions at different times, $a = 1.0$

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 100$, $MF = \sqrt{1 + u_x^2}$.

before $t = 1$. Noticing that we are using the arc-length monitor function $\sqrt{1 + u_x^2}$, the mesh points concentrate on areas where the values of $|u_x|$ are large.

For both cases, the code preserves the positivity of the solutions.

3.3.5 A Problem with Fast Decay

The following equation arises as a scaling limit in the study of interface fluctuations in a certain spin system [23] and also models the electron concentration in a quantum semiconductor device with zero temperature and negligible electric field [35]:

$$u_t = -(u(\log u)_{xx})_{xx} \tag{3.47}$$

with non-flux boundary condition

$$u_x(0,t) = u_x(1,t) = 0; \quad u_{xxx}(0,t) = u_{xxx}(1,t) = 0.$$

The initial periodic-boundary value problem for equation (3.47) was first studied by Bleher, Lebowitz and Speer in [11]. One can see [19] for the latest results.

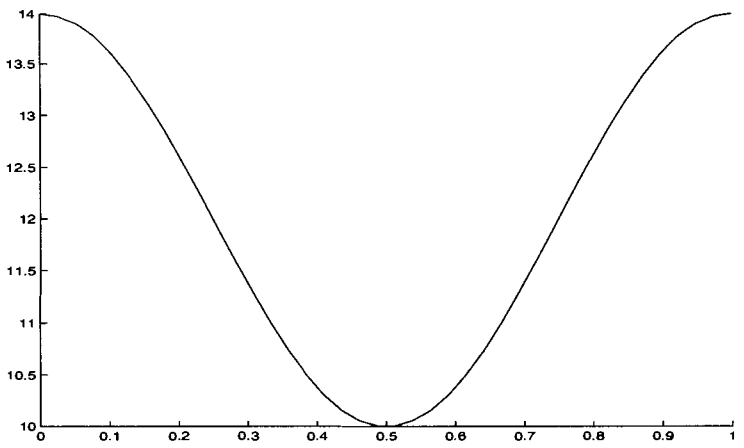

Figure 3.10: Problem V, The initial profile

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 40$, $MF = 1 + |u|$.

Equation (3.47) is a particular case of a class of fourth order diffusion equations which admit self-similar solutions. Using the definition given in section 2.2, one can easily show that (3.47) is invariant under the scaling

$$t' = \lambda t, \quad x' = \lambda^{1/4} x, \quad u' = u,$$

and hence has a self-similar solution of the form

$$u(x,t) = f(x/(T-t)^{1/4})$$

for a positive constant T such that $T - t > 0$. Theoretically, the solution shows an algebraic decay in the L^1 norm towards the constant steady state $\int_{[0,1]} u_0$, where u_0 is strictly positive initial data.

The initial condition is

$$w(x,0) = 12 + 2\cos(2\pi x).$$

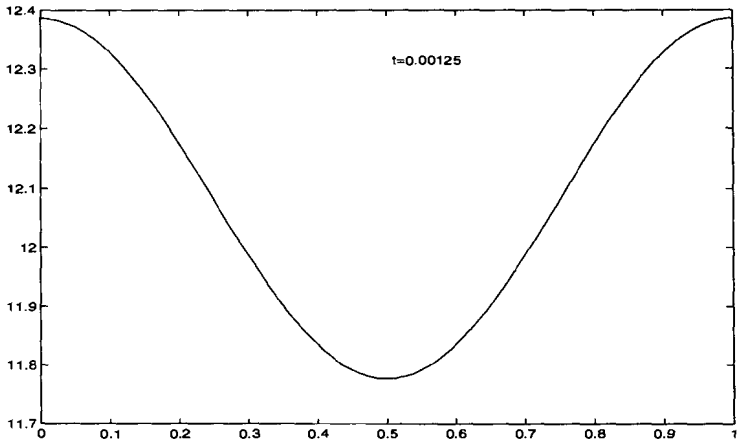

Figure 3.11: Problem V, Solution at t=0.00125

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 40$, $MF = 1 + |u|$.

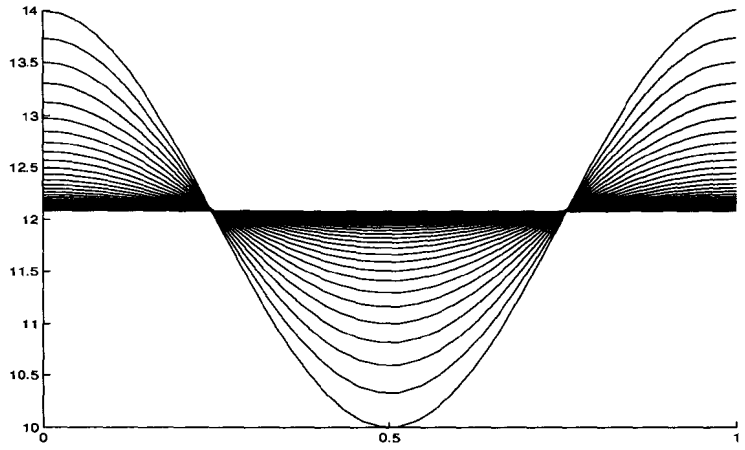

Figure 3.12: Problem V, Solutions at different times

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 40$, $MF = 1 + |u|$.

To compute the solution to (3.47), we set $u = w^2$ and use the equation

$$w_t = -w_{xxxx} + \frac{w_{xx}^2}{w}. \tag{3.48}$$

Figure 3.10 shows the initial profile. Figure 3.11 shows the solution at t=0.00125. By comparing figure 3.10 and figure 3.11 we can see the similarity of the solution to (3.48). Since the solution to (3.47) is simply the square of the solution to (3.48), the similarity of the solution to equation (3.47) follows naturally.

Figure 3.12 shows the solutions to equation (3.48) at different times. According to section 2.1, the solution should converge to the constant steady state

$$\left(\int_{[0,1]} w_0^2(x,0)dx\right)^{1/2} = 12.08304597,$$

whereas our computational solution converges to

$$u(x, t > 0.052) = 12.08304583.$$

As we can see the decay is fast.

3.3.6 Long Time Behavior of Cahn-Hilliard Equation

As indicated in chapter 2, a lot of diffusive processes, such as phase separation in binary alloys, growth and dispersal in population, spreading of oil film over a solid surface, can be described by the Cahn-Hilliard equation. A special case for this equation is

$$\begin{cases} u_t = -(\gamma_1 u_{xx} - \gamma_2 u^3 - \gamma_3 u^2 + u)_{xx} \\ u(x,0) = u_0(x) \\ u_x(0,t) = u_x(1,t) = 0 \\ u_{xxx}(0,t) = u_{xxx}(1,t) = 0\ . \end{cases} \tag{3.49}$$

It is proved in [53] that if $\gamma_2 < 0$, then for $u_0 \neq 0$, there exists a constant $\Gamma > 0$ depending only on u_0, such that for $\gamma_2 < -\Gamma$, the solution u of (3.49) blows up at a finite time; if $\gamma_2 > 0$, then for any initial value $u_0 \in H_E^2(0,1)$, (3.49) admits a unique

solution $u \in H^{4,1}(Q_T)$; if $\gamma_1 > \frac{1}{\pi^2}$, $\gamma_2 > 0$, $u_0 \in H_E^2(0,1)$, and $\|u_0\|_2$ is sufficiently small, then (3.49) admits a global solution $u \in H^{4,1}(Q_T)$, and

$$\lim_{t\to\infty} \|u(t) - M\|_\infty = \lim_{t\to\infty} \|Du(t)\|_\infty = \lim_{t\to\infty} \|D^2u(t)\|_\infty = 0; \tag{3.50}$$

where $\|\cdot\|$ denotes the norm in $H^2(0,1)$, $M = \int_0^1 u_0(x)dx$, and

$Q_T \equiv (0,1)\times(0,T)$, $\quad H_E^2(0,1) = \left\{v \in H^2(0,1); \frac{\partial v}{\partial x}|_{x=0,1} = 0|\right\}$,

$H^{4,1}(Q_T) = \left\{v; \frac{\partial v}{\partial t} \in L^2(Q_T), \frac{\partial^i v}{\partial x^i} \in L^2(Q_T), 0 \le i \le 4\right\}$.

For the numerical experiments in this section, we solve the equation

$$\begin{cases} u_t = -(\gamma_1 u_{xx} - \gamma_2 u^3 - \gamma_3 u^2 + u)_{xx} \\ u(x,0) = \cos(\pi x/k) \\ u_x(0,t) = u_x(6,t) = 0 \\ u_{xxx}(0,t) = u_{xxx}(6,t) = 0. \end{cases} \tag{3.51}$$

By choosing different values for γ_1, γ_2, γ_3 and k, we can numerically check these theoretical results in [53]. Noticing that those theoretical results have no requirement on γ_3, we use $\gamma_3 = 0$ for simplicity.

Type A When $\gamma_1 = 0.02$, $\gamma_2 = -1$, (3.51) represents a Cahn-Hilliard problem with finite time blow-up.

Figures 3.13-3.15 show the solution of (3.51) with $\gamma_1 = 0.02, \gamma_2 = -1, \gamma_3 = 0, k = 6$. From figure 3.14 we can see the solution blows up at two boundary points of the interval at about $t = 0.24$. Since $\gamma_2 = -1$, according to [53], for $\Gamma \ge 1$, (3.51) represents a Cahn-Hilliard problem with finite time blow-up.

Type B When $\gamma_1 = 0.02$, $\gamma_2 = 1$, (3.51) represents a Cahn-Hilliard problem with a metastable solution.

Figure 3.16 shows the solutions of (3.51) with $\gamma_1 = 0.02, \gamma_2 = 1, \gamma_3 = 0, k = 6$. Figures 3.17-3.18 show the solutions of (3.51) with $\gamma_1 = 0.02, \gamma_2 = 1, \gamma_3 = 0, k = 1$. Numerical results show that the solution first tends to a square wave of amplitude $\sqrt{3}$ and then evolves very slowly, hence becomes metastable. These results agree with the numerical results in [47].

Type C When $\gamma_1 = 1$, $\gamma_2 = 1$, (3.51) represents a Cahn-Hilliard problem with fast decay.

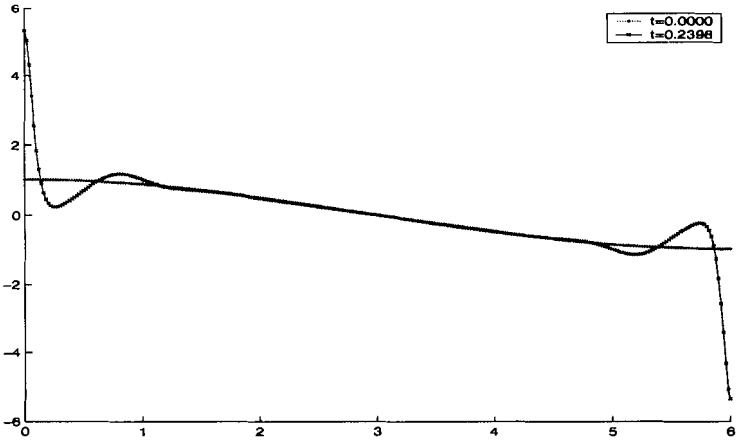

Figure 3.13: Problem VI-A, Solutions of the C-H Problem with finite time blow-up

$$\gamma_1 = 0.02,\ \gamma_2 = -1,\ \gamma_3 = 0,\ k = 6.$$
$$atol = 1 \times 10^{-8},\ rtol = 1 \times 10^{-8},\ npts = 301,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

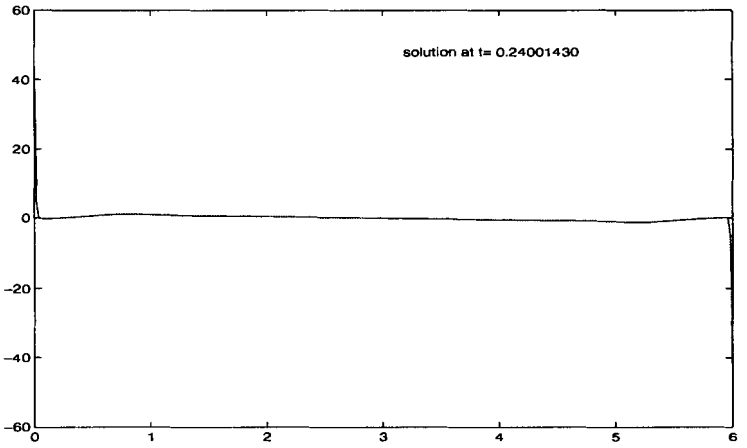

Figure 3.14: Problem VI-A, Finite time boundary blow-up

$$\gamma_1 = 0.02, \gamma_2 = -1, \gamma_3 = 0, k = 6.$$
$$atol = 1 \times 10^{-8},\ rtol = 1 \times 10^{-8},\ npts = 301,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

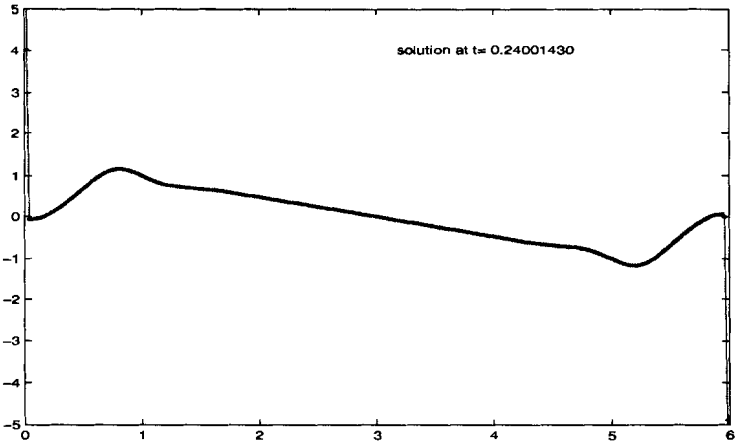

Figure 3.15: Problem VI-A, Enlargement of the non-blow-up part in figure 3.14

$$\gamma_1 = 0.02,\ \gamma_2 = -1,\ \gamma_3 = 0,\ k = 6.$$
$$atol = 1 \times 10^{-8},\ rtol = 1 \times 10^{-8},\ npts = 301,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

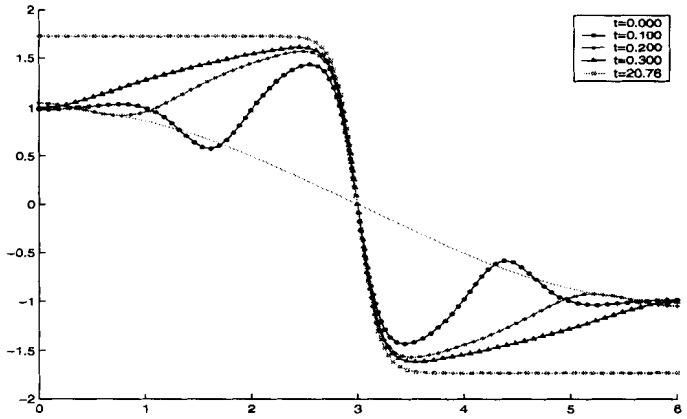

Figure 3.16: Problem VI-B, Solution at different times

$$\gamma_1 = 0.02,\ \gamma_2 = 1,\ \gamma_3 = 0,\ k = 6.$$
$$atol = 1 \times 10^{-6},\ rtol = 1 \times 10^{-6},\ npts = 101,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

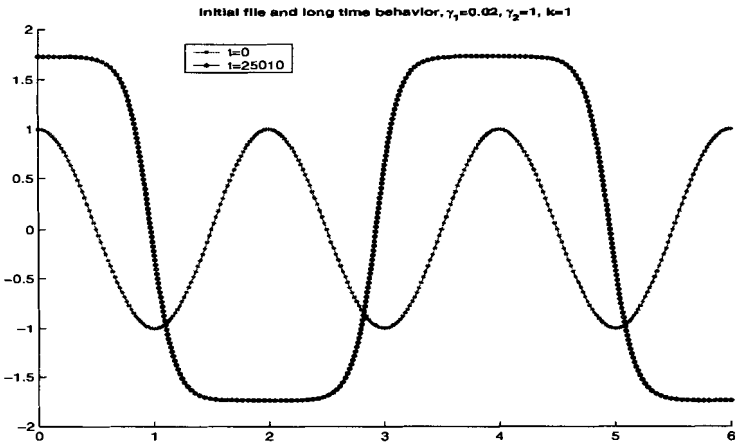

Figure 3.17: Problem VI-B, Solution at different times

$$\gamma_1 = 0.02,\ \gamma_2 = 1,\ \gamma_3 = 0,\ k = 1.$$
$$atol = 1 \times 10^{-6},\ rtol = 1 \times 10^{-6},\ npts = 301,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

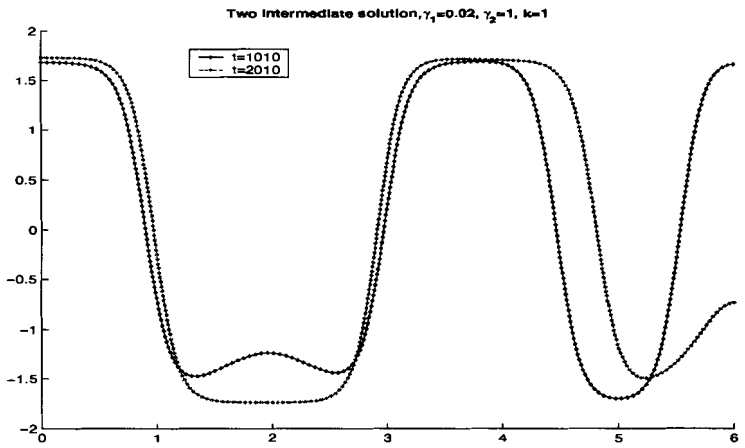

Figure 3.18: Problem VI-B, Solution at different times

$$\gamma_1 = 0.02,\ \gamma_2 = 1,\ \gamma_3 = 0,\ k = 1.$$
$$atol = 1 \times 10^{-6},\ rtol = 1 \times 10^{-6},\ npts = 301,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

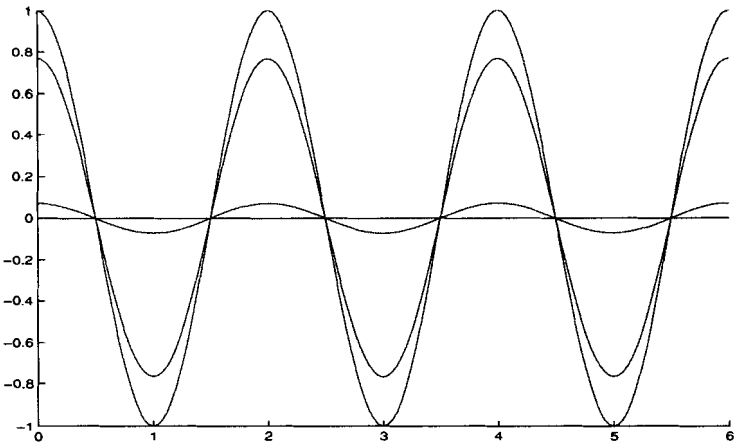

Figure 3.19: Problem VI-C, Solution at different times

$$\gamma_1 = 1,\ \gamma_2 = 1,\ \gamma_3 = 0,\ k = 1.$$
$$atol = 1 \times 10^{-6},\ rtol = 1 \times 10^{-6},\ npts = 201,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

Figure 3.19 shows the solutions of (3.51) with $\gamma_1 = 1, \gamma_2 = 1, \gamma_3 = 0, k = 1$. Numerical results show that the solution decays fast and converges to the constant steady state $M = \int_0^6 u_0(x)dx$. This confirms (3.50). Noticing that we are solving the equation on $[0, 6]$ instead of $[0, 1]$, we can rescale (3.51) into the form of (3.49) using $x' = x/6$ and $t' = 1/36$. One will find out using $\gamma_1 = 1$ in (3.51) is equivalent to using $\gamma_1 = \frac{1}{36}$ in (3.49). Notice that $\frac{1}{36} < \frac{1}{\pi^2}$. This implies that for $\gamma_1 < \frac{1}{\pi^2}$, it is still possible for (3.49) to admit a solution which decays fast.

Since this system conserves mass, the exact value of mass should be 0. Figure 3.20 and Figure 3.21 show how the mass changes with time. In figure 3.20, the error tolerance used when solving the equation is absolute error tolerance $= 1 \times 10^{-5}$ and relative error tolerance $= 1 \times 10^{-5}$. In figure 3.21, the error tolerance used when solving the equation is absolute error tolerance $= 1 \times 10^{-6}$ and relative error tolerance $= 1 \times 10^{-6}$. The time to output the solution is set to be tout(1)=0 tout(2)=1.d-12, tout(3)=1.d-11, $\cdots$, tout(13)=1.d-1, tout(14)=1, and then simply output the solution every unit of time. From figure 3.20 and figure 3.21 we can see that when the change

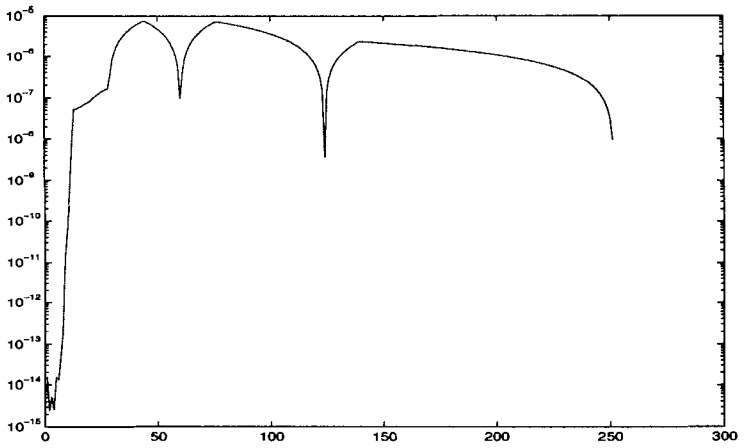

Figure 3.20: Problem VI-C, Error in mass in time

$$\gamma_1 = 1,\ \gamma_2 = 1,\ \gamma_3 = 0,\ k = 1.$$
$$atol = 1 \times 10^{-5},\ rtol = 1 \times 10^{-5},\ npts = 201,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

in time is small, the mass change is small, and the error in the mass is small. A good aspect of the result is the error is within the error tolerance, which means we can actually control the error at each time step.

Small change in time is related to a small change in the mesh points, this implies that when solving finite time blow-up problems or fast decay problems, a small time step size is preferred in order to control the error; this in turn will conserve the mass better in the long term so that it won't affect the patterns determined by the mass.

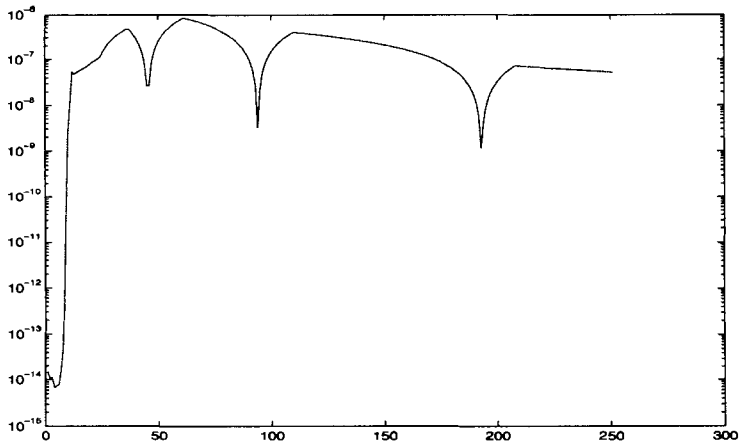

Figure 3.21: Problem VI-C, Error in mass in time

$$\gamma_1 = 1,\ \gamma_2 = 1,\ \gamma_3 = 0,\ k = 1.$$
$$atol = 1 \times 10^{-6},\ rtol = 1 \times 10^{-6},\ npts = 201,\ MF = \sqrt{1 + u_x^2/\langle u_x^2 \rangle}.$$

3.3.7 Moving Contact Lines in Thin Liquid Films

The following thin film equation is used in [54], [3], [4] to illustrate the effectiveness of using a positivity preserving (EDS) scheme over a generic one:

$$\begin{cases} u_t + (u^{1/2} u_{xxx})_x = 0 \\ u(x,0) = 0.8 - \cos(\pi x) + 0.25\cos(2\pi x) \\ u(-1,t) = u(1,t) = 2.05 \\ u_x(-1,t) = u_x(1,t) = 0\ . \end{cases} \tag{3.52}$$

It was computationally shown in [3] that the solution of this problem develops singularity in finite time. The solution develops a finite-time "pinching" singularity with a simultaneous blowup in the higher derivative. Near the pinch point the solution has a leading order asymptotic form

$$u(x,t) \approx c(t_c - t) + \frac{p(x - x_c)^2}{2} \tag{3.53}$$

where t_c is the time of pinch-off, x_c is the pinch point and the constant p is the curvature of the interface at the time of pinch.

The blow-up in higher derivatives can be seen in higher order terms

$$u_{xxx} \approx \frac{cx}{\sqrt{c(t_c - t) + p(x - x_c)^2/2}} \tag{3.54}$$

so that the local curvature u_{xx} has the form

$$u_{xx} \approx p + \frac{c}{p}\sqrt{c(t_c - t) + p(x - x_c)^2/2}. \tag{3.55}$$

The solution past the singularity time can be defined by introducing the regularization

$$\begin{aligned} u_{\epsilon t} &= (f_\epsilon(u_\epsilon)u_{\epsilon xxx})_x = 0 \\ f_\epsilon(u_\epsilon) &= \frac{u_\epsilon^4 f(u_\epsilon)}{\epsilon f(u_\epsilon) + u_\epsilon^4} \end{aligned}, \tag{3.56}$$

finding the solution of the regularized problem, and taking the regularization parameter ϵ to zero [14], [6], [7]. The resulting "weak solution" is guaranteed to be C^1 for almost every time and to have a bounded second derivative for almost every time. Since $f_\epsilon(u_\epsilon) \approx u_\epsilon^4/\epsilon$ as $u_\epsilon \to 0$ we know that $\forall \epsilon > 0$ the analytical solution of the regularized problem is positive.

The numerical solutions of the regularized equation in [4] suggest that after the initial singularity, the solution develops a region where it is identically zero. The regularized solution stays positive and develops a lot of structure near the edge of the support of the weak solution. In [4] a fine grid (1024 points) was required in order to resolve the spatial structure and keep the numerical solution positive.

Figures 3.22-3.23 show the solutions at specified times which are typical for this problem. In particular, figure 3.22 shows the initial profile($t = 0$)and the solution at the time of "pinch-off" singularity($t = 0.000729$). This "pinching" singularity can be observed only in the third derivative, see figure 3.26. From 3.23 we can see that the solution curve near $x = 0$ is concave up before the "pinch-off" time and is concave down after. Figure 3.23 also shows the solution at $t = 0.001$. By carefully comparing it with the solution of Andrea Bertozzi which is shown in figure 4-5 of [4], we conclude that the solutions are same. Notice that we are using a moving mesh and $\epsilon = 2.d-15$ while she is using a fixed mesh and $\epsilon = 1.d - 14$.

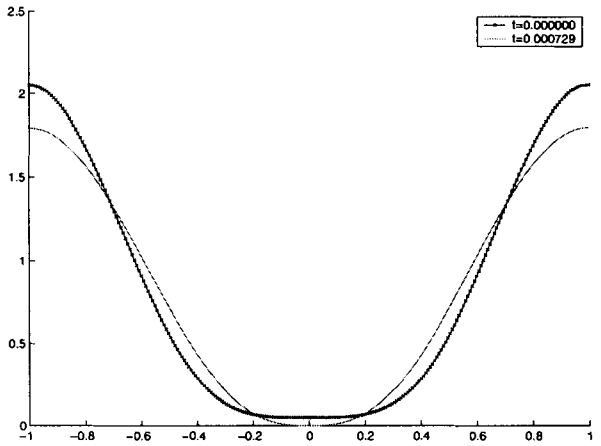

Figure 3.22: Problem VII, Solutions at specified times

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 251$.

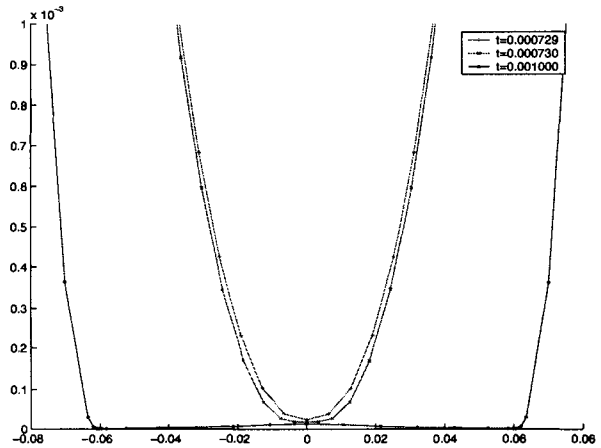

Figure 3.23: Problem VII, Solutions at specified times, enlargement

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 251$.

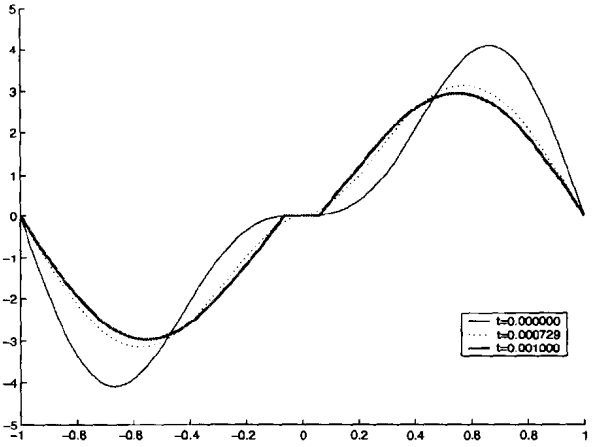

Figure 3.24: Problem VII, First derivative of solutions at specified times

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 251$.

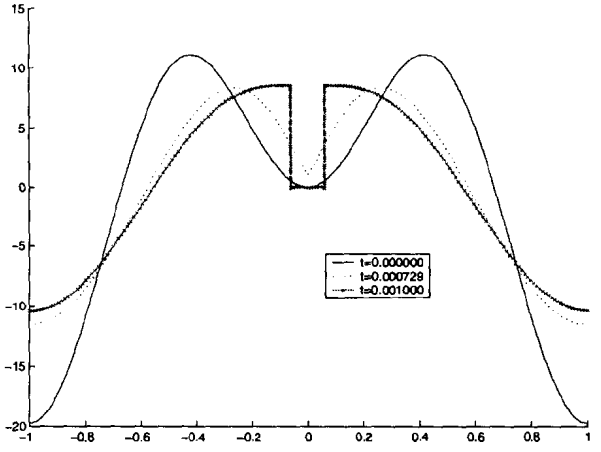

Figure 3.25: Problem VII, Second derivative of solutions at specified times

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 251$.

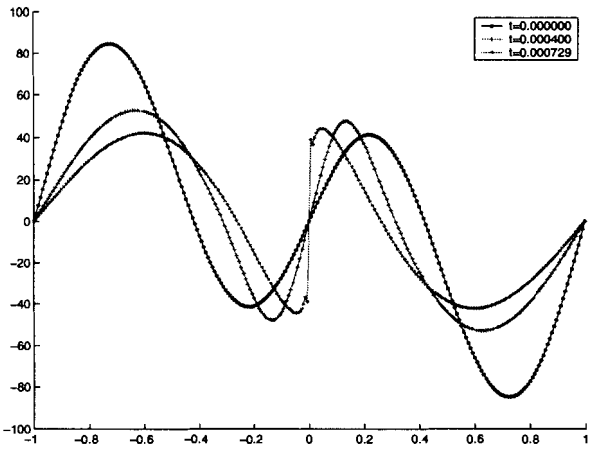

Figure 3.26: Problem VII, Third derivative of solutions at specified times

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 251$.

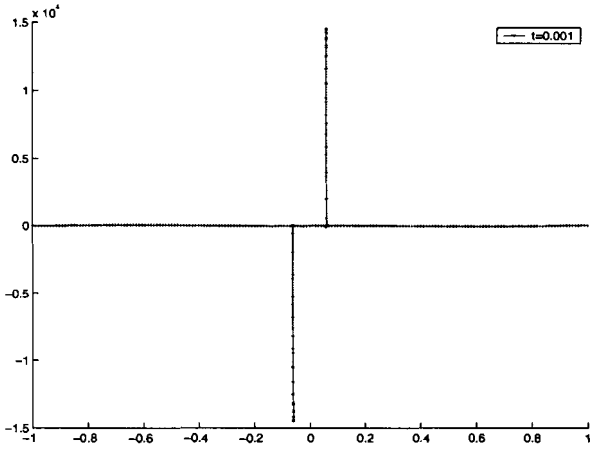

Figure 3.27: Problem VII, Third derivative of the solution at t=0.001

$atol = 1 \times 10^{-6}$, $rtol = 1 \times 10^{-6}$, $npts = 251$.

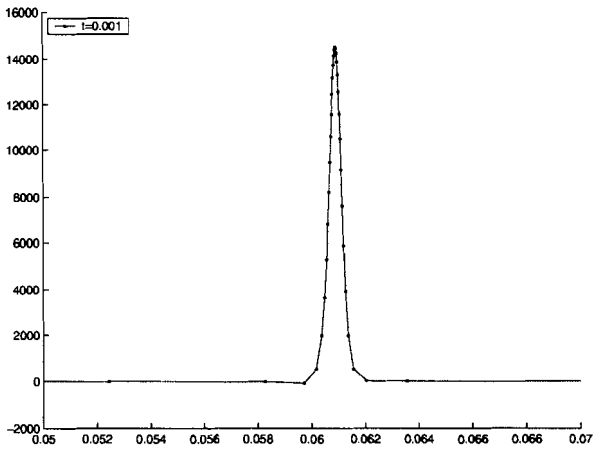

Figure 3.28: Problem VII, Third derivative of the solution at t=0.001, enlargement

$$atol = 1 \times 10^{-6},\ rtol = 1 \times 10^{-6},\ npts = 251.$$

Figure 3.24 shows the first derivative of the solutions at specified typical times. As above, $t = 0$ represents the first derivative of the function which serves as the initial condition; $t = 0.000729$ represents the derivative of the solution at the "pinch-off" time $t = 0.000729$. After the "pinch-off" time, the solution develops a region where it is identically zero. The way that this region develops highly depends on the value of ϵ as seen if we observe it closely. In figure 3.23 we observe that the solution is concave down for $\epsilon = 2.d - 15$. However, if $\epsilon = 0$, the solution should be concave up or be a straight line segment. For sufficiently small ϵ, the curvature is small, and we observe a straight line segment in the solution and its first derivative – see the solution and its first derivative at $t = 0.001$ in figures 3.22-3.24.

Figure 3.25 shows the second derivative of the solutions at specified typical times. This figure clearly shows where the "pinching" singularity happens and where the "zero-solution" region locates.

Figures 3.26-3.28 show the third derivative of the solutions at specified times. In figure 3.26, we can see clearly the "pinching" singularity at $t = 0.000729$. By carefully comparing this figure with figure 1 of [4], we can see that once again our result confirm

the result of [4].

In order to resolve the spatial structure developed in the third derivative at later times (see figure 3.27), we constructed the following monitor function:

$$MF = \begin{cases} 30, & \texttt{if } x < -0.4 \texttt{ or } x > 0.4 \\ 30, & \texttt{if } -0.4 < x < 0.4 \texttt{ and } |u_{xxx}| < 29 \\ 1 + |u_{xxx}|, & \texttt{otherwise} \end{cases} . \tag{3.57}$$

Since MMPDEs put more points where the values of the monitor function are larger, this monitor function forces the mesh points to concentrate in the neighborhood of the two singularities in the third derivative – see figure 3.28.

To compute the solution, we use $\epsilon = 2 \times 10^{-15}$. However we suggest to use successively smaller values of ϵ based on the following analysis.

To remove the singularity of a degenerate thin film problem, we use

$$f_\epsilon(u_\epsilon) = \frac{u_\epsilon^4 f(u_\epsilon)}{\epsilon f(u_\epsilon) + u_\epsilon^4} \tag{3.58}$$

to approximate $f(u)$, which is $u^{1/2}$ for this problem, where ϵ is a sufficiently small positive number acting as a regularization parameter. On the one hand, for large value of u, since $f_\epsilon(u_\epsilon) \approx f(u)$, this regularization parameter almost has no effect on the solution we obtain. On the other hand, since $f_\epsilon(u_\epsilon) \approx u_\epsilon^4/\epsilon$ as $u_\epsilon \to 0$, $\forall \epsilon > 0$ the analytical solution of the regularized problem is positive, hence we can remove the singularity.

Suppose at a certain time, $u_\epsilon = 10^{-4}$, and $\epsilon = 10^{-14}$, then

$$f_\epsilon(u_\epsilon) \approx u_\epsilon^4/\epsilon = \frac{1}{2} \times 10^{-2} \approx u_\epsilon^{1/2}.$$

Thus for $\epsilon < \frac{1}{2} \times 10^{-14}$, we can remove the singularity and still obtain a good approximation.

Now suppose at a certain time, $u_\epsilon = 10^{-6}$. If we still take $\epsilon = 10^{-14}$, then

$$f_\epsilon(u_\epsilon) \approx u_\epsilon^4/\epsilon \approx 10^{-10}$$

while $u_\epsilon^{1/2} = 10^{-3}$. We did remove the singularity; however, the speed at which the solution at this point approaches zero has been affected by a factor of 10^7. Since

the speed at which each point approaches zero highly depends on the value of ϵ, it is suggested that we choose ϵ according to the requirement of accuracy. For our computation, we aimed at the accuracy of 10^{-4}, thus we choose $\epsilon = 2 \times 10^{-15}$.

As indicated in [3], no rigorous results exist to describe the local structure of the singularity for this thin film problem, and there are also a number of open problems concerning the weak solutions. Thus, it is a good sign that our results confirm the results of Andrea Bertozzi et al. since we are using different methods.

Chapter 4

Conclusions and Further Work

4.1 Conclusions

In this thesis, we study the theory and computation of a moving collocation method for solving evolutionary partial differential equations of fourth order. We start by briefly discussing five types of fourth order evolutionary partial differential equations we have been investigating. Difficulties and challenges arising in solving these type of problems are also addressed. These difficulties and challenges include computing the solutions near a singularity time, preserving the positivity of the solution especially for degenerate equations, and preserving the mass, especially for finite time blow-up problems with mass conservation.

It is a challenging problem to compute solutions near the singularity time T. Even with implicit methods, one needs certain grid adaptation so as to have a more refined mesh near blow-up points. The preservation of nonnegativity or positivity is also a challenging problem to design a sign-preserving numerical schemes for fourth order equations. Even for strictly positive analytical solutions, if not carefully designed, a discretization scheme may still give negative solutions, causing unwanted numerical instabilities. In a mass conservation system, the behavior of the solution is very often determined by the mass within this system. Preserving the mass is especially important for a finite-time blow-up system with mass conservation since any small error in the mass may lead to a considerably large error in the solution and in turn

affect the patterns we observe.

In view of the breadth and nature of the problems considered, a moving collocation method for solving fourth order evolutionary partial differential equations is introduced in chapter 3. This method is basically an extension of the moving collocation method for solving second order evolutionary partial differential equations. MC4, a subroutine written in FORTRAN which implements the moving collocation method has been developed; the code is based on MOVCOL by Huang and Russell in 1996 [31].

Some practical aspects of the code, including the MMPDE approach, temporal and spatial smoothing and construction of the monitor function are discussed in section 3.1, A selection of monitor functions are also included in MC4, which is a new feature of MC4 that enables us to solve the fourth order partial differential equations efficiently.

Numerical experiments have been done successfully for finite time blow-up problems, the interface fluctuation equation, the Cahn-Hilliard equation and some other problems. As we can see in these experiments, the moving collocation method introduced in this thesis handles singularities effectively. For the thin film model within which the moving contact lines are present, the positivity of the solution is preserved without special treatment, while in many articles great efforts have been taken to design a code to preserve the positivity of the solution to this equation. Mass conservation is also checked and its implication in solving equations with finite time blow-up or fast decay is discussed.

4.2 Further Work

While we have successfully solved several problems with MC4, a lot of work still needs to be done in order to enable us to solve a large number of problems effectively.

First, there are some other problems with which we still have difficulty to formulate and start the integration efficiently (sometimes it takes hundreds of hours to integrate over a small time interval). There are also problems that the current version of MC4 can't handle. For instance, we only consider the equations with two boundary

conditions from the left end of the interval and two from the right. Modifications need to be done so that the code can handle equations with other types of boundary conditions. Further work is also needed if equations involving u_{tt} are to be solved.

Second, further theoretical analysis of the moving collocation method for fourth order partial differential equations is still in progress. How to choose the monitor functions, the MMPDEs and temporal and spatial smoothing parameters are still key issues, as they are for the moving collocation method for second order partial differential equations. In section 3.1, we have addressed issues related to the construction of the monitor functions, spatial and temporal smoothing and choosing the MMPDEs. However, most of these discussions are based on an empirical approach. A number of issues related to the formulation and solution of the MMPDE are still unclear. In [10], the effect of different choices of MMPDEs and monitor functions as well as the time smoothing parameter were studied for the second order semilinear parabolic PDE

$$u_t = u_{xx} + u^p, p > 1.$$

The authors claimed that those methods might be able to be generalized to other PDEs with blow-up. It is shown in [8] that in contrast to the solutions of the classical second order parabolic equations $u_t = u_{xx} + u^p, p > 1$ arising from the combustion theory, the blow-up in their fourth order counterparts is asymptotically self-similar. This property tends to make theoretical analysis for fourth order PDEs easier. It is very possible that we can do some theoretical analysis for fourth order problems. Even if just for a specific problem, it will provide us some guidance on how to solve a problem efficiently.

Finally, and most important of all, it is urgent that we do the convergence and error analysis. In our moving collocation method, the convergence rate and errors are interwoven with the performance of the ODE solver, which causes great difficulty to analyze them; however, even theoretical analysis of the local truncation error in formulating the resulting ODEs will be very helpful in understanding the numerical results we obtained by using MC4.

Bibliography

[1] F. Bernis, Change of sign of the solutions to some parabolic problems, *Nonlinear Analysis and Applications*, V. Lakshmikantham, ed., *Lecture Notes in Pure and Applied Mathematics***109**, Marcel Dekker, New York, 1987.

[2] F. Bernis, Finite speed of propagation and continuity of the interface for thin viscous flows, *Advances in Differential Equations* **3**, 337-368, 1996.

[3] A. L. Bertozzi, Loss and gain of regularity in a lubrication equation for thin viscous films, in *Free Boundary Problems: Theory and Applications, Proceedings of the International Colloquium on Free Boundary Problems, Toledo, Spain, June 1993*, J. I. Díaz, M. A. Herrero, A. Liñán, and J. L. Vázquez, eds., *Pitman Research Notes in Mathematics Series* **323**, Longman House, London, 72-85, 1995.

[4] A. L. Bertozzi, The mathematics of moving contact lines in thin liquid forms, *Notice of the AMS* **45**, 689-697, 1998.

[5] C. de Boor, Good approximaion by slines with variable knots, II, *Springer Lecture Notes Series* **363**, Springer-Verlag, Berlin, 1973.

[6] Elena Beretta, Michiel Berstch and Roberta Dal Passo, Nonnegative solutions of a fourth order nonlinear degenerate parabolic equation, *Archive for Rational Mechanics and Analysis* **129**, 175-200, 1995.

[7] F. Bernis and R. Friedman, Higher order nonlinear degenerate parabolic equations, *Journal of Differential Equations* **83**, 179-206, 1990.

[8] C. J. Budd, V. A. Galaktionov and J. F. Williams, Self-similar blow-up in higher-order semilinear parabolic equation, *in press*, 2002.

[9] J. Barták, L. Herrmann, V. Lovicar and O. Vejvoda. *Partial Differential Equations of Evolution*, Ellis Horwood, 1991.

[10] C. J. Budd, W. Huang and R. D. Russell, Moving mesh methods for problems with blow-up, *SIAM Journal on Science Computing* **17**, 305-327, 1996.

[11] P. M. Bleher, J. L. Lebowitz and E. R. Speer, Existence and positivity of solutions of a fourth order nonlinear PDE describing interface fluctuations, *Communications in Pure and Applied Mathematics* **47**, 923-942, 1994.

[12] G. Beckett, J. A. Mackenzie, Convergence analysis of finite-difference approximations on equidistributed grids to a singularly perturbed boundary value problems, *Applied and Numerical Mathematics* **35**, 109-131, 2000.

[13] G. Beckett, J. A. Mackenzie, A. Ramage and D. M. Sloan, On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution, *Journal of Computational Physics* **167**, 372-392, 2001.

[14] A. L. Bertozzi and M. Pugh, The lubrication approximation for thin viscous films: regularity and long time behavior of weak solutions, *Communications in Pure and Applied Mathematics* **49**, 85-123, 1996.

[15] A. L. Bertozzi and M. Pugh, Finite-time blow-up of solutions of some long-wave unstable thin film equations, *Indiana University Mathematics Journal* **49**, 1323-1366, 2000.

[16] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, 2002.

[17] J. B. van den Berg and R. C. A. M. van der Vorst, Stable patterns for fourth order parabolic equations, *Duke Mathematics Journal* **115**, 513-558, 2002.

[18] A. J. Bernoff and T. P. Witelski, Linear stability of source-type similarity solutions of the thin film equation, *Applied Mathematics Letters* **18**, 599-606, 2002.

[19] M. J. Caceres, J. A. Carrillo and G.Toscani. Long-time behavior for a nonlinear fourth order parabolic equation, *Transactions of the American Mathematical Society*, (in press), 2003.

[20] H.-Ch. Crunau and G. Sweers, Positivity for equations involving polyharmonic operators with Dirichlet boundary conditions, *Mathematische Annalen* **307**, 589-626, 1997.

[21] D. S. Dodson, Optimal order approximation by polynomial spline functions, Ph.D. thesis, Purdue University, West Layfette, IN, 1972.

[22] E. A. Dorfi and L. O'c. Drury, Simple adaptive grids for 1-D initial value problems. *Journal of Computational Physics***95**, 175-195, 1991.

[23] B. Derrida, J. L. Lebowitz, E. Speer and H. Sporn, Fluctuations of a stationary non-equilibrium interface, *Physics Review Letters.* **67**, 165-168, 1991.

[24] V. A. Galaktionov, Invariant subspaces and new explicit solutions to evolution equations with quadratic nonlinearities, *Proceedings of Royal Society of Edinburgh* **125A**, 225-246, 1995.

[25] V. A. Galaktionov, On a spectrum of blow-up patterns for a higher-order semilinear parabolic equations, *Proceedings of Royal Society of London* **457A**, 1623-1643, 2001.

[26] V. A. Galaktionov and S. I. Pohozaev, Existence and blow-up for higher-order semilinear parabolic equations: majorizing order-preserving operators, *Indiana University Mathematics Journal* **51**, 1321-1338, 2002.

[27] V. A. Galaktionov and J. L. Vazquez, *The Problem of Blow-Up in Nonlinear Parabolic Equations*, Pitman, 2000.

[28] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, *Journal of Computational Physics* **171**, 753-775, 2001.

[29] D. F. Hawken, J. J. Gottlieb and J. S. Hansen, Review of some adaptive node-movement techniques in finite element and finite difference solutions of PDEs, *Journal of Computational Physics* bf 95, 254-302, 1991.

[30] T. Hocherman and P. Rosenau. On KS-type equations describing the evolution and rupture of a liquid interface, *Physica D* **67**, 113-125, 1993.

[31] W. Huang and R. D. Russell, A moving collocation method for solving time dependent partial differential equations, *Applied Numerical Mathematics* **20**, 101-116, 1996.

[32] W. Huang and R. D. Russell, Analysis of moving mesh partial differential equations with spatial smoothing, *SIAM Journal on Numerial Analysis* **34**, 1106-1126, 1997.

[33] W. Huang, Y. Ren and R. D. Russell, Moving mesh methods based on moving mesh partial differential equations, *Journal of Computational Physics* **113**, 279-290, 1994.

[34] W. Huang, Y. Ren and R. D. Russell, Moving mesh partial differential equations(MMPDEs) based upon the equidistribution principle, *SIAM Journal on Numerial Analysis* **31**, 709-730, 1994.

[35] A. Jüngel and R. Pinnau, Global non-negative solutions of a nonlinear fourth-order parabolic equation for quantum systems, *SIAM Journal on Mathematical Analysis* **32**, 760-777, 2000.

[36] A. Jüngel and R. Pinnau, A positivity-preserving numerical scheme for a nonlinear fourth order parabolic system, *SIAM Journal on Numerical Analysis* **39**, 385-406, 2002.

[37] T. G. Mayers, Thin films with high surface tension, *SIAM Review* **40**, 441-462, 1998.

[38] K. Miller, Moving finite elements II, *SIAM Journal on Numerical Analysis* **18**, 1033-1057, 1981.

[39] K. Miller and R. N. Miller, Moving finite elements I, *SIAM Journal on Numerical Analysis* **18**, 1019-1032, 1981.

[40] C. W. Mastin, J. F. Thompson and Z. U. A. Warsi, *Numerical Grid Generation: Foundations and Applications*, Elsevier, New York, North-Holland, Amsterdam, 1985.

[41] L. R. Petzold, A description of DASSL: A differential/algebraic system solver, SAND 82-8637, Sandia Labs, Livermore, Cal., 1982.

[42] L. Pareschi, G. Russo and G. Toscani, A kinetic formulation of Hele-Shaw flow, *Comptes Rendus de l'Acadmie des Sciences de Paris*, to appear.

[43] L. A. Peletier and W. C. Troy, *Spatial Patterns: Higher Order Models in Physics and Mechanics*, Birkhauser, Boston, 2001.

[44] Y. Ren, Theory and computation of moving mesh methods for solving time-dependent partial differential equations, Ph.D. thesis, Simon Fraser University, 1992.

[45] P. Rosenau, Nonlinear dispersion and compact structures, *Physical Review Letters* **73**, 1737-1741, 1994.

[46] Y. Ren and R. D. Russell, Moving mesh techniques based upon equidistribution, and their stability, *SIAM Journal on Scientific and Statistical Computing* **13**, 1265-1286, 1992.

[47] L. G. Reyna and M. J. Ward, Metastable internal layer dynamics for the viscous Cahn-Hilliard equation, *Methods and Applications of Analysis* **2**, 285-306, 1995.

[48] A. A. Samarskii, V. A. Galaktionov, S. P. Kurdyumov and A. P. Mikhailov, *Blow-up in Quasilinear Parabolic Equations*, Walter de Gruyter, Berlin, New York, 1995.

[49] J. M. Stockie, J. A. Mackenzie and R. D. Russell, A moving mesh method for one dimensional hyperbolic conservation law, *SIAM Journal on Scientific Computing* **22**, 1791-1813, 2000.

[50] P. Saucez, A. Vande Wouwer, W. E. Schiesser and P. Zegeling, Adaptive grids for high-order PDE models, International Conference on Scientific Computation and Differential Equations(SCICADE'01), Vancouver, Canada, 2001.

[51] A. B. White, Jr., On selection of equidistributing meshes for two-point boundary value problems, *SIAM Journal on Numerical Analysis* **16**, 472-502, 1979.

[52] Jeff Williams, Scaling and singularities in higher-order nonlinear differential equations, Ph.D. thesis, University of Bath, 2003.

[53] Z. WU, J. Zhao, J. Yin and H. Li, *Nonlinear Diffusion Equations*, World Scientific Publishing, New Jersey, 2001.

[54] L. Zhornitskaya and A. L. Bertozzi, Positivity preserving numerical schemes for lubrication type equations, *SIAM Jounal on Numerical Analysis* **37**, 523-555, 2000.