NUMERICAL SIMULATION OF THE WAVE (SHOCK PROFILE) PROPAGATION OF THE KURAMOTO-SIVASHINSKY EQUATION USING AN ADAPTIVE MESH METHOD

A thesis submitted to the University of Zimbabwe
in partial fulfillment of the requirements for the degree of
Master of Science
in the Faculty of Science

By
Denson Muzadziwa
Supervised by Dr. S. T. Sikwila
Department of Mathematics
July 2015

Contents

List of Figures

List of Tables

Abstract

In this dissertation we solve the Kuramoto-Sivanshinsky equation numerically using an adaptive mesh method. Discretization in time is done using Crank-Nicolson with septic Hermite collocation method applied in space on an adaptive mesh. The adaptive mesh is a solution of moving mesh partial differential equations derived from the principle of equidistribution. A rezoning approach which works with a decoupled solution procedure is then used to develop a matlab code to produce the numerical results. The method is evaluated for effectiveness and computational efficiency with the most current best method available in the literature.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Acknowledgements

I would like to thank my supervisor Dr. S. T. Sikwila for his guidance and willingness to share his knowledge through this study. I am also grateful for his continuing patience and encouragement. I thank my friends for the support they rendered in various capacities which contributed to the success of this dessertation. The most important gratitude is for my wife Portia for her love, support and continuous encouragement to never give up.

Chapter 1

Introduction

The Kuramoto-Sivashinsky (K-S) equation is a non-linear fourth order partial differential equation (PDE) discovered separately by Yoshiki Kuramoto and Gregory Shivashinsky in the study of non-linear stability of travelling waves. Shivashinsky [33] came up with the equation while modelling small thermal diffusive instabilities in laminar flame front in 1977. In this case, the solution $u(x_1, x_2, t)$ is the perturbation of an unstable planar flame front in the direction of propagation.

Sivashinsky also discovered the equation as a model for cellular flame patterns produced in the irregular burning of premixed gases. His main area of concern was to qualitatively describe the forces affecting the shape of a flame front. The forces of interest to him were diffusion, heat conduction, temperature of the flame and the hydrodynamics of the underlying mixture. It is against this background that Sivashinsky used a flame model by Istratov and Librovich [17] which provided him with equations to describe these forces and enable him to come up with the equation.

Kuramoto [19, 20, 21, 22] derived the equation in the study of the Belousov-Zabotinskii reaction as a model of diffusion induced chaos. $u(x_1, x_2, x_3, t)$ was considered to be a small perturbation of a global periodic solution just beyond the parameter domain where the Hopf bifurcation has occurred.

In one dimension the K-S equation is

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+\frac{\partial^2 u}{\partial x^2}+\frac{\partial^4 u}{\partial x^4}=0. \tag{1.1}$$

It has been discovered that the equation describes other physical contexts such as long waves on thin films, long waves on the interface between two viscous fluids and unstable drift waves in plasmas.

The structure of the K-S equation is such that the second derivative is an energy source and thus has a distributing effect. The non-linear term $u\frac{\partial u}{\partial x}$ is a correction to the phase speed and is caused by the movement associated with mean flow for thin films. The term is responsible for transferring energy from low to high wave numbers while the fourth derivative is the dominating term and is responsible for stabilising the equation.

Amongst other applications, the K-S equation is used as a model to describe the fluctuations of the positions of a flame front, the motion of a fluid going down a vertical wall or a spatially uniform oscillating chemical reaction in a homogeneous medium.

Several methods have been used to solve the equation numerically and of interest are the methods based on the method of lines (MOL). In this method, spatial derivative terms are replaced by algebraic approximations which result in a system of ordinary differential equations (ODEs). Numerical integration is then implemented in time to obtain the numerical solution. Khater and Temsah [18] use the Chebyshev spectral collocation method whereby exact derivatives are replaced by derivatives of interpolating polynomials at Chebyshev points in the domain. The Backward difference formula is then applied in time. Mittal and Arora [26] use quintic B-splines which are fifth order B-splines to discretize the spatial derivatives in space and the Crank- Nicholson scheme is applied in time. The numerical solution was found to be close to the exact solution and error results proved to be less than those produced by the lattice Boltzmann method [23]. Zavalani [37] uses the Fourier spectral method to solve the K-S equation. A Fourier series is used to write the PDE solution in terms of its

Fourier series. A system of ODEs for the time dependent coefficients is obtained by substitution of this series into the PDE. A time stepping method is then used to solve the ODE system. In the meshless method of lines [12], the numerical solution of the equation is approximated by a set of scattered nodes and radial basis functions are used to approximate the spatial derivatives to transform the system into a system of first order ODEs. Multiquadric(MQ), Gaussian(GA) and inverse Multiquadric(IMQ) functions are used as basis functions. The resulting ODE system in time is solved using Runge Kutta method of fourth order. Results obtained are very close to the exact solution. The method implemented using MQ radial basis functions performs better than with the other two types of basis functions and it gives higher level of accuracy using a lower number of spatial nodes. In [36], Zarebnia and Parvaz implements the method of lines for the solution of the K-S equation using the septic B-spline collocation method. Seventh order B-splines are used to discretize the derivatives on a fixed mesh in space and finite difference formula in conjunction with the θ-weight method is applied in time. Each seventh order B-spline covers eight elements and thus each element is covered by eight splines. Each septic B-spline is also continuous up to the seventh derivative. Thus the method gives a better numerical approximation for the solution of the equation. A comparison of global relative errors at different times shows that the septic B-spline collocation method is more accurate than the quintic B-spline method.

In this dissertation, an adaptive method is implemented to obtain the numerical solution of the K-S equation. This has the advantage of enabling sufficient mesh points to be concentrated in the region of rapid solution variations thereby increasing accuracy and efficiency. It is the aim of this thesis to reduce the error in obtaining the numerical solution of the K-S equation in comparison to that obtained in solving the equation on a fixed uniform mesh using septic B-spline collocation method, which has proved to be the current accurate and efficient method.

Using a fine mesh is not necessary when implementing adaptive methods as accuracy of the numerical solution is achieved by concentrating computational effort in regions involving high solution variation. This in turn improves computational efficiency as fewer mesh points can be used with the bulk of them located in regions were there are needed the most. Adapting a mesh is one way of eliminating oscillations in a solution as it allows the use of a small mesh. An adaptive small mesh is capable of keeping track of important structures that occur in PDEs and in turn an accurate solution is obtained. The major drawback of adaptive methods is that they sometimes introduce an extra level of complexity to the system through the mesh equation and this may lead to additional computational cost and numerical instability.

The three types of techniques that exist for adaptive mesh methods are h-refinement, p-refinement and r-refinement method. In h-refinement, extra nodes are added or removed to an existing uniform mesh when a certain criteria is satisfied resulting in a refined mesh or coarse mesh [1, 11, 31]. p-refinement method improves the accuracy of the numerical solution by increasing or decreasing the order of the basis functions in each element so as to minimize the error in the approximation. The technique is mainly applied in finite elements method [11]. The relocation refinement method or r-refinement technique as they are usually called, relocates nodes towards areas which need a high spatial resolution in order to capture important characteristics in the solution. This results in smaller mesh cells in the regions of interest thus focussing computational effort in those areas whilst at the same time enjoying the benefits of using a fixed number of mesh points. The method is described by Miller and Miller in [25] and then extended by Haung and Russell in [14, 15]. In this dissertation we are interested in adapting our mesh using the r-refinement method.

Chapter 2

Grid Generation

2.1 Equidistribution Principle

The problem of how to choose a suitable grid for the adaptive moving mesh method can be solved by several techniques. Amongst these are the functional minimisation and the Equidistribution Principle (EP). The functional minimisation uses the measurable properties of a mesh to generate a moving mesh from variational principle. In this dissertation we will consider the EP, a concept first introduced by Burchard [6] as a method of finding changing nodes for optimal spline approximations. Deboors in [8, 9] and Dodson [10] developed the concept for the solution of boundary value problems and in particular, it is deBoors who came up with a simpler algorithm for an equidistributed mesh in [9]. Ren and Russell [28] and Huang, Ren and Russell [14, 15] came up with different forms of the discrete and continuous EP and the associated two point boundary value problem method. Moving mesh partial differential equations (MMPDEs) based on the EP were also derived in these papers.

The idea underlying the EP concept is that if some measure of the error in the numerical solution can be approximated, then mesh points may be selected in a way as to equally distribute the error in each subinterval. Such a measure of the solution error is called a monitor function, denoted by $M(u(x,t),t)$, which is usually a user defined function. It is a positive definite function of the solution and/or its derivatives.

We can consider the EP idea as a coordinate mapping from a computational domain Ω_c to a physical space Ω_p where $x, \xi \in [a, b]$. For an arbitrary function $f = f(x,t) = f(x(\xi,t),t)$, we use the notation

$$\begin{aligned}
f_x &\equiv \frac{\partial f}{\partial x} \\
&\equiv \left.\frac{\partial f}{\partial x}\right|_{t \text{ fixed}} \\
f_t &\equiv \frac{\partial f}{\partial t} \\
&\equiv \left.\frac{\partial f}{\partial t}\right|_{x \text{ fixed}} \\
\frac{\partial f}{\partial \xi} &\equiv \left.\frac{\partial f}{\partial \xi}\right|_{t \text{ fixed}} = \left.\frac{\partial f}{\partial x}\right|_{t \text{ fixed}} \left.\frac{\partial x}{\partial \xi}\right|_{t \text{ fixed}} \\
\dot{f} &\equiv \frac{df}{dt} \\
&\equiv \left.\frac{\partial f}{\partial t}\right|_{\xi \text{ fixed}} = \left.\frac{\partial f}{\partial x}\right|_{t \text{ fixed}} \left.\frac{\partial x}{\partial t}\right|_{\xi \text{ fixed}} + \left.\frac{\partial f}{\partial t}\right|_{x \text{ fixed}}.
\end{aligned} \tag{2.1}$$

The computational domain is usually considered as a uniform mesh

$$\xi_i = \frac{(i-1)(b-a)}{N}, \qquad i = 1, ..., N+1, \tag{2.2}$$

where N is a positive integer. In this regard, the goal of the grid generation problem becomes one of finding a suitable coordinate mapping or transformation. Huang and Russell [15] use the EP to derive moving mesh partial differential equations (MMPDEs) whose solutions are mesh functions $\{X_i(t)\}_{i=1}^{N+1}$ or moving meshes

$$\Pi := \{a = X_1 < X_2(t) < ... < X_N(t) < X_{N+1} = b\} \tag{2.3}$$

which are equidistributing for all values of t.

The continuous form of the EP is the integral

$$\int_a^{x(\xi,t)} M(s(\xi,t))ds = \xi\theta(t) \tag{2.4}$$

where

$$\theta(t) = \int_a^b M(s,t)ds. \tag{2.5}$$

Differentiating (2.4) with respect to ξ once and twice, we obtain

$$M(x(\xi,t),t)\frac{\partial}{\partial\xi}x(\xi,t) = \theta(t) \tag{2.6}$$

and

$$\frac{\partial}{\partial\xi}\left\{M(x(\xi,t),t)\frac{\partial}{\partial\xi}x(\xi,t)\right\} = 0 \tag{2.7}$$

Equations (2.6) and (2.7) are called quasi-static equidistribution principles (QSEPs) as they do not contain the node speed $\dot{x}(\xi,t)$.

Time differentiation of equation (2.7) and expansion gives

$$(MMPDE1) \quad \frac{\partial}{\partial\xi}\left(M\frac{\partial\dot{x}}{\partial\xi}\right) + \frac{\partial}{\partial\xi}\left(\frac{\partial M}{\partial\xi}\dot{x}\right) = -\frac{\partial}{\partial\xi}\left(\frac{\partial M}{\partial t}\frac{\partial x}{\partial\xi}\right). \tag{2.8}$$

The derivative $\frac{\partial M}{\partial t}$ is regarded as a source of mesh movement in MMPDE1 since it will have solutions with zero speed if the value of the derivative is zero. Thus we can conclude that the mesh remains stationary regardless of the initial mesh if $M(x,t)$ is independent of time.

We can achieve stability in the movement of the mesh by requiring the EP (2.7) to be satisfied at a later time $t+\tau$ and $(0 < \tau \ll 1)$ [15]. Therefore the mesh need to satisfy the condition

$$\frac{\partial}{\partial\xi}\left\{M(x(\xi,t+\tau),t+\tau)\frac{\partial}{\partial\xi}x(\xi,t+\tau)\right\} = 0 \tag{2.9}$$

which is capable of controlling mesh movement. Taylor's expansions for $x(\xi,t+\tau)$ and $M(x(\xi,t+\tau),t+\tau)$ are given by

$$\begin{aligned} \frac{\partial}{\partial\xi}x(\xi,t+\tau) &= \frac{\partial}{\partial\xi}x(\xi,t) + \tau\frac{\partial}{\partial\xi}\dot{x}(\xi,t) + O(\tau^2) && (2.10) \\ M(x(\xi,t+\tau),t+\tau) &= M(x(\xi,t),t) + \tau\dot{x}\frac{\partial}{\partial x}M(x(\xi,t),t) \\ &\quad +\tau\frac{\partial}{\partial t}M(x(\xi,t),t) + O(\tau^2) && (2.11) \end{aligned}$$

respectively. We substitute expansions (2.10) and (2.11) into the result (2.9) and ignore higher order terms to obtain

$$(MMPDE2)\ \frac{\partial}{\partial \xi}\left(M\frac{\partial \dot{x}}{\partial \xi}\right)+\frac{\partial}{\partial \xi}\left(\frac{\partial M}{\partial \xi}\dot{x}\right)=-\frac{\partial}{\partial \xi}\left(\frac{\partial M}{\partial t}\frac{\partial x}{\partial \xi}\right)-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right). \tag{2.12}$$

An additional correction term

$$-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right) \tag{2.13}$$

exists in (2.12) and is a measurement of how close the mesh $x(\xi, t)$ satisfy QSEP. It contains the relaxation parameter τ which plays a pivotal role in driving the mesh towards equidistribution. A smaller τ will result in high mesh velocities where as a very large value of τ tends to give a stationary mesh. In this regard, the correction term can be considered as the source of mesh movement and mechanism to drive the mesh towards equidistribution even in the absence of the source of mesh movement, $\frac{\partial M}{\partial t}$. Consequently, dropping the term $\frac{\partial x}{\partial \xi}\frac{\partial M}{\partial t}$ or both terms $\frac{\partial x}{\partial \xi}\frac{\partial M}{\partial t}$ and $\dot{x}\frac{\partial M}{\partial \xi}$ in MMPDE2 is of less or no significance and gives the following MMPDEs

$$(MMPDE3)\qquad \frac{\partial^2}{\partial \xi^2}(M\dot{x})=-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right) \tag{2.14}$$

and

$$(MMPDE4)\qquad \frac{\partial}{\partial \xi}\left(M\frac{\partial \dot{x}}{\partial \xi}\right)=-\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right). \tag{2.15}$$

A MMPDE which explicitly computes the node speed was computed by Anderson in [2] and is given by

$$(MMPDE5)\qquad \dot{x}=\frac{1}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right). \tag{2.16}$$

There are other MMPDEs which can be derived using attraction and repulsion of pseudo-forces. Literature on these can be found in [15].

The moving mesh method in this dissertation makes use of MMPDE4 because we can obtain unique solutions for the mesh velocities with Dirichlet boundary conditions. This is discretized using centred finite difference approximation in space to give

$$\frac{M_{i+1}+M_i}{2(\frac{1}{N})^2}(\dot{x}_{i+1}-\dot{x}_i)-\frac{M_i+M_{i-1}}{2(\frac{1}{N})^2}(\dot{x}_i-\dot{x}_{i-1})=-\frac{Ei}{\tau} \tag{2.17}$$

where

$$E_i = \frac{M_{i+1} + M_i}{2(\frac{1}{N})^2}(x_{i+1} - x_i) - \frac{M_i + M_{i-1}}{2(\frac{1}{N})^2}(x_i - x_{i-1}). \tag{2.18}$$

2.2 Moving mesh methods

Three major components which enables a mesh to be moved are:

- the strategy used to move the mesh
- the discretization method for the physical PDE
- the approach used to solve the system of physical and mesh equations

The quasi-Lagrange approach and the rezoning approach are used to treat the effect caused by the mesh movement in the time discretization of the physical PDE. The quasi-Lagrange approach uses either the coupled or decoupled solution procedure and the rezoning approach is designed to work only with the decoupled solution procedure.

In the coupled solution procedure, the mesh equation and the physical PDE are solved simultaneously as one system for the mesh and the physical solution [16]. Interpolation of the solution from one mesh to the next is not needed and the mesh is able to keep its dynamical properties such as scaling structures [3, 4, 5]. However the procedure has the disadvantage of creating highly nonlinear equations even from linear physical PDEs which makes the resulting system difficult and expensive to solve.

For the decoupled solution procedure, a mesh at the new time level is first generated using old values of the mesh and solution. The solution is then obtained at the new time level. Huang [13], Huang and Russell [4] and Ceniceros and Hou [7] solved moving mesh equations using the decoupled procedure. The procedure has the advantage of ease of coding as the PDE solution and moving mesh part can be coded separately thus improving efficiency in each section of the solution. The decoupled solution procedure has the the disadvantage of producing a mesh that lags in time. It also lacks a built-in mechanism to force a badly generated mesh on track thus causing instability in the solution integration.

2.2.1 Quasi-Lagrange approach

In the quasi-Lagrange approach, there is continuous movement of mesh points in time. The physical PDE is reformulated into derivatives along the trajectories and a convective term is added to reflect the mesh movement. The extra convective term and the new time derivatives are treated in the same way as other terms in the physical PDE. Interpolation of the solution from the old mesh to the new mesh in not necessary. The approach was used in the solution of Burger's equation and scalar reaction-diffusion equation from combustion in [14] and to solve Fisher's equation in [27].

2.2.2 Rezoning approach

The mesh in the rezoning approach moves in an intermittent manner and as result the method is applied with the decoupled solution procedure. A mesh is held stationary while solving for the physical PDE and then updated at each time level using mesh equations. The physical solution is then interpolated from the old mesh to the new one and the PDE is discretized on the new mesh held fixed for the current time step. In this dissertation, we are interested in the rezoning approach because it allows for the coding of the mesh equation and the physical equation separately and improvements can be made on each part without altering the whole code.

Tang [35] and H and T Tang [34] developed the rezoning approach for solving one and two dimensional hyperbolic systems of conservation laws. In [35], experiments were carried out on the inviscid burgers equation in one dimension where a second order finite volume scheme was used in conjunction with a second order Runge-Kutta discretization. It was observed that the adaptive mesh method is able to follow the moving shock with a second order rate of convergence. In the same paper, the method was also tested with the one dimension equations of gas dynamics where the method was able to resolve the contact and shock discontinuities in the solution profile. According to [5, 34, 35], an algorithm for the rezoning approach is as follows:

1. Solve the given physical PDE on the current mesh.
2. Use the PDE solution obtained to calculate the monitor function.
3. Find the new mesh by solving a MMPDE.
4. Adjust the current PDE solution to suite the new mesh by interpolation.
5. Solve the physical PDE on the new mesh for the solution in the next time.

2.2.3 Monitor functions

The choice of monitor functions play an important role in the adaptive moving mesh strategy. This has a bearing on the efficiency and optimization of the initial data and thus precise choices must be made for different functions. The choice is influenced by the need to place mesh nodes so that a certain quantity is equally distributed, that is the idea of EP. This quantity might be an area when the EP is applied to a non-linear differential equation or the arc length in the case of problems with steep fronts. The mass monitor function is suitable for the former whilst the arc-length monitor function is suitable for the latter scenario. Given that $u(x,t)$ is the solution for an unsteady one-dimensional problem, the mass monitor function and the arc-length monitor function are given by

$$M(x,t) = u(x,t). \tag{2.19}$$

and

$$M(x,t) = \sqrt{1+\alpha^2\left(\frac{\partial u}{\partial x}\right)^2} \tag{2.20}$$

respectively where α is a user specified parameter chosen depending on the behaviour of the solution $u(x,t)$. Another commonly used monitor function is the curvature monitor function given by

$$M(x,t) = \left(1 + \alpha^2\left(\frac{\partial^2 u}{\partial x^2}\right)^2\right)^{\frac{1}{4}}. \tag{2.21}$$

Monitor functions are sometimes modified so as to suite specific solution functions. This can be done by combining two or more different types of monitor functions which have the ability to locate mesh points in different regions of interest thus giving a more refined representation of the solution profile. For example, a combination of an arc-length monitor function and a curvature monitor function can be given by

$$M(x,t) = \left(1 + \alpha^2\left(\frac{\partial u}{\partial x}\right)^2 + \alpha^2\left(\frac{\partial^2 u}{\partial x^2}\right)^2\right)^{\frac{1}{2}}. \tag{2.22}$$

With the correct choice of α, this monitor function is a powerful tool to equidistribute mesh points in problems such as the K-S equation. Another example of a modified monitor function is the extended curvature monitor function which is of the form

$$M(x,t) = \left(1 + \alpha^2(1-u)^2 + \beta^2(a-u)^2\left(\frac{\partial^2 u}{\partial x^2}\right)^2\right)^{\frac{1}{2}}. \tag{2.23}$$

The monitor function was developed in [27] for the moving mesh solution of Fisher's equation. It works with three user-defined parameters α, β and a.

To illustrate how monitor functions equidistribute mesh points, we consider the linear boundary value problem of the function $u(x)$ given by

$$\varepsilon\frac{d^2u}{dx^2} + \frac{du}{dx} = 0, \qquad x \in (0,1), \tag{2.24}$$

with Dirichlet boundary conditions

$$u(0) = 1 \qquad\qquad u(1) = \exp\left(-\frac{1}{\varepsilon}\right). \tag{2.25}$$

This problem has the exact solution given by

$$u(x) = \exp\left(-\frac{x}{\varepsilon}\right). \tag{2.26}$$

We take the value of ε in the ODE and its exact solution to be 10^{-1}. Given a mesh $\{x_i\}_{i=1}^{N+1}$ where N is the number of mesh intervals, the finite difference discretizations of first and second derivatives on this mesh are given by

$$\begin{aligned} \frac{du}{dx} &\approx \frac{u_{i+1}-u_{i-1}}{x_{i+1}-x_{i-1}}, \\ \frac{d^2u}{dx^2} &\approx \frac{2}{x_{i+1}-x_{i-1}}\left[\frac{u_{i+1}-u_i}{x_{i+1}-x_i} - \frac{u_i-u_{i-1}}{x_i-x_{i-1}}\right], \end{aligned} \tag{2.27}$$

for $i = 2, ..., N$.

We substitute the expressions in (2.27) into the ODE (2.24) to give the discretized form

$$\frac{2\varepsilon}{x_{i+1}-x_{i-1}}\left[\frac{u_{i+1}-u_i}{x_{i+1}-x_i} - \frac{u_i-u_{i-1}}{x_i-x_{i-1}}\right] + \left[\frac{u_{i+1}-u_{i-1}}{x_{i+1}-x_{i-1}}\right] = 0, \tag{2.28}$$

for $\quad i = 2, ..., N$, with

$$u_1 = 1 \quad \text{and} \quad u_{N+1} = \exp\left(-\frac{1}{\varepsilon}\right).$$

For the moving mesh method, we use QSEP (2.6) as the equation for the mesh since the ODE is a steady-state problem. The finite difference discretization of QSEP (2.6) on the mesh $\{x_i\}_{i=1}^{N+1}$ is given by

$$\frac{M_{i+1}+M_i}{2(\frac{1}{N})^2}(x_{i+1}-x_i) - \frac{M_i+M_{i-1}}{2(\frac{1}{N})^2}(x_i-x_{i-1}) = 0, \quad i = 2, ..., N \tag{2.29}$$

with

$$x_1 = 0 \quad \text{and} \quad x_{N+1} = 1.$$

The discrete forms of the arc-length, curvature and modified monitor functions are given by

$$M_i = \sqrt{1+\alpha^2\left(\frac{u_{i+1}-u_{i-1}}{x_{i+1}-x_{i-1}}\right)^2}, \tag{2.30}$$

$$M_i^4 = 1+\alpha^2\left[\frac{2}{x_{i+1}-x_{i-1}}\left(\frac{u_{i+1}-u_i}{x_{i+1}-x_i} - \frac{u_i-u_{i-1}}{x_i-x_{i-1}}\right)\right]^2 \tag{2.31}$$

and

$$M_i^2 = 1+\alpha^2\left[\left(\frac{u_{i+1}-u_{i-1}}{x_{i+1}-x_{i-1}}\right)^2+\left(\frac{2}{x_{i+1}-x_{i-1}}\left(\frac{u_{i+1}-u_i}{x_{i+1}-x_i} - \frac{u_i-u_{i-1}}{x_i-x_{i-1}}\right)\right)^2\right] \tag{2.32}$$

respectively for $i = 2, ..., N$. The number of mesh points used in this illustration is 21.

Figure 2.1 shows the distribution of mesh points on a uniform fixed mesh. The mesh points are concentrated in the flatter region and there are fewer points in the steep region. Thus the behaviour of the solution is poorly represented on this mesh.

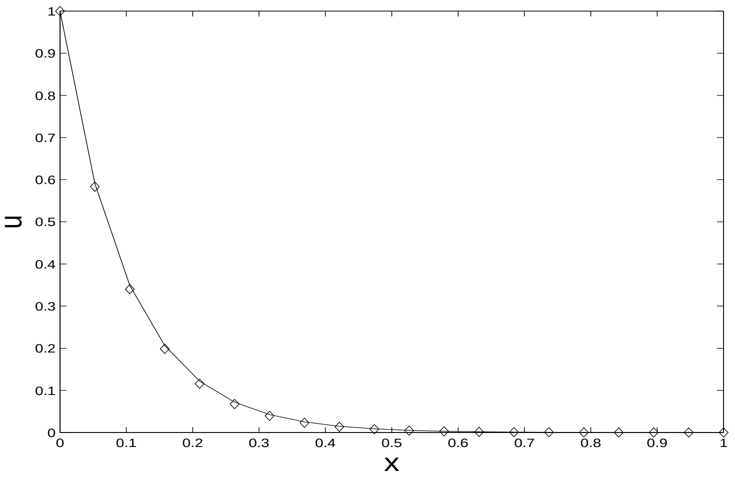

Figure 2.1: Distribution of mesh points on a uniform fixed mesh

It can be seen in figure 2.2 that the arc-length monitor function distributes mesh points both at the steep and flatter regions thus giving a smoother representation of the solution. The same level of equidistribution, can also be achieved by using the curvature monitor function with $\alpha = 10$ as shown in figure 2.3.

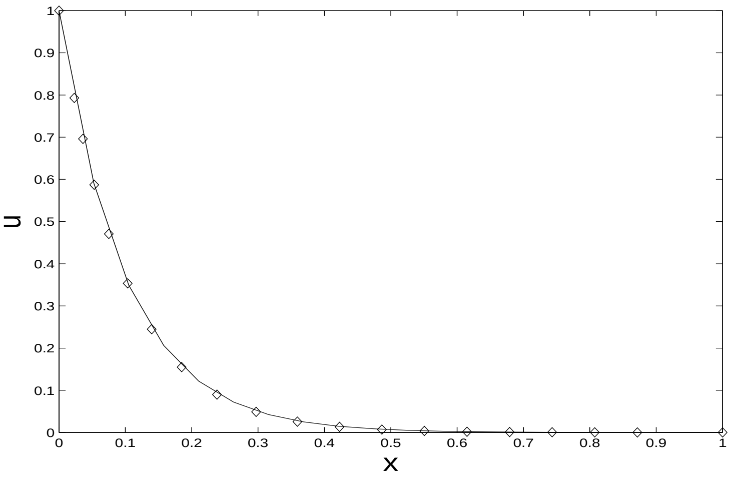

Figure 2.2: Equidistributed mesh with arc-length monitor function, $\alpha = 5$ and $\varepsilon = 10^{-1}$

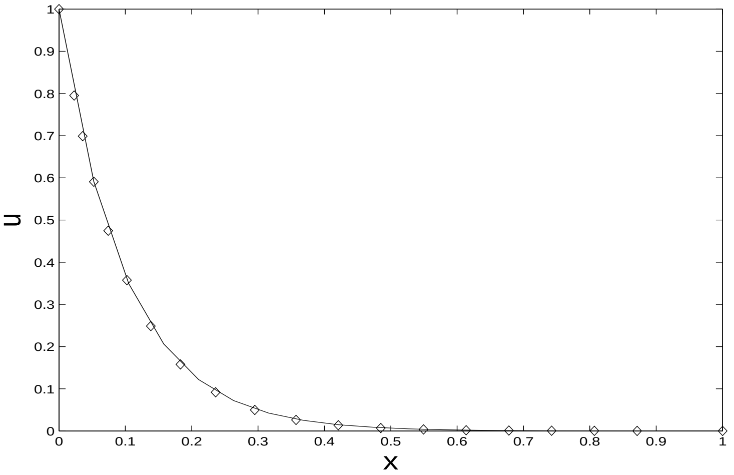

Figure 2.3: Equidistributed mesh with curvature monitor function, $\alpha = 10$ and $\varepsilon = 10^{-1}$

Figure 2.4 shows how mesh points are equidistributed using the modified monitor function (2.22).

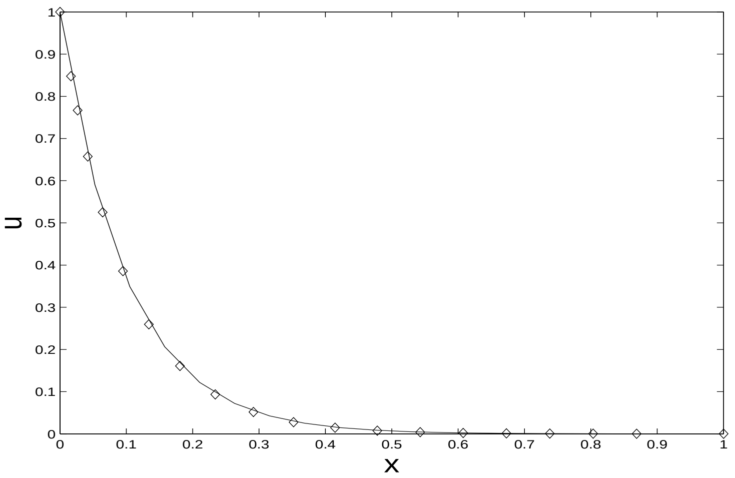

Figure 2.4: Equidistributed mesh using a modified monitor function, $\alpha = 1$ and $\varepsilon = 10^{-1}$

We note that once the value of ε is less than 10^{-1}, the scheme will not be able to resolve the boundary layer. An upwinding finite difference scheme has to be used to discretize the first derivative term.

2.2.4 Smoothing

A non-smooth mesh is usually obtained when solving problems with rapid solution variation. This might be caused by neighbouring intervals of the mesh nodes being very different such that the discretization process is highly compromised. As a consequence, such a mesh gives solutions with oscillatory behaviour thereby increasing the error in approximating the numerical solution. Non-smooth meshes also have a tendency of slowing the convergence rate thus causing inefficiencies in the solution process.

A process called smoothing has the ability to eliminate these undesirable occurrences thereby giving reasonably accurate solutions. Smoothing can be applied on either the mesh or monitor function. A fairly straight forward smoothing process is described in [14, 15] whereby smoothing is implemented on the monitor function. Values of the smoothed monitor ($\tilde{M}$) function at the point x_i are given by

$$\tilde{M}_i = \sqrt{\frac{\sum_{k=i-p}^{i+p}(M_k)^2(\frac{\gamma}{1+\gamma})^{|k-i|}}{\sum_{k=i-p}^{i+p}(\frac{\gamma}{1+\gamma})^{|k-i|}}}. \tag{2.33}$$

The parameter p is called the smoothing index and is non-negative. It determines the extent of smoothing where $p = 0$ indicates the non-smooth case and cases $p = 1, 2, 3$ produce good results. γ is non-negative and is called the smoothing parameter. It represents the rigidity of the grid and requires that neighbouring intervals should have a grid spacing not differing by more $\frac{\gamma}{1+\gamma}$ where $1 < \gamma < 2$ so as to have a stable grid.

Chapter 3

Hermite Collocation Method

3.1 Introduction

Collocation method involves determining an approximate solution to a differential equation by requiring that the approximate solution satisfy the differential equation at some discrete points (collocation points) together with boundary conditions. Consider a general fourth order PDE on the domain $[a, b]$ given by

$$u_t = F(t, x, u, u_x, u_{xx}, u_{xxx}, u_{xxxx}), \quad t > 0 \tag{3.1}$$

where F is a given continuous function. The PDE(3.1) has the initial condition given by

$$u(x, t_a) = u_0(x), \quad a < x < b, \tag{3.2}$$

and four suitable boundary conditions (two at the left boundary point a and the other two at right boundary point b) so that the Initial Boundary Value Problem (IBVP) is well posed. The boundary conditions are of the form

$$\begin{aligned}
B_{a1}(t, x, u, u_x, u_{xx}, u_{xxx}, u_{xxxx}) &= y_1(t) \\
B_{a2}(t, x, u, u_x, u_{xx}, u_{xxx}, u_{xxxx}) &= y_1(t) \\
B_{b1}(t, x, u, u_x, u_{xx}, u_{xxx}, u_{xxxx}) &= z_1(t) \\
B_{b2}(t, x, u, u_x, u_{xx}, u_{xxx}, u_{xxxx}) &= z_2(t)
\end{aligned} \tag{3.3}$$

where $y_1(t), y_2(t), z_1(t), z_2(t)$ are any suitable values that the boundary conditions can take at $x = a$ and $x = b$ respectively. We also consider a time dependent mesh with

$$a = X_1(t) < X_2(t) < ... < X_{N+1}(t) = b \quad t \in [t_a, t_b], \tag{3.4}$$

and denote the variable spatial step by $H_i = X_{i+1}(t) - X_i(t)$ for $i = 1, ..., N$. We let Φ be the functions space we wish to represent the approximate solution of the IBVP(3.1) where

$$\Phi = span\{\phi_0, \phi_1, ..., \phi_7\} \tag{3.5}$$

and $\phi_j = \phi_j(x, t)$ for $j = 1, 2, ..., 7$. The space Φ can be the space of polynomials, splines of a certain degree or some radial basis functions space. We write the approximate solution as

$$U(x, t) = \sum_{j=0}^{7} c_j \phi_j \quad X_a < x < X_b, \tag{3.6}$$

where c_j are the coefficients to be determined which can be constants or functions. Suppose we consider the internal points in the mesh (3.4) as the collocation points in this illustration. The collocation method then requires that the approximate solution (3.6) satisfies the IBVP (3.1) at each of the collocation points and the boundary conditions (3.4) at the points x_a and x_b. A quantity R_i called the residual is obtained and is set equal to zero at each collocation point where

$$R_i = U_t - F(t, x, U, U_x, U_{xx}, U_{xxx}, U_{xxxx}). \tag{3.7}$$

The expressions for the residual enforced to zero at collocation points and the boundary conditions evaluated at the boundary points gives a system of equations which is solved for the unknown coefficients c_j for $j = 0, 1, ..., 7$.

The main feature of collocation is that its output is a function (usually a piecewise polynomial) that approximates the solution of a differential equation throughout some interval. Thus it can be used to estimate the behaviour of a solution in the entire interval in question which is in contrast with finite difference method which produce a discrete set of solution values [24]. Collocation methods produce results which are usually of a higher order of accuracy as compared to finite difference methods and it is this higher accuracy that permits the use of coarser grid of points and hence a small number of points to store data and operate upon [30]. Derivatives with respect to space variables are computed explicitly and correctly and higher order derivatives need not be approximated by difference quotients which can be a problem and is a feature of finite difference method [32].

However, collocation methods are affected by the presence of boundaries as these tend to introduce instability problems that are highly restrictive as regards to time step. The methods also tend to be expensive with regard to computational time [24].

3.2 Hermite collocation method

The Hermite collocation method results from using a piecewise Hermite polynomial as the approximate solution for the physical solution $u(x,t)$ of the PDE. The piecewise polynomial makes use of the Hermite basis which is a local representation which allows the use of the same information with some scaling in each subinterval of the mesh [24]. In this section we derive the septic hermite interpolating polynomials and the corresponding 7^{th} order Hermite basis functions. The description of the septic Hermite collocation method then follows after.

3.2.1 Hermite interpolating polynomials

Suppose that we know the values of a function $u(x)$ and its consecutive derivatives up to the p^{th} order at $N+1$ points given by

$$\begin{aligned} u_i &= u(x_i) \\ u_i^{(1)} &= u(x_i) \\ &\vdots \\ u_i^{(p)} &= u(x_i). \end{aligned}$$

Our approach is to find a polynomial that matches both the function and its consecutive derivatives at these points and such a polynomial is called a Hermite polynomial. Our conditions are then given by

$$\begin{aligned} U(x_i) &= u_i, \quad i = 1, 2, ..., N+1 \\ U^{(1)}(x_i) &= u_i^{(1)}, \quad i = 1, 2, ..., N+1 \\ &\vdots \\ U^{(p)}(x_i) &= u_i^{(p)}, \quad i = 1, 2, ..., N+1. \end{aligned} \tag{3.8}$$

Since there are $(p+1)(N+1)$ conditions, we are potentially able to determine up to $(p+1)(N+1)$ unknowns in our model. Typically, this means that our polynomial will have a degree of $(p+1)(N+1)-1$ and is given by

$$U(x) = \sum_{j=0}^{(p+1)(N+1)-1} a_j x^j. \tag{3.9}$$

In order to develop a hermite interpolating polynomial $U(x)$ which can interpolate a function and its consecutive derivatives up to the third order for the interval $[0, 1]$, we substitute $p = 3$ and $N = 1$ into (3.9) to give

$$U(s) = \sum_{j=0}^{7} a_j s^j \tag{3.10}$$

We then apply the conditions in (3.8) to give the matrix system

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 12 & 20 & 30 & 42 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 24 & 60 & 120 & 210 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix}}_{\mathbf{a}} = \underbrace{\begin{pmatrix} u_0 \\ u_1 \\ u_0^{(1)} \\ u_1^{(1)} \\ u_0^{(2)} \\ u_1^{(2)} \\ u_0^{(3)} \\ u_1^{(3)} \end{pmatrix}}_{\mathbf{u}} \tag{3.11}$$

If we let S be the set of basis functions $\{1, s^2, s^3, s^4, s^5, s^6, s^7\}$ and

$$\mathbf{S} = \begin{pmatrix} 1 \\ s \\ s^2 \\ s^3 \\ s^4 \\ s^5 \\ s^6 \\ s^7 \end{pmatrix}$$

we have

$$\begin{aligned} U(s) =& \mathbf{S^T a} = \mathbf{S^T A^{-1} u} \\ =& \mathbf{L^T u} \end{aligned} \tag{3.12}$$

where

$$\mathbf{L} = \begin{pmatrix} L_{0,0}(s) \\ L_{1,0}(s) \\ L_{0,1}(s) \\ L_{1,1}(s) \\ L_{0,2}(s) \\ L_{1,2}(s) \\ L_{0,3}(s) \\ L_{1,3}(s) \end{pmatrix}$$

and

$$\begin{aligned} L_{0,0} &= (20s^3 + 10s^2 + 4s + 1)(s-1)^4 \\ L_{0,1} &= s(10s^2 + 4s + 1)(s-1)^4 \\ L_{0,2} &= \frac{s^2}{2}(4s+1)(s-1)^4 \\ L_{0,3} &= \frac{s^3}{6}(s-1)^4 \\ L_{1,0} &= -(20s^3 - 70s^2 + 84s - 35)s^4 \\ L_{1,1} &= s^4(s-1)(10s^2 - 24s + 15) \\ L_{1,2} &= -\frac{s^4}{2}(s-1)^2(4s-5) \\ L_{1,3} &= \frac{s^4}{6}(s-1)^3 \end{aligned} \tag{3.13}$$

are the septic hermite basis functions on the interval $[0, 1]$. These Hermite basis functions $L_{0,l}(s)$ and $L_{1,l}(s)$ for $l = 0, 1, 2, 3$ are polynomials of degree 7 which observe the conditions

$$\begin{array}{lll} \frac{d^k}{ds^k}L_{0,l}(0) = \delta_{k,l}, & \frac{d^k}{ds^k}L_{0,l}(1) = 0, & k, l = 0, 1, 2, 3 \\ \frac{d^k}{ds^k}L_{0,l}(0) = 0, & \frac{d^k}{ds^k}L_{1,l}(1) = \delta_{k,l}, & k, l = 0, 1, 2, 3 \end{array}$$

where $\delta_{k,l}$ denotes the Kronecker delta. The variable s is a local coordinate which takes values in the interval $(0, 1)$.

3.2.2 Septic Hermite collocation method

We consider the collocation discretization of the IBVP (3.1) on the time dependent mesh (3.4). According to (3.12), we can approximate the physical solution $u(x,t)$ on the mesh (3.4) by a piecewise Hermite polynomial of 7^{th} degree given by

$$\begin{aligned} U(x,t) &= U_i(t)L_{0,0}(s) + U_{x,i}H_i(t)L_{0,1}(s) + U_{xx,i}H_i^2(t)L_{0,2}(s) \\ &\quad + U_{xxx,i}(t)H_i^3(t)L_{0,3}(s) + U_{i+1}(t)L_{1,0}(s) + U_{x,i+1}H_i(t)L_{1,1}(s) \\ &\quad + U_{xx,i+1}H_i^2(t)L_{1,2}(s) + U_{xxx,i+1}(t)H_i^3(t)L_{1,3}(s) \end{aligned} \tag{3.14}$$

for $x \in [X_i(t), X_{i+1}(t)], i = 1, 2, ...N$ where $U_i(t), U_{x,i}(t), U_{xx,i}(t)$ and $U_{xxx,i}(t)$ are the unknown variables [30]. We define the variable s as

$$s = \frac{x - X_i(t)}{H_i(t)}, \tag{3.15}$$

where $H_i(t) = X_{i+1}(t) - X_i(t)$ for $i = 1, ..., N$. We also obtain derivatives of $U(x,t)$ with respect to the spatial variable x for $x \in [X_i(t), X_{i+1}(t)]$, by direct differentiation of (3.14) to give

$$\begin{aligned} \frac{\partial^{(l)}U(x,t)}{\partial x^l} \quad = \quad & \frac{1}{H_i(t)^l}\Bigg[U_i(t)\frac{d^{(l)}L_{0,0}}{ds^{(l)}} + U_{x,i}H_i(t)\frac{d^{(l)}L_{0,1}}{ds^{(l)}} + U_{xx,i}H_i^2(t)\frac{d^{(l)}L_{0,2}}{ds^{(l)}} + \\ & U_{xxx,i}(t)H_i^3(t)\frac{d^{(l)}L_{0,3}}{ds^{(l)}} + U_{i+1}(t)\frac{d^{(l)}L_{1,0}}{ds^{(l)}} + U_{x,i+1}H_i(t)\frac{d^{(l)}L_{1,1}}{ds^{(l)}} + \\ & U_{xx,i+1}H_i^2(t)\frac{d^{(l)}L_{1,2}}{ds^{(l)}} + U_{xxx,i+1}(t)H_i^3(t)\frac{d^{(l)}L_{1,3}}{ds^{(l)}}\Bigg] \end{aligned} \tag{3.16}$$

for $l = 1, 2, 3$. In each subinterval $[X_i, X_{i+1}]$ of the mesh (3.4), we define the four Gauss-Legendre points

$$0 < \rho_1 < \rho_2 < \rho_3 < \rho_4 < 1.$$

which are given by

$$\begin{aligned}\rho_1 &= \frac{1}{2} - \frac{\sqrt{525+70\sqrt{30}}}{70} \\ \rho_2 &= \frac{1}{2} - \frac{\sqrt{525-70\sqrt{30}}}{70} \\ \rho_3 &= 1-\rho_1 \\ \rho_4 &= 1-\rho_2.\end{aligned}$$

These Gauss-Legendre points are the roots of the Legendre polynomials of degree four mapped onto the interval $(0,1)$. We regard these points as the collocation points in each subinterval of the mesh (3.4). The endpoints are excluded as collocation points since $\rho_1 > 0$ and $\rho_4 < 1$. In order to scale Gauss-Legendre points into subsequent subintervals, we define the collocation points as

$$X_{ij} = X_i + H_i\rho_j, \quad i = 1,...,N, \quad j = 1,2,3,4. \tag{3.17}$$

We can determine the unknown variables consisting of the solution U and its first three spatial derivatives by substituting (3.14) and its appropriate derivatives into (3.1) to give the residual (3.7) at each of the four gauss points in each subinterval. We then set the residuals to be equal to zero which result in $4N$ equations in $4(N+1)$ unknowns. The boundary conditions evaluated at X_1 and X_{N+1} gives the remaining four equations thus creating a system of $4(N+1)$ equations in $4(N+1)$ unknowns.

3.3 Adaptive hermite collocation method

We can write the piecewise Hermite polynomial (3.14) in the compact form

$$U(x,t) = \sum_{l=0}^{3}(h_i)^l U_i^{(l)}(t)L_{0,l}(s) + \sum_{l=0}^{3}(h_l)^l U_{i+1}^{(l)} L_{1,l}(s) \tag{3.18}$$

Where the $4(N+1)$ unknowns are given by

$$U_i^{(l)} \approx \frac{\partial^l u}{\partial x^l}(X_i(t),t) \qquad U_{i+1}^{(l)} \approx \frac{\partial^l u}{\partial x^l}(X_{i+1}(t),t), \quad l = 0,1,2,3.$$

We discretize the time domain $[t_a, t_b]$ using the following finite sequence

$$\{t_a = t_0 < ... < t_n < ... < ... < t_k = t_b\}. \tag{3.19}$$

At each time $t = t_n = n \times dt$, we consider a non-uniform mesh $\{X_i^n\}_{i=1}^{N+1}$ where $X_i^n = X_i(t_n)$ given by

$$a = X_1^n < ... < X_{N+1}^n = b \tag{3.20}$$

with $H_i^n = X_{i+1}^n - X_i^n$ being a non-uniform spatial step for $i = 1, ..., N$. At the same time step $t = t_n$, we also consider the approximations to the exact solution $u(x, t)$ and its derivatives given by $\{U_i^n\}_{i=1}^{N+1}$ and $\{(U_i^{(l)})^n\}_{i=1}^{N+1}$ respectively where $(U_i^{(l)})^n$ represents the l^{th} derivative approximation with respect to the spatial variable x at the time $t = t_n$ for $l = 1, 2, 3$.

Chapter 2 of this dissertation describes how we construct a new mesh $\{\tilde{X_i}^n\}_{i=1}^{N+1}$ using the EP if we are given the mesh, approximate solution and/or its derivative approximations at the current time step. Our desire then is determine the new approximations $\{\tilde{U_i}^n\}_{i=1}^{N+1}$ and $\{(\tilde{U_i}^{(l)})^n\}_{i=1}^{N+1}$ which are related to the new mesh $\{\tilde{X_i}^n\}_{i=1}^{N+1}$ in a similar manner the approximations $\{U_i^n\}_{i=1}^{N+1}$ and $\{(U_i^{(l)})^n\}_{i=1}^{N+1}$ are related to the old mesh $\{X_i^n\}_{i=1}^{N+1}$ in each subinterval $[X_i, X_{i+1}]$. This process of updating the solution and its derivatives from the old mesh to the new mesh is achieved by interpolation. Interpolation can be done over the whole domain $[a, b]$ or over small subintervals $[X_i, X_{i+1}]$ which leads to piecewise polynomial interpolation. A simple example of a piecewise interpolation polynomial is the cubic Hermite polynomial. This is capable of giving a smoother interpolant over the subinterval $[X_i, X_{i+1}]$ which is continuous only up to the first derivative. Thus the use of this interpolant to estimate higher derivatives tends to give inconsistent estimates.

In this dissertation we combine the moving mesh method with the septic Hermite collocation method to solve the K-S equation which is of the form (3.1). Thus it is desirable to have a piecewise interpolating polynomial which demands that the function values and its three consecutive derivatives are satisfied in each subinterval. It must also be able to maintain properties of the data such as positivity and monotonicity. Such an interpolant is the septic Hermite interpolating polynomial.

Given the partition (3.20) and approximations $\{(U_i^{(l)})^n\}$ for $l = 0, 1, 2, 3$, suppose we desire to interpolate $U^{(l)}(x)$ at $x = \tilde{X}_i^{\,n}$ where $\tilde{X}_i^{\,n} \in [X_i^n, X_{i+1}^n]$ for $i = 1, ..., N$. Firstly, we define the local coordinate s of $\tilde{X}_i^{\,n}$ by

$$s = \frac{\tilde{X}_i^{\,n} - X_i^n}{H_i^n}. \tag{3.21}$$

We then define

$$\tilde{U}^{(l)}(\tilde{X}_i^{\,n}) = \sum_{l=0}^{3} H_i^{l-p} U_i^{(l)} \frac{d^{(l)} L_{0,l}(s)}{ds^{(l)}} + \sum_{l=0}^{3} H_i^{l-p} U_{i+1}^{(l)} \frac{d^{(l)} L_{1,l}(s)}{ds^{(l)}} \tag{3.22}$$

for $l = 0, 1, 2, 3$ to be the interpolated values of $\tilde{U}$ and the first three consecutive derivatives on the new subinterval $[\tilde{X}_i^{\,n}, \tilde{X}_{i+1}^n]$. In order to compute the approximations of U at the next time step $t = t_{n+1}$ denoted by $\{U_i^{n+1}\}_{i=1}^{N+1}$, we use the values of the new mesh $\{\tilde{X}_i^n\}_{i=1}^{N+1}$ and the updated approximations $\{\tilde{U}_i^n\}_{i=1}^{N+1}$ in a septic Hermite collocation numerical scheme. The new approximations $\{U_i^{n+1}\}_{i=1}^{N+1}$ and the new mesh $\{\tilde{X}_i^{n+1}\}_{i=1}^{N+1}$ will become the starting conditions for repeating the whole adaptive process.

Chapter 4

Numerical Results

4.1 Introduction

In this chapter we present the numerical results for the K-S problem (4.1)

$$u_t + uu_x + u_{xx} + u_{xxxx} = 0, \qquad x \in (-30, 30), \quad t > 0 \tag{4.1}$$

with boundary conditions

$$u(-30, t) = \sigma, \qquad u_x(-30, t) = \beta \tag{4.2}$$

$$u(30, t) = \omega, \qquad u_x(30, t) = \zeta \tag{4.3}$$

where σ, β, ω and ζ are obtained from the exact solution (4.4) with $c = 0.1$, $x_0 = -10$ and $k = \frac{1}{2}\sqrt{\frac{11}{19}}$. The results are obtained using septic Hermite collocation method and finite difference method on both a uniform and a non-uniform mesh.

$$u(x, t) = c + \frac{5}{19}\sqrt{\frac{11}{19}}[11\tanh^3(k(x - ct - x_0)) - 9\tanh(k(x - ct - x_0))] \tag{4.4}$$

We begin by deriving the finite difference and collocation stencils for the problem, more specifically by constructing numerical schemes which are valid for both the uniform and non-uniform mesh. To determine the efficiency of the numerical simulations on both the uniform and non-uniform mesh, we calculate the l^∞ norm errors at different times and make comparison with the values given by the method in [36].

The spatial discretization of the domain is given by

$$-30 = x_1 < x_2 < ... < x_{N+1} = 30 \tag{4.5}$$

where x_i represents the nodal points and $h_i = x_{i+1} - x_i$ with $h_i > 0$ is the variable spatial step. We define $\{U_i\}_{i=1}^{N+1}$ and $\{u_i\}_{i=1}^{N+1}$ as sequences of numerical and exact solutions respectively in the l^∞ space evaluated at x_i. If E_i represents the absolute error in the numerical solution at x_i where $E_i = |U_i - u_i|$, then the l^∞ norm error at any given time t is given by

$$||E_i||_\infty = \max_{1 \le i \le N+1} |U(x_i, t) - u(x_i, t)| \tag{4.6}$$

where $||.||_\infty$ is the infinity norm.

4.2 Discretization

4.2.1 The K-S equation finite difference stencil

The first and second derivative finite difference approximations on a general three point mesh with grid points $x_{i-1} < x_i < x_{i+1}$ are given by

$$u_x \approx \frac{u_{i+1} - u_{i-1}}{x_{i+1} - x_{i-1}} \tag{4.7}$$

and

$$u_{xx} \approx \frac{2}{h_i + h_{i+1}} \left[\frac{u_{i+1} - u_i}{h_{i+1}} - \frac{u_i - u_{i-1}}{h_i} \right]. \tag{4.8}$$

respectively. The five point finite difference scheme for the fourth derivative on a general mesh with grid points $x_{i-2} < x_{i-1} < x_i < x_{i+1} < x_{i+2}$ is given by

$$u_{xxxx} \approx au_{i-2} + bu_{i-1} + cu_i + cu_{i+1} + eu_{i+2}$$

where the coefficients are given by

$$\begin{aligned}
a &= \tfrac{24}{(x_{i-1}-x_{i-2})(x_i-x_{i-2})(x_{i+1}-x_{i-2})(x_{i+2}-x_{i-2})} \\
b &= -\tfrac{24}{(x_{i-1}-x_{i-2})(x_i-x_{i-1})(x_{i+1}-x_{i-1})(x_{i+2}-x_{i-1})} \\
c &= \tfrac{24}{(x_i-x_{i-1})(x_i-x_{i-2})(x_{i+1}-x_i)(x_{i+2}-x_i)} \\
d &= -\tfrac{24}{(x_{i+1}-x_i)(x_{i+1}-x_{i-1})(x_{i+1}-x_{i-2})(x_{i+2}-x_{i+1})} \\
e &= \tfrac{24}{(x_{i+2}-x_{i+1})(x_{i+2}-x_i)(x_{i+2}-x_{i-1})(x_{i+2}-x_{i-2})}.
\end{aligned} \tag{4.9}$$

We discretize the K-S equation (4.1) using the finite difference schemes (4.7), (4.8) and (4.9) to obtain

$$\begin{aligned}
&\tfrac{u_i^{n+1}-u_i^n}{\delta t} + \left(\tfrac{u_{i+1}^{n+\frac{1}{2}}+u_i^{n+\frac{1}{2}}+u_{i-1}^{n+\frac{1}{2}}}{3}\right)\left(\tfrac{u_{i+1}^{n+\frac{1}{2}}-u_{i-1}^{n+\frac{1}{2}}}{x_{i+1}-x_{i-1}}\right)\\
&+\tfrac{2}{x_{i+1}-x_{i-1}}\left(\tfrac{u_{i+1}^{n+\frac{1}{2}}-u_i^{n+\frac{1}{2}}}{x_{i+1}-x_i} - \tfrac{u_i^{n+\frac{1}{2}}-u_{i-1}^{n+\frac{1}{2}}}{x_i-x_{i-1}}\right)\\
&+\left(au_{i-2}^{n+\frac{1}{2}} + bu_{i-1}^{n+\frac{1}{2}} + cu_i^{n+\frac{1}{2}} + du_{i+1}^{n+\frac{1}{2}} + eu_{i+2}^{n+\frac{1}{2}}\right)\\
&= 0
\end{aligned} \tag{4.10}$$

where

$$\begin{aligned}
&u_i^{n+\frac{1}{2}} = \tfrac{u_i^{n+1}+u_i^n}{2}\\
&u^n = u(t_n)\\
&u^{n+1} = u(t_{n+1}).
\end{aligned}$$

t_n and t_{n+1} are the current and later times respectively with $\delta t = t_{n+1} - t_n$ representing the time step. We simplify equation (4.10) and obtain the difference equation

$$\begin{aligned}
&u_{i-2}^{n+1}\left[\tfrac{a\delta t}{2}\right]+\\
&u_{i-1}^{n+1}\Bigg[\tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_i-x_{i-1})}+\tfrac{b\delta t}{2}\\
&\qquad+\tfrac{\delta t(u_{i+1}^{n+1}-u_{i-1}^{n+1}-u_{i-1}^{n+1}+u_{i+1}^{n}-u_{i-1}^{n})}{12(x_{i+1}-x_{i-1})}-\tfrac{\delta t(u_{i+1}^{n}+u_{i}^{n}+u_{i+1}^{n})}{12(x_{i+1}-x_{i-1})}\Bigg]+\\
&u_{i}^{n+1}\Bigg[1-\tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}-\tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_i-x_{i-1})}\\
&\qquad+\tfrac{c\delta t}{2}+\tfrac{\delta t(u_{i+1}^{n+1}-u_{i-1}^{n+1}+u_{i+1}^{n}-u_{i-1}^{n})}{6}\Bigg]+\\
&u_{i+1}^{n+1}\Bigg[\tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}+\tfrac{d\delta t}{2}+\tfrac{\delta t(u_{i+1}^{n+1}-u_{i-1}^{n+1}+u_{i+1}^{n}-u_{i-1}^{n})}{6}\\
&\qquad+\tfrac{\delta t(u_{i-1}^{n}-u_{i}^{n}+u_{i+1}^{n})}{2(x_{i+1}-x_{i-1})}\Bigg]+\\
&u_{i+2}^{n+1}\left[\tfrac{e\delta t}{2}\right]\\
&=u_i^n-\tfrac{\delta t(u_{i-1}^{n}+u_{i}^{n}+u_{i+1}^{n})}{6}\tfrac{(u_{i+1}^{n}-u_{i-1}^{n})}{2(x_{i+1}-x_{i-1})}\\
&-\tfrac{\delta t(u_{i+1}^{n}-u_{i-1}^{n})}{2(x_{i+1}-x_{i-1})}-\delta t\left[\tfrac{au_{i-2}^{n}+bu_{i-1}^{n}+cu_{i}^{n}+du_{i+1}^{n}+eu_{i+2}^{n}}{2}\right].
\end{aligned}\tag{4.11}$$

with

$$\begin{aligned}
a&=\tfrac{24}{(x_{i-1}-x_{i-2})(x_i-x_{i-2})(x_{i+1}-x_{i-2})(x_{i+2}-x_{i-2})}\\
b&=-\tfrac{24}{(x_{i-1}-x_{i-2})(x_i-x_{i-1})(x_{i+1}-x_{i-1})(x_{i+2}-x_{i-1})}\\
c&=\tfrac{24}{(x_i-x_{i-1})(x_i-x_{i-2})(x_{i+1}-x_i)(x_{i+2}-x_i)}\\
d&=-\tfrac{24}{(x_{i+1}-x_i)(x_{i+1}-x_{i-1})(x_{i+1}-x_{i-2})(x_{i+2}-x_{i+1})}\\
e&=\tfrac{24}{(x_{i+2}-x_{i+1})(x_{i+2}-x_i)(x_{i+2}-x_{i-1})(x_{i+2}-x_{i-2})}.
\end{aligned}\tag{4.12}$$

The difference equation (4.11) can be written as

$$f_iu_{i-2}^{n+1}+g_iu_{i-1}^{n+1}+h_iu_{i}^{n+1}+m_iu_{i+1}^{n+1}+r_iu_{i+2}^{n+1}=D_i^n\tag{4.13}$$

where

$$\begin{aligned}
D_i^n=u_i^n&-\tfrac{\delta t(u_{i-1}^{n}+u_{i}^{n}+u_{i+1}^{n})}{6}\tfrac{(u_{i+1}^{n}-u_{i-1}^{n})}{2(x_{i+1}-x_{i-1})}-\tfrac{\delta t(u_{i+1}^{n}-u_{i-1}^{n})}{2(x_{i+1}-x_{i-1})}\\
&-\delta t\left[\tfrac{au_{i-2}^{n}+bu_{i-1}^{n}+cu_{i}^{n}+du_{i+1}^{n}+eu_{i+2}^{n}}{2}\right].
\end{aligned}\tag{4.14}$$

and f_i, g_i, h_i, m_i, r_i represents the leading coefficients given by

$$\begin{aligned}
f_i &= \tfrac{a\delta t}{2} \\
g_i &= \tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_i-x_{i-1})} + \tfrac{b\delta t}{2} - \tfrac{\delta t(u_{i+1}^n+u_i^n+u_{i+1}^n)}{12(x_{i+1}-x_{i-1})} \\
&\quad + \tfrac{\delta t(u_{i+1}^{n+1}-u_{i-1}^{n+1}-u_{i-1}^{n+1}+u_{i+1}^n-u_{i-1}^n)}{12(x_{i+1}-x_{i-1})} \\
h_i &= 1 - \tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} - \tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_i-x_{i-1})} \\
&\quad + \tfrac{c\delta t}{2} + \tfrac{\delta t(u_{i+1}^{n+1}-u_{i-1}^{n+1}+u_{i+1}^n-u_{i-1}^n)}{6} \\
m_i &= \tfrac{\delta t}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} + \tfrac{d\delta t}{2} + \tfrac{\delta t(u_{i+1}^{n+1}-u_{i-1}^{n+1}+u_{i+1}^n-u_{i-1}^n)}{6} \\
&\quad + \tfrac{\delta t(u_{i-1}^n-u_i^n+u_{i+1}^n)}{2(x_{i+1}-x_{i-1})} \\
r_i &= \tfrac{e\delta t}{2}
\end{aligned} \tag{4.15}$$

for $i = 3, ..., N-1$. The system (4.13) consists of $N-3$ equations in $N+1$ unknowns. The boundary conditions (4.3) gives us the remaining four equations giving a consistent system of $N+1$ equations in $N+1$ unknowns. From the boundary conditions we have

$$\begin{aligned}
u_1 &= \sigma \\
u_2 - u_1 &= 0 \\
u_{N+1} &= \omega \\
u_{N+1} - u_N &= 0.
\end{aligned} \tag{4.16}$$

In matrix form, we write the finite difference system as follows,

$$\mathbf{CU} = \mathbf{D} \tag{4.17}$$

where

$$\mathbf{C}=\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0 & 0\\
1 & -1 & 0 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0 & 0\\
f_3 & g_3 & h_3 & m_3 & r_3 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0 & 0\\
0 & f_4 & g_4 & h_4 & m_4 & r_4 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & \dots & f_i & g_i & h_i & m_i & r_i & \dots & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & f_{N-2} & g_{N-2} & h_{N-2} & m_{N-2} & r_{N-2} & 0\\
0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & f_{N-1} & g_{N-1} & h_{N-1} & m_{N-1} & r_{N-1}\\
0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & -1 & 1\\
0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$\mathbf{U}=\begin{pmatrix} u_1^{n+1}\\ u_2^{n+1}\\ u_3^{n+1}\\ \vdots\\ u_i^{n+1}\\ \vdots\\ u_{N-1}^{n+1}\\ u_N^{n+1}\\ u_{N+1}^{n+1}\end{pmatrix} \qquad \text{and} \qquad \mathbf{D}=\begin{pmatrix} \sigma\\ 0\\ D_3^n\\ \vdots\\ D_i^n\\ \vdots\\ D_{N-1}^n\\ 0\\ \omega\end{pmatrix}$$

4.2.2 The K-S equation Hermite collocation stencil

We begin by changing the notation used to define the mesh (4.5) and the interval length for our convenience. We represent the nodal points x_i by X_i and the interval length h_i by H_i. We place four collocation points defined by $X_{ij} = X_i + \rho_j H_i$ in each subinterval of the domain and define the local variable of the collocation points by

$$s_j^{(i)} = \frac{X_{ij} - X_i}{H_i}, \tag{4.18}$$

for $i = 1, ..., N$ and $j = 1, 2, 3, 4$. From here on, we drop superscript in $s_j^{(i)}$ for convenience.

We discretize the time derivative with finite difference and apply Crank-Nicolson scheme to equation (4.1) and get

$$\begin{aligned} &\left[\tfrac{u^{n+1}-u^n}{\delta t}\right] + \left[\tfrac{(uu_x)^{n+1}+(uu_x)^n}{2}\right] + \left[\tfrac{(u_{xx})^{n+1}+(u_{xx})^n}{2}\right] \\ &+\left[\tfrac{(u_{xxxx})^{n+1}+(u_{xxxx})^n}{2}\right] = 0 \end{aligned} \tag{4.19}$$

where δt is the time step. In order to linearize the non-linear term $(uu_x)^{n+1}$, we use the linearization form given by Rubin and Graves [29],

$$(uu_x)^{n+1} = u^{n+1}u_x^n + u^n u_x^{n+1} - (uu_x)^n. \tag{4.20}$$

Substituting equation (4.20) into equation (4.1), we get

$$\begin{aligned} &\left[\tfrac{u^{n+1}-u^n}{\delta t}\right] + \left[\tfrac{u^{n+1}u_x^n+u^n u_x^{n+1}}{2}\right] + \left[\tfrac{(u_{xx})^{n+1}+(u_{xx})^n}{2}\right] \\ &+\left[\tfrac{(u_{xxxx})^{n+1}+(u_{xxxx})^n}{2}\right] = 0. \end{aligned} \tag{4.21}$$

We rearrange the terms in equation (4.21)and simplify to get

$$\begin{aligned} &u^{n+1} + \tfrac{\delta t}{2}\left[u^{n+1}u_x^n + u^n u_x^{n+1} + (u_{xx})^{n+1} + (u_{xxxx})^{n+1}\right] \\ &= u^n - \tfrac{\delta t}{2}\left[(u_{xx})^n + (u_{xxxx})^n\right]. \end{aligned} \tag{4.22}$$

We then evaluate the Hermite polynomial approximation (3.14) at the four internal collocation points in each subinterval $[X_i, X_{i+1}]$ to give

$$\begin{aligned} U(x,t) &= U_i(t)L_{0,0}(s_j) + U_{x,i}(t)H_i(t)L_{0,1}(s_j) + U_{xx,i}(t)H_i^2(t)L_{0,2}(s_j) \\ &+ U_{i,xxx}(t)H_i^3(t)L_{0,3}(s_j) + U_{i+1}(t)L_{1,0}(s_j) \\ &+ U_{x,i+1}(t)H_i(t)L_{1,1}(s_j) + U_{i+1,xx}(t)H_i^2(t)L_{1,2}(s_j) \\ &+ U_{i+1,xxx}(t)H_i^3(t)L_{1,3}(s_j). \end{aligned} \tag{4.23}$$

We use the notation

$$\begin{aligned} L'(s) &= \frac{dL(s)}{ds} \\ L''(s) &= \frac{d^2L(s)}{ds} \\ L^{(iv)}(s) &= \frac{d^4L(s)}{ds^4} \end{aligned} \tag{4.24}$$

in the expressions for the first, second and fourth derivatives of U as obtained from (3.16). Similarly, we evaluate these at the four internal collocations points in each subinterval to get

$$\begin{aligned}
U_x(x,t) &= \frac{1}{H_i(t)}\Big[U_i(t)L'_{0,0}(s_j) + U_{x,i}(t)H_i(t)L'_{0,1}(s_j) \\
&+ U_{xx,i}(t)H_i^2(t)L'_{0,2}(s_j) + U_{i,xxx}(t)H_i^3(t)L'_{0,3}(s_j) \\
&+ U_{i+1}(t)L'_{1,0}(s_j) + U_{x,i+1}(t)H_i(t)L'_{1,1}(s_j) \\
&+ U_{i+1,xx}(t)H_i^2(t)L'_{1,2}(s_j) + U_{i+1,xxx}(t)H_i^3(t)L'_{1,3}(s_j)\Big]
\end{aligned} \tag{4.25}$$

$$\begin{aligned}
U_{xx}(x,t) &= \frac{1}{H_i^2(t)}\Big[U_i(t)L''_{0,0}(s_j) + U_{x,i}(t)H_i(t)L''_{0,1}(s_j) \\
&+ U_{xx,i}(t)H_i^2(t)L''_{0,2}(s_j) + U_{i,xxx}(t)H_i^3(t)L''_{0,3}(s_j) \\
&+ U_{i+1}(t)L''_{1,0}(s_j) + U_{x,i+1}(t)H_i(t)L''_{1,1}(s_j) \\
&+ U_{i+1,xx}(t)H_i^2(t)L''_{1,2}(s_j) + U_{i+1,xxx}(t)H_i^3(t)L''_{1,3}(s_j)\Big]
\end{aligned} \tag{4.26}$$

$$\begin{aligned}
U_{xxxx}(x,t) &= \frac{1}{H_i^4(t)}\Big[U_i(t)L^{(iv)}_{0,0}(s_j) + U_{x,i}(t)H_i(t)L^{(iv)}_{0,1}(s_j) \\
&+ U_{xx,i}(t)H_i^2(t)L^{(iv)}_{0,2}(s_j) + U_{i,xxx}(t)H_i^3(t)L^{(iv)}_{0,3}(s_j) \\
&+ U_{i+1}(t)L^{(iv)}_{1,0}(s_j) + U_{x,i+1}(t)H_i(t)L^{(iv)}_{1,1}(s_j) \\
&+ U_{i+1,xx}(t)H_i^2(t)L^{(iv)}_{1,2}(s_j) + U_{i+1,xxx}(t)H_i^3(t)L^{(iv)}_{1,3}(s_j)\Big].
\end{aligned} \tag{4.27}$$

Substituting the expressions (4.23), (4.25), (4.26) and (4.27) into (4.22), we obtain the following difference equation,

$$\begin{aligned}
&\beta^{(i)}_{j1}U^{n+1}_i + \beta^{(i)}_{j2}U^{n+1}_{x,i} + \beta^{(i)}_{j3}U^{n+1}_{xx,i} + \beta^{(i)}_{j4}U^{n+1}_{xxx,i} + \beta^{(i)}_{j5}U^{n+1}_{i+1} + \\
&\beta^{(i)}_{j6}U^{n+1}_{x,i+1} + \beta^{(i)}_{j7}U^{n+1}_{xx,i+1} + \beta^{(i)}_{j8}U^{n+1}_{xxx,i+1} = \Psi^n_{ij}
\end{aligned} \tag{4.28}$$

where

$$\begin{aligned}
\beta_{j1}^{(i)} &= L_{0,0}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n L_{0,0}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n L_{0,0}'(s_j) + \tfrac{\delta t}{2H_i^2(t)} L_{0,0}''(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^4(t)} L_{0,0}^{(iv)}(s_j) \\
\beta_{j2}^{(i)} &= H_i(t) L_{0,1}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n H_i(t) L_{0,1}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n H_i(t) L_{0,1}'(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^2(t)} H_i(t) L_{0,1}''(s_j) + \tfrac{\delta t}{2H_i^4(t)} H_i(t) L_{0,1}^{(iv)}(s_j) \\
\beta_{j3}^{(i)} &= H_i^2(t) L_{0,2}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n H_i^2(t) L_{0,2}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n H_i^2(t) L_{0,2}'(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^2(t)} H_i^2(t) L_{0,2}''(s_j) + \tfrac{\delta t}{2H_i^4(t)} H_i^2(t) L_{0,2}^{(iv)}(s_j) \\
\beta_{j4}^{(i)} &= H_i^3(t) L_{0,3}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n H_i^3(t) L_{0,3}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n H_i^3(t) L_{0,3}'(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^2(t)} H_i^3(t) L_{0,3}''(s_j) + \tfrac{\delta t}{2H_i^4(t)} H_i^3(t) L_{0,3}^{(iv)}(s_j) \\
\beta_{j5}^{(i)} &= L_{1,0}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n L_{1,0}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n L_{1,0}'(s_j) + \tfrac{\delta t}{2H_i^2(t)} L_{1,0}''(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^4(t)} L_{1,0}^{(iv)}(s_j) \\
\beta_{j6}^{(i)} &= H_i(t) L_{1,1}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n H_i(t) L_{1,1}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n H_i(t) L_{1,1}'(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^2(t)} H_i(t) L_{1,1}''(s_j) + \tfrac{\delta t}{2H_i^4(t)} H_i(t) L_{1,1}^{(iv)}(s_j) \\
\beta_{j7}^{(i)} &= H_i^2(t) L_{1,2}(s_j) + \tfrac{\delta t}{2} U_{x,i}^n H_i^2(t) L_{1,2}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n H_i^2(t) L_{1,2}'(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^2(t)} H_i^2(t) L_{1,2}''(s_j) + \tfrac{\delta t}{2H_i^4(t)} H_i^2(t) L_{1,2}^{(iv)}(s_j) \\
\beta_{j8}^{(i)} &= H_i^3(t) L_{1,3}(s) + \tfrac{\delta t}{2} U_{x,i}^n H_i^3(t) L_{1,3}(s_j) + \tfrac{\delta t}{2H_i(t)} U_i^n H_i^3(t) L_{1,3}'(s_j) \\
&\quad + \tfrac{\delta t}{2H_i^2(t)} H_i^3(t) L_{1,3}''(s_j) + \tfrac{\delta t}{2H_i^4(t)} H_i^3(t) L_{1,3}^{(iv)}(s_j)
\end{aligned} \tag{4.29}$$

and

$$\begin{aligned}
\Psi_{ij}^n &= U_i^n(t) L_{0,0}(s_j) + U_{x,i}^n(t) H_i(t) L_{0,1}(s_j) + U_{xx,i}^n(t) H_i^2(t) L_{0,2}(s_j) \\
&\qquad + U_{i,xxx}^n(t) H_i^3(t) L_{0,3}(s_j) + U_{i+1}^n(t) L_{1,0}(s_j) + U_{x,i+1}^n(t) H_i(t) L_{1,1}(s_j) \\
&\qquad + U_{i+1,xx}^n(t) H_i^2(t) L_{1,2}(s_j) + U_{i+1,xxx}^n(t) H_i^3(t) L_{1,3}(s_j) \\
-\tfrac{\delta t}{2H_i^2} &\Big[U_i^n(t) L_{0,0}''(s_j) + U_{x,i}^n(t) H_i(t) L_{0,1}''(s_j) + U_{xx,i}^n(t) H_i^2(t) L_{0,2}''(s_j) \\
&\qquad + U_{i,xxx}^n(t) H_i^3(t) L_{0,3}''(s_j) + U_{i+1}^n(t) L_{1,0}''(s_j) + U_{x,i+1}^n(t) H_i(t) L_{1,1}''(s_j) \\
&\qquad + U_{i+1,xx}^n(t) H_i^2(t) L_{1,2}''(s_j) + U_{i+1,xxx}^n(t) H_i^3(t) L_{1,3}''(s_j) \Big] \\
-\tfrac{\delta t}{2H_i^4} &\Big[U_i^n(t) L_{0,0}^{(iv)}(s_j) + U_{x,i}^n(t) H_i(t) L_{0,1}^{(iv)}(s_j) + U_{xx,i}^n(t) H_i^2(t) L_{0,2}^{(iv)}(s_j) \\
&\qquad + U_{i,xxx}^n(t) H_i^3(t) L_{0,3}^{(iv)}(s_j) + U_{i+1}^n(t) L_{1,0}^{(iv)}(s_j) + U_{x,i+1}^n(t) H_i(t) L_{1,1}^{(iv)}(s_j) \\
&\qquad + U_{i+1,xx}^n(t) H_i^2(t) L_{1,2}^{(iv)}(s_j) + U_{i+1,xxx}^n(t) H_i^3(t) L_{1,3}^{(iv)}(s_j) \Big].
\end{aligned} \tag{4.30}$$

From the boundary conditions (4.3), we obtain

$$\begin{aligned} U(x_1) &= \sigma \\ U_x(x_1) &= \beta \\ U(x_{N+1}) &= \omega \\ U_x(x_{N+1}) &= \zeta \end{aligned} \tag{4.31}$$

which results in a consistent system of $4N+4$ equations in $4N+4$ unknowns. In matrix form we have

$$\mathbf{A}\mathbf{U}^{n+1} = \mathbf{B}^n \tag{4.32}$$

where

$$\mathbf{A} = \left(\begin{array}{cccccccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta_{11}^{(1)} & \beta_{12}^{(1)} & \beta_{13}^{(1)} & \beta_{14}^{(1)} & \beta_{15}^{(1)} & \beta_{16}^{(1)} & \beta_{17}^{(1)} & \beta_{18}^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta_{21}^{(1)} & \beta_{22}^{(1)} & \beta_{23}^{(1)} & \beta_{24}^{(1)} & \beta_{25}^{(1)} & \beta_{26}^{(1)} & \beta_{27}^{(1)} & \beta_{28}^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta_{31}^{(1)} & \beta_{32}^{(1)} & \beta_{33}^{(1)} & \beta_{34}^{(1)} & \beta_{35}^{(1)} & \beta_{36}^{(1)} & \beta_{37}^{(1)} & \beta_{38}^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta_{41}^{(1)} & \beta_{42}^{(1)} & \beta_{43}^{(1)} & \beta_{44}^{(1)} & \beta_{45}^{(1)} & \beta_{46}^{(1)} & \beta_{47}^{(1)} & \beta_{48}^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta_{11}^{(2)} & \beta_{12}^{(2)} & \beta_{13}^{(2)} & \beta_{14}^{(2)} & \beta_{15}^{(2)} & \beta_{16}^{(2)} & \beta_{17}^{(2)} & \beta_{18}^{(2)} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta_{21}^{(2)} & \beta_{22}^{(2)} & \beta_{23}^{(2)} & \beta_{24}^{(2)} & \beta_{25}^{(2)} & \beta_{26}^{(2)} & \beta_{27}^{(2)} & \beta_{28}^{(2)} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta_{31}^{(2)} & \beta_{32}^{(2)} & \beta_{33}^{(2)} & \beta_{34}^{(2)} & \beta_{35}^{(2)} & \beta_{36}^{(2)} & \beta_{37}^{(2)} & \beta_{38}^{(2)} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta_{41}^{(2)} & \beta_{42}^{(2)} & \beta_{43}^{(2)} & \beta_{44}^{(2)} & \beta_{45}^{(2)} & \beta_{46}^{(2)} & \beta_{47}^{(2)} & \beta_{48}^{(2)} & 0 & 0 & 0 & 0 \\
\ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & ... & 0 & 0 & 0 & 0 & \beta_{11}^{(N)} & \beta_{12}^{(N)} & \beta_{13}^{(N)} & \beta_{14}^{(N)} & \beta_{15}^{(N)} & \beta_{16}^{(N)} & \beta_{17}^{(N)} & \beta_{18}^{(N)} \\
0 & 0 & 0 & ... & 0 & 0 & 0 & 0 & \beta_{21}^{(N)} & \beta_{22}^{(N)} & \beta_{23}^{(N)} & \beta_{24}^{(N)} & \beta_{25}^{(N)} & \beta_{26}^{(N)} & \beta_{27}^{(N)} & \beta_{28}^{(N)} \\
0 & 0 & 0 & ... & 0 & 0 & 0 & 0 & \beta_{31}^{(N)} & \beta_{32}^{(N)} & \beta_{33}^{(N)} & \beta_{34}^{(N)} & \beta_{35}^{(N)} & \beta_{36}^{(N)} & \beta_{37}^{(N)} & \beta_{38}^{(N)} \\
0 & 0 & 0 & ... & 0 & 0 & 0 & 0 & \beta_{41}^{(N)} & \beta_{42}^{(N)} & \beta_{43}^{(N)} & \beta_{44}^{(N)} & \beta_{45}^{(N)} & \beta_{46}^{(N)} & \beta_{47}^{(N)} & \beta_{48}^{(N)} \\
0 & 0 & 0 & ... & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & ... & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}\right)$$

$$\mathbf{U}^{n+1} = \begin{pmatrix} U_1^{n+1} \\ U_{x,1}^{n+1} \\ U_{xx,1}^{n+1} \\ U_{xxx,1}^{n+1} \\ \vdots \\ U_N^{n+1} \\ U_{x,N}^{n+1} \\ U_{xx,N}^{n+1} \\ U_{xxx,N}^{n+1} \\ U_{N+1}^{n+1} \\ U_{x,N+1}^{n+1} \\ U_{xx,N+1}^{n+1} \\ U_{xxx,N+1}^{n+1} \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} \sigma \\ \beta \\ \Psi_{11}^n \\ \Psi_{12}^n \\ \Psi_{13}^n \\ \Psi_{14}^n \\ \vdots \\ \Psi_{N1}^n \\ \Psi_{N2}^n \\ \Psi_{N3}^n \\ \Psi_{N4}^n \\ \omega \\ \zeta \end{pmatrix}$$

4.3 Uniform Mesh Results

4.3.1 Finite Difference Method

Figures 4.1 and 4.2 shows the numerical solutions of the problem for $N = 100$ and $\delta t = 0.001$ at $t = 4$ using finite difference method on a uniform mesh. We can observe in Figure 4.1 that the numerical solution does not track the exact solution and has a tendency to oscillate on a section of the solution. Figure 4.2 shows the behaviour of the absolute error at $t = 4$.

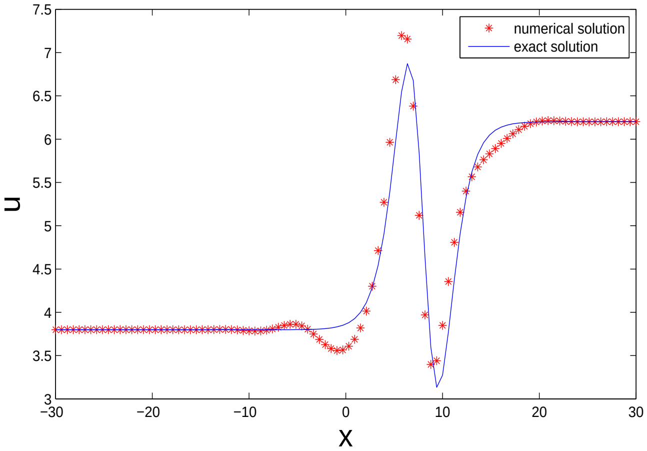

Figure 4.1: Finite difference method, uniform mesh, behaviour of numerical solution of K-S problem at $t = 4$, $N = 100$ and $\delta t = 0.001$

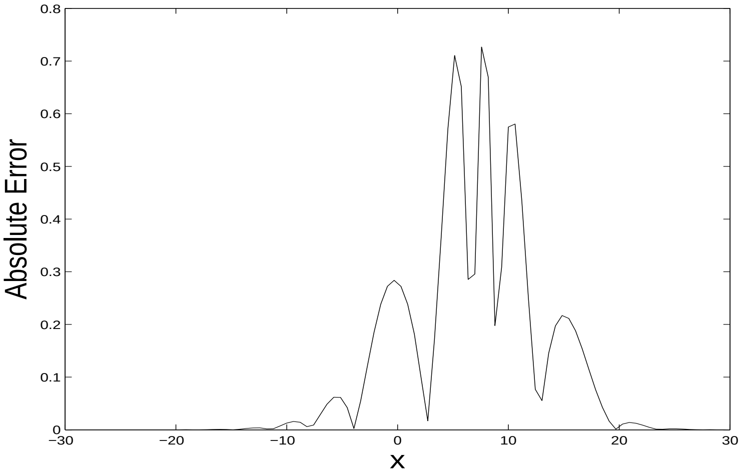

Figure 4.2: Finite difference method, uniform mesh, absolute error in the numerical solution of K-S problem at $t = 4$, $N = 100$ and $\delta t = 0.001$

4.3.2 Hermite Collocation Method

Figures 4.3 and 4.4 shows the behaviour of the numerical solution and the absolute error respectively of the K-S equation on a uniform mesh using Hermite collocation method at $t = 4$ with $N = 100$ and $\delta t = 0.001$. In figure 4.3, we observe that the numerical solution tracks the exact solution with the absolute error variation as shown in figure 4.4.

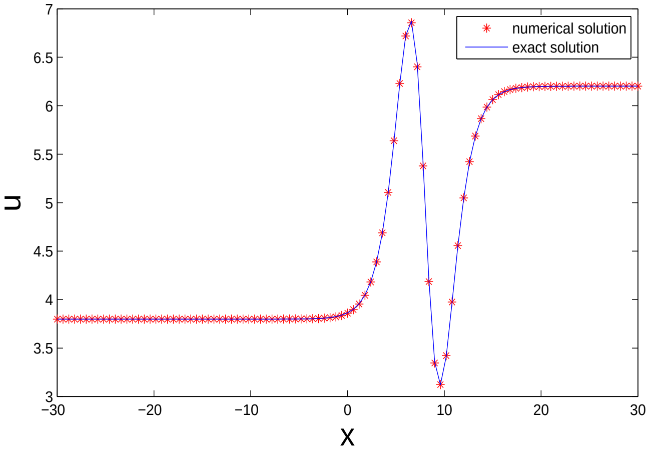

Figure 4.3: Hermite collocation method, uniform mesh, numerical solution behaviour of K-S problem at $t = 4$ with $N = 100$ and $\delta t = 0.001$

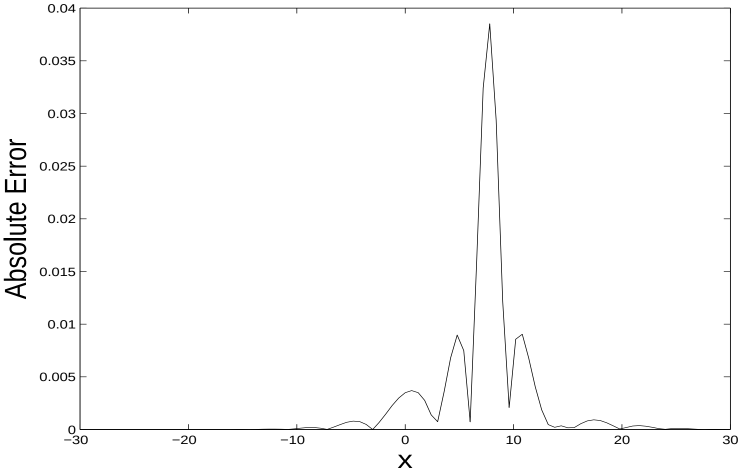

Figure 4.4: Hermite collocation method, uniform mesh, absolute error in numerical solution of K-S problem at $t = 1$, $N = 100$ and $\delta t = 0.001$

Table 4.1: Comparison of maximum pointwise errors for K-S problem at different times with $\delta t = 0.001$, $N = 100$ and $\alpha = 8$

Time	Finite difference	Hermite collocation	Method in [36]
0.5	2.821×10^{-1}	4.4×10^{-3}	1.03619×10^{-3}
1	4.66×10^{-1}	7.6×10^{-3}	1.63762×10^{-3}
1.5	6.029×10^{-1}	1.05×10^{-2}	2.07273×10^{-3}
2	7.023×10^{-1}	1.49×10^{-2}	2.48375×10^{-3}
2.5	7.66×10^{-1}	1.98×10^{-2}	2.79434×10^{-3}
3	7.931×10^{-1}	2.45×10^{-2}	3.00439×10^{-3}
3.5	7.814×10^{-1}	3.06×10^{-2}	3.16038×10^{-3}
4	7.285×10^{-1}	3.85×10^{-2}	3.43704×10^{-3}

Table 4.2: Maximum pointwise errors for K-S problem on a uniform mesh using collocation method with $T = 1$ and $\delta t = 0.001$

Number of subintervals(N)	20	40	80	160
Maximum pointwise error	$9.997\ \times10^{-2}$	2.18×10^{-2}	9×10^{-3}	5.4×10^{-3}

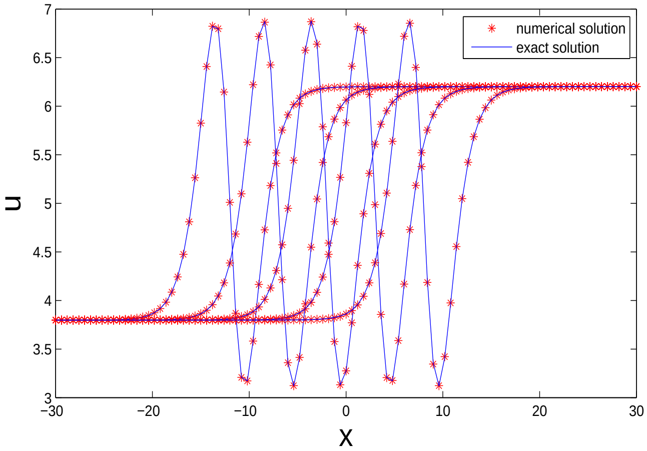

Figure 4.5: Hermite collocation method, uniform mesh, numerical solutions behaviour of K-S problem with $N = 100$, $\delta t = 0.001$ up to final time $T = 4$

Table 4.1 compares the maximum absolute errors produced in the numerical solution of the K-S equation by the finite difference method and Hermite collocation method with the method in [36]. In table 4.2, we can see that the maximum absolute error decreases with the increase in the number of subintervals N for the collocation method on a uniform mesh. Figure 4.5 shows the solution obtained by the collocation method on a uniform mesh for time $t = 0, 1, 2, 3, 4$. The movement of the solution is from left to right as time increases and the solution tracks the exact solution with no oscillations. We also observe that the concentration of mesh points is higher in the flatter regions of the solution profile in comparison to the concentration in the steeper region.

4.4 Non-uniform Mesh Results

4.4.1 Finite difference method

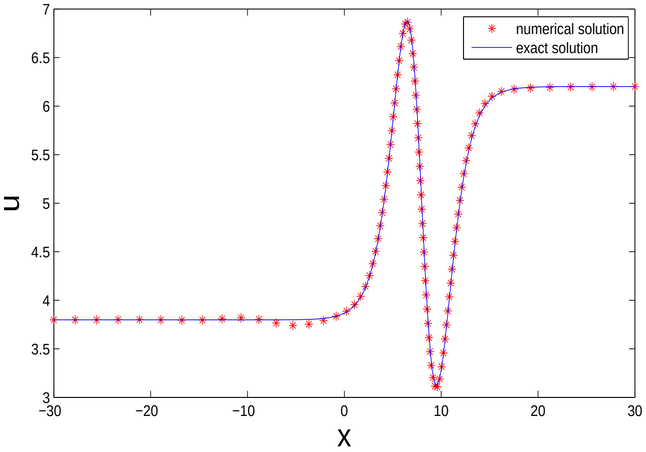

Figure 4.6: Finite difference method, non-uniform mesh, numerical solution behaviour of K-S problem at $t = 4$ with $N = 100$, $\delta t = 0.001$, $\tau = 2 \times 10^{-2}$ and $\alpha = 15$

Figures 4.6 and 4.7 shows the numerical solutions of the K-S equation on non-uniform mesh using finite difference method for $N = 100$, $\delta t = 0.001$, $\tau = 2\times10^{-2}$ and $\alpha = 15$ at $t = 4$. Figure 4.6 is the numerical solution profile for the conditions which give the minimum value of the absolute errors at time $t = 4$. Figure 4.7 shows the corresponding absolute error behaviour in determining the numerical solution.

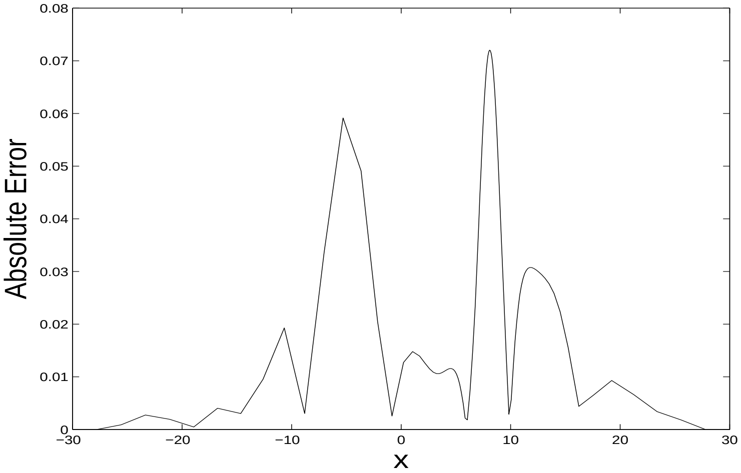

Figure 4.7: Finite difference method, non-uniform mesh, absolute error in the numerical solution for $N = 100$, $\delta t = 0.001$, $\tau = 2 \times 10^{-2}$ and $\alpha = 15$ at $t = 4$

4.4.2 Hermite collocation method

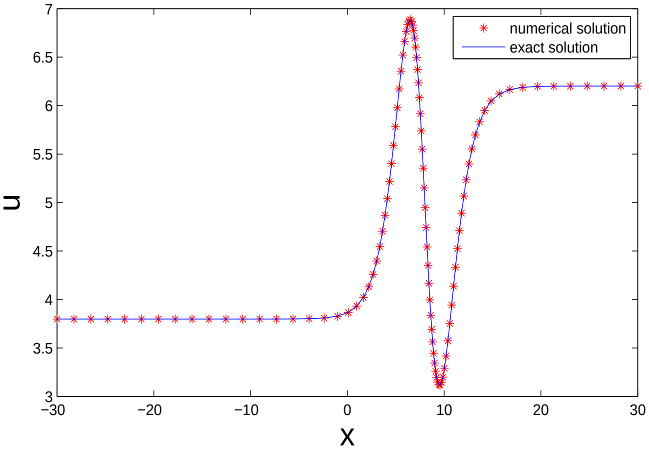

Figure 4.8: Hermite collocation method, non-uniform mesh, numerical solution behaviour of K-S problem at $t = 4$ with $N = 100$, $\delta t = 0.001$, $\tau = 2 \times 10^{-2}$ and $\alpha = 8$

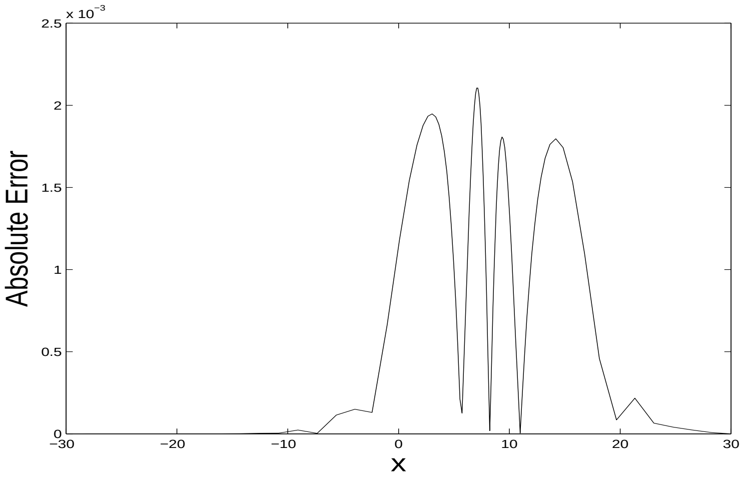

Figure 4.9: Hermite collocation method, non-uniform mesh, absolute error in numerical solution of K-S problem at $t = 4$ with $N = 100$, $\delta t = 0.001$, $\tau = 2 \times 10^{-2}$ and $\alpha = 8$

Figures 4.8 and 4.9 shows the numerical solution profile and the behaviour of the maximum absolute error respectively at $t = 4$ with $N = 100$, $\delta t = 0.001$ and $\alpha = 8$. In Figure 4.8, we observe that the numerical solution is able to track the exact solution and the distribution of mesh points is almost equal along the solution profile which enables resolution of the solution with minimum errors. Table 4.3 compares finite difference method and collocation method on a non-uniform mesh at different values of time with optimum values of α and τ being used for each method. The time step is $\delta t = 0.001$ and $N = 100$. We also observe in table 4.4 that the maximum pointwise error decreases with increase in the number of partitions N.

Figure 4.10 shows the numerical solution profiles produced by the adaptive collocation method for time $t = 0, 1, 2, 3, 4$. We observe that the solution moves from left to right as time progresses. The mesh points at different times keep on tracking the solution profile and maintain an almost equal distribution along the profile up to final time $T = 4$. Figure 4.11 shows the paths taken by the mesh points in tracking the solution profile.

Table 4.3: Comparison of maximum pointwise errors in the numerical solution of the K-S equation on a non-uniform mesh at different times with $\delta t = 0.001$ and $N = 100$

parameters	$\alpha = 8$ $\tau = 2 \times 10^{-2}$	$\alpha = 8$ $\tau = 2 \times 10^{-2}$	
Time	Hermite Collocation	Finite Difference	method in [36]
0.5	9.0×10^{-4}	4.42×10^{-2}	1.03619×10^{-3}
1	1.4×10^{-3}	3.63×10^{-2}	1.63762×10^{-3}
1.5	1.7×10^{-3}	3.78×10^{-2}	2.07273×10^{-3}
2	1.9×10^{-3}	4.22×10^{-2}	2.48375×10^{-3}
2.5	2.0×10^{-3}	4.78×10^{-2}	2.79434×10^{-3}
3	2.1×10^{-3}	5.45×10^{-2}	3.00439×10^{-3}
3.5	2.1×10^{-3}	6.26×10^{-2}	3.16038×10^{-3}
4	2.1×10^{-3}	7.20×10^{-2}	3.43704×10^{-3}

Table 4.4: maximum pointwise errors for K-S equation on a non-uniform mesh using collocation method with $T = 1$, $\delta t = 0.001$, 2×10^{-2} and $\alpha = 8$

Number of subintervals(N)	20	40	80	160
Maximum pointwise error	$2.02\ \times10^{-2}$	2.9×10^{-3}	8.0×10^{-4}	3.2×10^{-2}

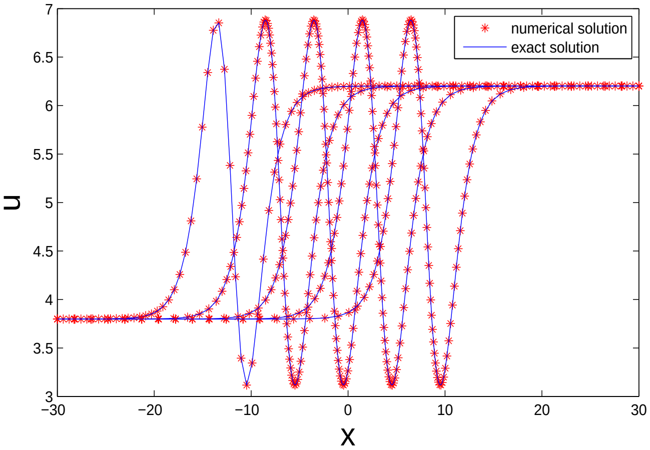

Figure 4.10: Hermite collocation method, non-uniform mesh, numerical solution behaviour of K-S equation up to final time $T = 4$ for $N = 100$, $\delta t = 0.001$, $\tau = 2 \times 10^{-2}$ and $\alpha = 8$

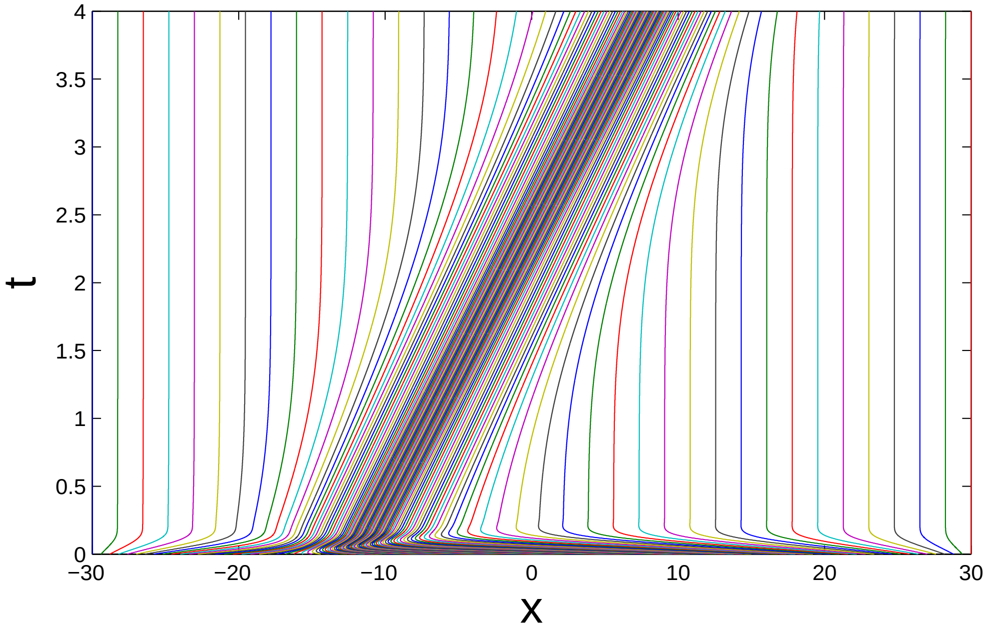

Figure 4.11: Hermite collocation method, mesh trajectories of K-S equation up to final time $T = 4$ with $N = 100$, $\delta t = 0.001$, $\tau = 2 \times 10^{-2}$ and $\alpha = 8$

Chapter 5

Conclusions and Further Work

5.1 Conclusions

In this dissertation, we successfully solved the K-S equation numerically using an adaptive mesh method which is more computationally efficient than the method in [36]. The algorithm for the numerical simulation is based on the rezoning approach which works with the decoupled solution procedure. We then wrote matlab codes based on this approach.

In chapter 2, we started by giving a review of the EP and how it is used to derive MMPDEs whose solutions give the adaptive non-uniform meshes. We illustrated how different choices of the monitor functions in the EP are able to give different meshes. We then came up with a modified monitor function which when smoothed works better with MMPDE4 for the resolution of an adaptive mesh for the numerical solution of the K-S equation.

In chapter 3, we studied the septic Hermite collocation method in detail which is our choice of the discretization method for the K-S equation. We also discretized the equation using finite difference in chapter 4 so as to make a comparison of numerical results on an adapted mesh. We then developed the septic Hermite interpolant which is able to update the approximate solution of the K-S problem and its derivatives up to the third order in each subinterval of the mesh.

This is a necessary step in the rezoning approach algorithm.

Numerical results showed that Hermite collocation method on a non-uniform adaptive mesh is able to improve the accuracy of the numerical solution of the K-S equation. The method is computationally efficient in comparison to the method used in [36] and the adaptive finite difference method. It was also observed that a non-uniform mesh produced by mesh adaptation has the ability to reduce oscillations in the numerical solution. This was seen when we made a comparison of the numerical solution of the problem using finite difference method on a uniform mesh and non-uniform mesh. Another observation we made is the ability of an adaptive mesh to keep track of the region of rapid solution variation in the K-S equation, which is one of the desired properties of an adaptive mesh method.

5.2 Future Work

Whilst septic Hermite collocation method on an adaptive non-uniform mesh has proved to be a superior method than the method in [36], there is still need to hold experiments for different values of the time step and the domain to ascertain whether it is possible to come up with optimum parameters that would improve the computational efficiency of the resulting models.

We also would want to solve the K-S equation using septic Hermite collocation method on an adaptive mesh with the qausi-langrange approach working with the coupled solution procedure and give a comparative analysis of which scenario gives the best results for the problem.

Finally, we make note of the fact that the choice of monitor functions, the MMPDEs and temporal and spatial smoothing parameters are still key issues warranting further study as these are based on an empirical approach.

Bibliography

[1] S. Adjerid, J.E. Flaherty, P.K. Moore, and Y.J. Wang. Higher order adaptive methods for parabolic systems. *Physica D*, 60:94 – 111, 1992.

[2] D.A. Anderson. Adaptive mesh strategies based on grid speeds. *AIAA paper*, pages 83 – 1931, 1983.

[3] M.J. Baines, M.E. Hubbard, P.K. Jimack, and A.C. Jones. Scale-invariant moving finite elements for non-linear partial differential equations in two-dimensions. *Appl. Numer. Math*, 56:230 – 252, 2006.

[4] C.J. Budd, Weizhang Huang, and Robert D. Russell. Moving mesh methods for the problems with blow-up. *SIAM J. Scie. Comput*, 17:305 – 327, 1996.

[5] C.J. Budd and J.F. Williams. Parabolic monge-ampere methods for blow-up problems in several dimensions. *J. Phys. A*, 39:5425 – 5444, 2006.

[6] H.G. Burchard. Splines (with optimal knots) are better. *Appl. Anal*, 3:309 – 319, 1974.

[7] H.D. Ceniceros and T.Y.Hou. An efficient dynamically adaptive mesh for potentially singular solutions. *J. Comput. Phys*, 172:609 – 639, 2001.

[8] Carl deboor. Good approximation by splines with variable knots. In A. Meir and A. Sharma, editors, *Spline functions and approximation theory*, pages 57 – 73. Birkauser Verlag, Baselund Stuttgart, 1973.

[9] Carl deboor. Good approximation by splines with variable knots. ii 363. In G.A. Watson, editor, *Conference on the Numerical Solution of Differential Equations*, pages 12 – 20. Springer Berlin Heidelberg, 1974.

[10] D.S Dodson. *Optimal order approximation by polynomial spline functions. Technical report.* PhD thesis, Purdue University, 1972.

[11] W. Gui and I. Babuska. The h, p and h-p versions of the finite element method in one dimension. part 1. the error analysis of the p-version. *Numerical Mathematics*, 49:577 – 612, 1986.

[12] Sirajul Haq, Nagina Bibi, S.I.A. Tirmizi, and M. Usman. Meshless method of lines for the numerical solution of generalized kuramoto-sivashinsky equation. *Applied Mathematics and Computation*, 217(6):2404 – 2413, 2010.

[13] Weizhang Huang. Anisotropic mesh adaptation and movement. *Adaptive Computations: Theory and Algorithms*, 56:230 – 252, 2006.

[14] Weizhang Huang, Yuhe Ren, and Robert D. Russell. Moving mesh methods based on moving mesh partial differential equations equations. *Journal of Computational Physics*, 113:279 – 290, 1994.

[15] Weizhang Huang, Yuhe Ren, and Robert D. Russell. Moving mesh partial differential equations (mmpdes) based on the equidistribution principle. *SIAM Journal on Numerical Analysis*, 31(3):709 – 730, 1994.

[16] Weizhang Huang and Robert D. Russell. A moving collocation method for solving time dependent partial differential equations. *Appl. Numer. Math*, 20:101 – 116, 1996.

[17] A.G Istratov and V.B. Librovich. Effect of transport processes on the stability of a plane flame front. *J. Appl. Math. Mech.*, 30(3):441 – 557, 1966.

[18] A.H. Khater and R.S. Temah. Numerical solution of the generalised kuramoto-sivashinsky equation by chebyshev spectral collocation methods. *Computers and Mathematics Applications*, 56:1465 – 1472, 2008.

[19] Y. Kuramoto. Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Prog of Theor. Phys.*, 55:356 – 369, 1976.

[20] Y. Kuramoto. Diffusion-induced chaos in reactions systems. *Suppl. Prog. of Theor. Phys.*, 64:346 – 367, 1978.

[21] Y. Kuramoto. Instabily and turbulence of wavefronts in reaction-diffusion systems. *Prog. of Theor. Phys.*, 63:1885 – 1903, 1980.

[22] Y. Kuramoto and T. Tsuzki. Diffusion-induced chaos in reactions systems. *Prog. of Theor. Phys.*, 54:687 – 699, 1975.

[23] Huilin Lai and Changfeng Ma. Lattice boltzmann method for the generalized kuramotosivashinsky equation. *Physica A: Statistical Mechanics and its Applications*, 388(8):1405 – 1412, 2009.

[24] Kam Nang Hareton Leung. Spline collocation for solving time dependent parabolic problems in one space variable. Master's thesis, Simon Fraser University, 1983.

[25] K. Miller and R.N. Miller. Moving finite elements. *I. SIAM J. Numerical Anal*, 18:1019 – 1032, 1981.

[26] R.C. Mittal and Geeta Arora. Quintic b-spline collocation method for numerical solution of the kuramotosivashinsky equation. *Communications in Nonlinear Science and Numerical Simulation*, 15(10):2798 – 2808, 2010.

[27] Y. Qui and D.M. Sloan. Numerical solution of fisher's equation using a moving mesh method. *Journal of Computational Physics*, 146:726 – 746, 1998.

[28] Y. Ren and R.D. Russell. Moving mesh techniques based upon equidistribution and their stability. *SIAM. J. Sci. Stat. Comput*, 13:1265 – 1286, 1992.

[29] S.G. Rubin and R.A. Graves. Cubic spline approximation for problems in fluid mechanics. *NASA TR-436, Washington DC*, 1975.

[30] Robert D. Russell, J.F. William, and X. Xu. Movcol4: A moving mesh code for fourth order time dependent partial differential equations. *SIAM J. Sci. Comput*, 29:197 – 220, 2007.

[31] J.M. Sanz-Serna and I. Christie. A simple adaptive technique for nonlinear wave problems. *Journal of Computational Physics*, 67:348 – 360, 1986.

[32] P. Saucez, A. Vande Wouwer, and P.A. Zegeling. Adaptive mesh methods of lines solutions for the extended fifth order kortweg-devries equation. *J. Comput. Appl. Math*, 183:343 – 387, 2005.

[33] G.I. Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames-i derivation of basic equations. *Acta Astronautica*, 4:1177 – 1206, 1977.

[34] T. Tan. Moving mesh methods for computational fluid dynamics flow and transport. *Amer. Math. Soc.,Providence,RI*, 383 of AMS Contemporary Mathematics:141 – 173, 2005.

[35] H.Z. Tang and T. Tang. Adaptive mesh methods for one-dimensional and two dimensional hyperbolic conservation laws. *SIAM J. Numer. Anal*, 41:487 – 515, 2003.

[36] M. Zarebnia and R. Parvaz. Septic b-spline collocation method for numerical solution of the kuramotosivashinsky equation. *Communications in Nonlinear Science and Numerical Simulation*, 7(3):354 – 358, 2013.

[37] Gentian Zavalani. Fourier spectral collocation methods for the numerical solving of the kuramoto-sivashinsky equation. *American Journal of Numerical Analysis*, 2(3):90 – 97, 2014.