A multi-domain implementation of the pseudo-spectral method and compact finite difference schemes for solving time-dependent differential equations

Dyke Mathale

Supervisor: Prof. M Khumalo

Dr. PG Dlamini

Pure and Applied Mathematics

University of Johannesburg

This dissertation is submitted for the degree of

Masters of Science

January 2019

Declaration

The work presented in this dissertation was carried out at the department of Pure and Applied Mathematics, University of Johannesburg. Unless otherwise stated, it is the original work of the author.

While registered as a candidate for the degree of Masters of science, for which submission is now made, the author has not been registered as a candidate for any other award. This dissertation has not been submitted in whole, or in part, for any other degree.

Dyke Mathale
January 2019

Acknowledgements

I wish to express my sincere gratitude to the following people and organisations who made this dissertation possible:

Firstly, I wold like to thank my supervisor's Prof. M Khumalo and Dr PG. Dlamini for the guidance and support they provided throughout the duration of my work. I also thank the South African National Space Agency (SANSA) for the funding they gave me to undertake my masters studies.

Lastly, I would like to thank my family and friends for their encouragements, love and support.

MATHALE WEDNESDAY 23RD JANUARY, 2019

"If there is a 50-50 chance that something can go wrong, then 9 times out of ten it will."

Paul Harvey

Abstract

In this dissertation, we introduce new numerical methods for solving time-dependant differential equations. These methods involve dividing the domain of the problem into multiple sub domains. The nonlinearity of the differential equations is dealt with by using a Gauss-Seidel like relaxation or quasilinearisation technique. To solve the linearized iteration schemes obtained we use either higher order compact finite difference schemes or spectral collocation methods and we call the resulting methods the multi-domain compact finite difference relaxation method (MD-CFDRM), multi-domain compact finite difference quasilinearisation method (MD-CFDQLM) and multi-domain bivariate spectral quasilinearisation method (MD-BSQLM) respectively.

We test the applicability of these methods in a wide variety of differential equations. The accuracy is compared against other methods as well as other results from literature. The MD-CFDRM is used to solve famous chaotic systems and hyperchaotic systems. Chaotic and hyperchaotic systems are characterized by high sensitivity to small perturbation on initial data and rapidly changing solutions. Such rapid variations in the solution pose tremendous problems to a number of numerical approximations. We modify the CFDs to be able to deal with such systems of equations. We also used the MD-CFDQLM to solve the nonlinear evolution partial differential equations, namely, the Fisher's equation, Burgers-Fisher equation, Burgers-Huxley equation and the coupled Burgers' equations over a large time domain. The main advantage of this approach is that it offers better accuracy on coarser grids which significantly improves the computational speed of the method for large time domain. We also studied the generalized Kuramoto-Sivashinsky (GKS) equations. The KS equations exhibit chaotic behaviour under certain conditions. We used the multi-domain bivariate spectral quasilinearisation method (MD-BSQLM) to approximate the numerical solutions for the generalized KS equations.

Table of contents

List of figures

List of tables

Chapter 1

Preliminaries

1.1 Introduction

Differential equations are used to model numerous natural occurring phenomena emanating from fluid mechanics, mathematical biology, financial mathematics etc. Researchers are often interested in determining long time behaviour of such differential equations. However, most of the existing numerical methods converge slowly over large time intervals, often resulting in inaccurate results. This leads many researchers to start investigating the use of spectral methods in time. The main advantage of the spectral methods is their high accuracy, which also means that the desired accuracy could be achieved with fewer grid points. However, for large time-domain problems the spectral methods can also become less accurate, even with an increase in the number of collocation points. Many researchers have addressed this by using the multi-domain approach. The multi-domain approach assumes that the main interval can be decomposed into a finite number of sub-intervals. It uses the divide-and-conquer philosophy. A significant advantage of the multi-domain approach is that small time domains are considered and hence the accuracy of the solution in each of the sub-intervals is improved and it enhances the accuracy of the approximation significantly. According to Canuto et al [12], the multi-domain approach has three main families, patching method, overlapping and variation methods. In this study we focus on the patching method originally suggested by Orszag [65]. There is quite a substantial number of articles in the literature where the idea of domain decomposition has been applied for both analytical and numerical methods. Examples of analytical multi-domain methods include the: Adomian decomposition method [3, 37, 5], multistage homotopy analysis method [6, 1], multi-stage differential transformation method [28, 32, 63], multi-stage variational iteration method [9, 33, 34], multistage homotopy perturbation methods [16, 17, 15, 79]. Magagula et al [57] used the multi-domain technique

together with the Spectral collocation method to solve non-linear evolution partial differential equations. Examples of multi-domain (see[62, 61, 57])

In this work we implement the multi-domain approach on the spectral method and compact finite differences to solve wide variety of nonlinear time-dependent differential equations which are presented in Chapter 2 - Chapter 4.

To solve the nonlinear differential equations, we first linearize the differential equations. In this dissertation we use the relaxation method [62] and quasi-linearization method [10] to linearize the nonlinear operators of the differential equation.

1.2 Convergence, Consistency, and Stability

A problem in differential equations can rarely be solved analytically, and so often is discretized, resulting in a discrete problem which can be solved in a finite sequence of algebraic operations, a numerical method or scheme is measured on how well the discrete solution U_h approximates the exact solution u of the continuous problem.

Consider the following non-linear differential equation:

$$\frac{\partial u}{\partial t} = f\left(x,t,u,\frac{\partial u}{\partial x},\frac{\partial^2 u}{\partial x^2}\ldots,\frac{\partial^n U}{\partial x^n}\right) \tag{1.1}$$

with boundary conditions

$$\begin{aligned} u(a,t) &= a(t) \\ u(b,t) &= b(t) \end{aligned} \tag{1.2}$$

and initial condition

$$u(x,0) = u_0(x) \tag{1.3}$$

valid in the physical region $\{(t,x)|t \in [t_0,T], x \in [a,b]\}$ The constant n denotes the order of differentiation and f is a function of $u(x,t)$ and its spatial derivatives. Representing the above equations as continuous problem denoted by

$$LU = F \tag{1.4}$$

where

$$LU = \begin{cases} \frac{\partial u}{\partial t} & , \ t \in [t_0, T] \\ u(x_a, t) \\ u(x_b, t) \\ u(x, t_0) \end{cases} \tag{1.5}$$

$$F = \begin{cases} f(x,t,u(x,t),u_x(x,t),\ldots,u^n(x,t)) & , \ t \in [t_0, T] \\ a(t) \\ b(t) \\ u_0(x) \end{cases} \tag{1.6}$$

If the exact solution u is evaluated at grid points $x_i = ih, \ i = 0,1,2,\ldots,N$ and $t_j = jn, \ j = 0,1,2,\ldots,N$, then

$$L_h U_h = F_h \tag{1.7}$$

denotes the corresponding discrete problem of (1.1) where U_h is the approximate solution and $h = \frac{x_b - x_a}{N+1}$ as spatial step-size.

1.2.1 Convergence

The solution U_h of (1.7) converges [71] to the solution u of the continuous problem (1.5) if

$$\|u_h - U_h\| \to 0 \tag{1.8}$$

as $h \to 0$. If there exist constant $k > 0$ and constant $C_o > 0$ that does not depend on k, such that

$$\|u_h - U_h\| \leq C_0 h^k \tag{1.9}$$

then the solution U_h converges and is a convergence of order h^k and the numerical method or scheme has $k - th$ order accuracy.

1.2.2 Consistency and Stability

When the approximated solution U_h of (1.1) is substituted back in the continuous problem, it results in the following equation

$$L_h U_h = \mathbf{F}_h + \Delta f_h \tag{1.10}$$

where Δf_h is the residual term.

The numerical solution giving rise to U_h is consistent with (refss1) if the residual term in (1.10) satisfies $\|\Delta f_h\| \to 0$ as $h \to 0$. Similarly if the inequality

$$\|\Delta f_h\| \leq C_1 h^k \tag{1.11}$$

where $C_1 > 0$ and $k > 0$, then the method has consistency of order h^k.

If the method approximated solution of a boundary value problem or initial value problem is not oversensitive to small changes in boundary or initial conditions the numerical method is said to be stable.

Moreover, the numerical method is stable if there exist $h_0 > 0$ and $\Delta > 0$ such that for any $h < h_0$ and $\varepsilon_h \in \mathbf{F}_h$ such that $\|\varepsilon_h\| < \Delta$, we have

$$L_h w_h = f_h + \Delta\varepsilon_h \tag{1.12}$$

and w_h is one solution that satisfies

$$\|w_h - U_h\| \leq C\|\varepsilon_h\| \tag{1.13}$$

where $C > 0$ does not depend on h.

In Chapter **2**, **3**, and Chapter **4**, we use the Chebyshev spectral collocation and Higher order compact finite difference schemes to approximate the solutions for a number of nonlinear initial and boundary value problems to demonstrate the Convergence, consistency and stability of the proposed numerical techniques. In the next chapter, we demonstrate the implementation of the proposed higher order (Sixth-Order) compact finite difference and the Chebyshev spectral collocation method with Chebyshev-Gauss-Lobatto points.

1.3 Compact finite difference schemes

Compact finite difference schemes (CFDS) are capable of producing higher order accuracy without any increase in the numerical stencil when compared with the traditional finite difference schemes. Recently higher CFDS have become popular in solving differential equations.

In the derivation of the CFDS, a uniform one-dimensional mesh is considered on the region $[a,b]$ consisting of N points: $a = x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_N = b$.

Consider a function $u_i = u(x_i)$ of one variable defined on the real line $\mathbb{R}$ at the nodes with the spatial grid spacing $h = x_i - x_{i-1}$. The procedure of finding the spatial derivatives using CFDS is as follows:

1.3.1 First-order derivative

The first-order derivative at $i-th$ collocation point can be given as follow

$$\alpha_{-1}u'_{i-1} + u'_i + \alpha_1 u'_{i+1} = \frac{\beta_{-2}u_{i-2} + \beta_{-1}u_{i-1} + \beta_0 u_i + \beta_1 u_{i+1} + \beta_2 u_{i+2}}{h} \tag{1.14}$$

where $\alpha_{-1}, \alpha_1, \beta_{-2}, \beta_{-1}, \beta_0, \beta_1$ and β_2 are arbitrary constants. We expand (1.14) using the Taylor series at the i^{th} collocation point with respect to the step size h to get

$$\begin{aligned}
0 = {} & (-\beta_{-2} - \beta_{-1} - \beta_0 - \beta_1 - \beta_2)\frac{u(x)}{h} \\
& + (1 + 2\beta_{-2} + \beta_{-1} - \beta_1 - 2\beta_2 + \alpha_{-1} + \alpha_1)u'(x) \\
& + (-4\beta_{-2} - \beta_{-1} - \beta_1 - 4\beta_2 - 2\alpha_{-1} + 2\alpha_1)\frac{h}{2}u''(x) \\
& + (8\beta_{-2} + \beta_{-1} - \beta_1 - 8\beta_2 + 3\alpha_{-1} + 3\alpha_1)\frac{h^2}{6}u^3(x) \\
& + (-16\beta_{-2} - \beta_{-1} - \beta_1 - 16\beta_2 - 4\alpha_{-1} + 4\alpha_1)\frac{h^3}{24}u^{(4)}(x) \\
& + (32\beta_{-2} + \beta_{-1} - \beta_1 - 32\beta_2 + 5\alpha_{-1} + 5\alpha_1)\frac{h^4}{120}u^{(5)}(x) \\
& + (-64\beta_{-2} - \beta_{-1} - \beta_1 - 64\beta_2 + 6\alpha_{-1} + 6\alpha_1)\frac{h^5}{720}u^{(6)}(x) \\
& + (128\beta_{-2} + \beta_{-1} - \beta_1 - 128\beta_2 + 7\alpha_{-1} + 7\alpha_1)\frac{h^6}{5040}u^7(x) \\
& + \ldots
\end{aligned} \tag{1.15}$$

The unknown constants $\alpha_{-1}, \alpha_1, \beta_{-2}, \beta_{-1}, \beta_0, \beta_1$ and β_2 are obtained by equating the Taylor series coefficients of various orders of h. As a result, we obtain a system of seven linear algebraic equations as shown below

$$
\begin{aligned}
-\beta_{-2}-\beta_{-1}-\beta_0-\beta_1-\beta_2 &= 0,\\
1+2\beta_{-2}+\beta_{-1}-\beta_1-2\beta_2+\alpha_{-1}+\alpha_1 &= 0,\\
-4\beta_{-2}-\beta_{-1}-\beta_1-4\beta_2-2\alpha_{-1}+2\alpha_1 &= 0,\\
8\beta_{-2}+\beta_{-1}-\beta_1-8\beta_2+3\alpha_{-1}+3\alpha_1 &= 0,\\
-16\beta_{-2}-\beta_{-1}-\beta_1-16\beta_2-4\alpha_{-1}+4\alpha_1 &= 0,\\
32\beta_{-2}+\beta_{-1}-\beta_1-32\beta_2+5\alpha_{-1}+5\alpha_1 &= 0,\\
-64\beta_{-2}-\beta_{-1}-\beta_1-64\beta_2+6\alpha_{-1}+6\alpha_1 &= 0.
\end{aligned}
$$

Solving the above equations gives:

$$\alpha_{-1}=\frac{1}{3},\alpha_1=\frac{1}{3},\beta_{-2}=-\frac{1}{36},\beta_{-1}=-\frac{7}{9},\beta_0=0,\beta_1=\frac{7}{9},\beta_2=\frac{1}{36} \tag{1.16}$$

Therefore, the sixth-order CFDS approximation for first order derivatives is given by

$$\frac{1}{3}u'_{i-1}+u'_i+\frac{1}{3}u'_{i+1}=\frac{7}{9h}(u_{i+1}-u_{i-1})+\frac{1}{36h}(u_{i+2}-u_{i-2}) \tag{1.17}$$

with a local truncation error

$$\tau_i=-\frac{4h^6}{5040}u^{(7)}_{i+\eta},\quad -2<\eta<2 \tag{1.18}$$

For illustrative purpose, we describe the application of the CFDS to the following First order differential equations:

$$u'=f(x) \tag{1.19}$$

with known boundary conditions at $u(a)$ and $u(b)$, where $f(x)$ is a continuous function. Since we know boundary conditions at $i=1$ and $i=N$, the CFDS must be adjusted for the nodes near the boundary points (one-sided schemes).

When $i=2$

$$u'_2+\frac{1}{3}u'_3=\frac{1}{h}(a_1u_1+a_2u_2+a_3u_3+a_4u_4+a_5u_5+a_6u_6+a_7u_7) \tag{1.20}$$

We expand (1.20) using the Taylor series at the $i-th$ collocation point with respect to the step size h to get

$$
\begin{aligned}
0 = {} & (-a_1 - a_2 - a_3 - a_4 - a_5 - a_6 - a_7)\frac{u(x)}{h} \\
& + (4 - 3a_2 - 6a_3 - 9a_4 - 12a_5 - 15a_6 - 18a_7)\frac{u'(x)}{3} \\
& + (10 - 3a_2 - 12a_3 - 27a_4 - 48a_5 - 75a_6 - 108a_7)\frac{h}{6}u''(x) \\
& + (7 - a_2 - 8a_3 - 27a_4 - 64a_5 - 125a_6 - 216a_7)\frac{h^2}{6}u'''(x) \\
& + (44 - 3a_2 - 48a_3 - 243a_4 - 768a_5 - 1875a_6 - 3888a_7)\frac{h^3}{72}u^{(4)(x)} \\
& + (95 - 3a_2 - 96a_3 - 729a_4 - 3072a_5 - 9375a_6 - 23328a_7)\frac{h^4}{360}u^{(5)}(x) \\
& + (70 - a_2 - 64a_3 - 729a_4 - 4096a_5 - 16525a_6 - 46656a_7)\frac{h^5}{720}u^{(6)}(x)
\end{aligned} \tag{1.21}
$$

The unknown constants $a_1, a_2, a_3, a_4, a_5, a_6$ and a_7 are obtained by equating the Taylor series coefficients of various orders of h. As a result, we obtain a system of seven linear algebraic equations as shown below

$$
\begin{aligned}
-a_1 - a_2 - a_3 - a_4 - a_5 - a_6 - a_7 &= 0, \\
\frac{1}{3}(4 - 3a_2 - 6a_3 - 9a_4 - 12a_5 - 15a_6 - 18a_7) &= 0, \\
\frac{1}{6}(10 - 3a_2 - 12a_3 - 27a_4 - 48a_5 - 75a_6 - 108a_7) &= 0, \\
\frac{1}{6}(7 - a_2 - 8a_3 - 27a_4 - 64a_5 - 125a_6 - 216a_7) &= 0, \\
\frac{1}{72}(44 - 3a_2 - 48a_3 - 243a_4 - 768a_5 - 1875a_6 - 3888a_7) &= 0, \\
\frac{1}{360}(95 - 3a_2 - 96a_3 - 729a_4 - 3072a_5 - 9375a_6 - 23328a_7) &= 0, \\
\frac{1}{720}(70 - a_2 - 64a_3 - 729a_4 - 4096a_5 - 16525a_6 - 46656a_7) &= 0.
\end{aligned} \tag{1.22}
$$

Solving the above equations gives:

$$
a_1 = -\frac{7}{45}, a_2 = -\frac{17}{12}, a_3 = \frac{83}{36}, a_4 = -\frac{11}{9}, a_5 = \frac{2}{3}, a_6 = -\frac{37}{180}, a_7 = \frac{1}{36} \tag{1.23}
$$

Therefore, the sixth-order CFDS approximation for first order derivatives at $i = 2$ is given by

$$
u'_2 + \frac{1}{3}u'_3 = \frac{1}{h}\left(-\frac{7}{45}u_1 - \frac{17}{12}u_2 + \frac{83}{36}u_3 - \frac{11}{9}u_4 + \frac{2}{3}u_5 - \frac{37}{180}u_6 + \frac{1}{36}u_7\right) \tag{1.24}
$$

and when $i = N-1$

$$\frac{1}{3}u'_{N-2} + u'_{N-1} = \frac{1}{h}(b_1u_N + b_2x_{N-1} + b_3u_{N-2} + b_4u_{N-3} + b_5u_{N-4} + b_6u_{N-5} + b_7u_{N-6}) \tag{1.25}$$

We expand (1.25) using the Taylor series at the i^{th} collocation point with respect to the step size h to get

$$\begin{aligned}
0 = {} & (-b_1 - b_2 - b_3 - b_4 - b_5 - b_6 - b_7)\frac{u(x)}{h} \\
& + (4 + 3b_2 + 6b_3 + 9b_4 + 12b_5 + 15b_6 + 18b_7)\frac{u'(x)}{3} \\
& + (-10 - 3b_2 - 12b_3 - 27b_4 - 48b_5 - 75b_6 - 108b_7)\frac{h}{6}u''(x) \\
& + (7 + b_2 + 8b_3 + 27b_4 + 64b_5 + 125b_6 + 216b_7)\frac{h^2}{6}u'''(x) \\
& + (-44 - 3b_2 - 48b_3 - 243b_4 - 768b_5 - 1875b_6 - 3888b_7)\frac{h^3}{72}u^{(4)(x)} \\
& + (95 + 3b_2 - 96b_3 + 729b_4 + 3072b_5 + 9375b_6 + 23328b_7)\frac{h^4}{360}u^{(5)}(x) \\
& + (-70 - b_2 - 64b_3 - 729b_4 - 4096b_5 - 16525b_6 - 46656b_7)\frac{h^5}{720}u^{(6)}(x)
\end{aligned} \tag{1.26}$$

The unknown constants $a_1, a_2, a_3, a_4, a_5, a_6$ and a_7 are obtained by equating the Taylor series coefficients of various orders of h. As a result, we obtain a system of seven linear algebraic equations as shown below

$$\begin{aligned}
-b_1 - b_2 - b_3 - b_4 - b_5 - b_6 - b_7 &= 0, \\
\frac{1}{3}(4 + 3b_2 + 6b_3 + 9b_4 + 12b_5 + 15b_6 + 18b_7) &= 0, \\
\frac{1}{6}(-10 - 3b_2 - 12b_3 - 27b_4 - 48b_5 - 75b_6 - 108b_7) &= 0, \\
\frac{1}{6}(7 + b_2 + 8b_3 + 27b_4 + 64b_5 + 125b_6 + 216b_7) &= 0, \\
\frac{1}{72}(-44 - 3b_2 - 48b_3 - 243b_4 - 768b_5 - 1875b_6 - 3888b_7) &= 0, \\
\frac{1}{360}(95 + 3b_2 - 96b_3 + 729b_4 + 3072b_5 + 9375b_6 + 23328b_7) &= 0, \\
\frac{1}{720}(-70 - b_2 - 64b_3 - 729b_4 - 4096b_5 - 16525b_6 - 46656b_7) &= 0.
\end{aligned} \tag{1.27}$$

Solving the above equations gives:

$$b_1 = \frac{7}{45}, b_2 = \frac{17}{12}, b_3 = -\frac{83}{36}, b_4 = \frac{11}{9}, b_5 = -\frac{2}{3}, b_6 = \frac{37}{180}, b_7 = -\frac{1}{36} \tag{1.28}$$

Therefore, the sixth-order CFDS approximation for first order derivatives at $i = N-1$ is given by

$$\begin{aligned}\frac{1}{3}u'_{N-2} + u'_{N-1} = \frac{1}{h}\Bigg(&\frac{7}{45}u_N + \frac{17}{12}u_{N-1} - \frac{83}{36}u_{N-2} + \frac{11}{9}u_{N-3} - \frac{2}{3}u_{N-4} \\ &+ \frac{37}{180}u_{N-5} - \frac{1}{36}u_{N-6}\Bigg)\end{aligned} \tag{1.29}$$

The approximations of the first derivatives at the end points are given by the above expressions.

Combining Eq.(1.17), Eq.(1.24) and Eq.(1.29) into matrix form, the equations for approximating the first order derivatives can be expressed as

$$A_1 U' = B_1 U + K_1 \tag{1.30}$$

where

$$A_1 = \begin{bmatrix} 1 & \frac{1}{3} & & & & \\ \frac{1}{3} & 1 & \frac{1}{3} & & & \\ & \frac{1}{3} & 1 & \frac{1}{3} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{3} & 1 & \frac{1}{3} \\ & & & & \frac{1}{3} & 1 \end{bmatrix}_{(N-2)\times(N-2)} \qquad K_1 = \frac{1}{h}\begin{bmatrix} -\frac{7}{45}u_1 \\ -\frac{u_1}{36} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{u_N}{36} \\ \frac{7}{45}u_N \end{bmatrix}_{(N-2)\times 1}$$

$$B_1 = \frac{1}{h}\begin{bmatrix} -\frac{17}{12} & \frac{83}{36} & -\frac{11}{9} & \frac{2}{3} & -\frac{37}{180} & \frac{1}{36} & \\ -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} & & & \\ -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ & & & -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} \\ & -\frac{1}{36} & \frac{37}{180} & -\frac{2}{3} & \frac{11}{9} & -\frac{83}{36} & \frac{17}{12} \end{bmatrix}_{(N-2)\times(N-2)}$$

and

$$U = [u_1, u_2, u_3, \ldots, u_N] \tag{1.31}$$

The equation for approximating the first derivatives is given by

$$U' = E_1 U + H_1 \tag{1.32}$$

where

$$E_1 = A_1^{-1} B_1, \qquad H_1 = A_1^{-1} K_1$$

1.3.2 Second-order derivative

The second-order derivative at $i-th$ collocation point can be given as follows

$$\alpha_1 u''_{i-1} + u''_i + \alpha_2 u''_{i+1} = \frac{\beta_{-2} u_{i-2} + \beta_{-1} u_{i-1} + \beta_0 u_i + \beta_1 u_{i+1} + \beta_2 u_{i+2}}{h^2} \tag{1.33}$$

We expand (1.33) using the Taylor series at the i^{th} collocation point with respect to the step size h to get

$$\begin{aligned}
0 = {} & (-\beta_{-2} - \beta_{-1} - \beta_0 - \beta_1 - \beta_2)\frac{u(x)}{h^2} \\
& + (2\beta_{-2} + \beta_{-1} - \beta_1 - 2\beta_2)\frac{u'(x)}{h} \\
& + (2 - 4\beta_{-2} - \beta_{-1} - \beta_1 - 4\beta_2 + 2\alpha_{-1} + 2\alpha_1)\frac{1}{2}u''(x) \\
& + (8\beta_{-2} + \beta_{-1} - \beta_1 - 8\beta_2 - 6\alpha_{-1} + 6\alpha_1)\frac{h}{6}u^3(x) \\
& + (-16\beta_{-2} - \beta_{-1} - \beta_1 - 16\beta_2 + 12\alpha_{-1} + 12\alpha_1)\frac{h^2}{24}u^{(4)}(x) \\
& + (32\beta_{-2} + \beta_{-1} - \beta_1 - 32\beta_2 - 20\alpha_{-1} + 20\alpha_1)\frac{h^3}{120}u^{(5)}(x) \\
& + (-64\beta_{-2} - \beta_{-1} - \beta_1 - 64\beta_2 + 30\alpha_{-1} + 30\alpha_1)\frac{h^4}{720}u^{(6)}(x) \\
& + (128\beta_{-2} + \beta_{-1} - \beta_1 - 128\beta_2 - 42\alpha_{-1} + 42\alpha_1)\frac{h^5}{5040}u^7(x) \\
& + (-256\beta_{-2} - \beta_{-1} - \beta_1 - 256\beta_2 + 56\alpha_{-1} + 56\alpha_1)\frac{h^6}{40320}u^{(8)}(x) \\
& + \ldots
\end{aligned} \tag{1.34}$$

The unknown constants $\alpha_{-1}, \alpha_1, \beta_{-2}, \beta_{-1}, \beta_0, \beta_1$ and β_2 are obtained by equating the Taylor series coefficients of various orders of h. As a result, we obtain a system of seven linear algebraic equations as shown below

$$\begin{aligned}
-\beta_{-2} - \beta_{-1} - \beta_0 - \beta_1 - \beta_2 &= 0,\\
2\beta_{-2} + \beta_{-1} - \beta_1 - 2\beta_2 &= 0,\\
2 - 4\beta_{-2} - \beta_{-1} - \beta_1 - 4\beta_2 + 2\alpha_{-1} + 2\alpha_1 &= 0,\\
8\beta_{-2} + \beta_{-1} - \beta_1 - 8\beta_2 - 6\alpha_{-1} + 6\alpha_1 &= 0,\\
-16\beta_{-2} - \beta_{-1} - \beta_1 - 16\beta_2 + 12\alpha_{-1} + 12\alpha_1 &= 0,\\
32\beta_{-2} + \beta_{-1} - \beta_1 - 32\beta_2 - 20\alpha_{-1} + 20\alpha_1 &= 0,\\
-64\beta_{-2} - \beta_{-1} - \beta_1 - 64\beta_2 + 30\alpha_{-1} + 30\alpha_1 &= 0,\\
128\beta_{-2} + \beta_{-1} - \beta_1 - 128\beta_2 - 42\alpha_{-1} + 42\alpha_1 &= 0,
\end{aligned}$$

Solving the above equations gives

$$\alpha_{-1} = \frac{2}{11}, \alpha_1 = \frac{2}{11}, \beta_{-2} = -\frac{3}{44}, \beta_{-1} = \frac{12}{11}, \beta_0 = -\frac{51}{22}, \beta_1 = \frac{12}{11}, \beta_2 = \frac{3}{44} \tag{1.35}$$

Therefore, the sixth-order CFDS approximation for second-order derivatives at interior points is given by

$$\frac{2}{11}u''_{i-1} + u''_i + \frac{2}{11}u''_{i+1} = \frac{12}{11h^2}(u_{i+1} - 2u_i + u_{i-1}) + \frac{3}{44h^2}(u_{i+2} - 2u_i + u_{i-2}) \tag{1.36}$$

with a local truncation error

$$\tau_i = -\frac{16.7h^6}{40320}u^{(7)}_{i+\eta}, \quad -2 < \eta < 2 \tag{1.37}$$

For illustrative purpose, we consider a single boundary value problem of the form

$$u'' + u' = f(x), \;\; a < x < b, \tag{1.38}$$

with Dirichlet boundary condition

$$u(a) = \alpha_a, \;\; u(b) = \alpha_b \tag{1.39}$$

where $f(x)$ is a continuous function, α_a and α_b are known constants. Note that the schemes at the boundaries are adjusted in order to maintain the $O(h^6)$ accuracy. This discussion will only be limited to the derivation of CFDS for Dirichlet boundary conditions at both ends.

The procedure of deriving the schemes is similar to the previous section.

When $i = 2$

$$u_2'' + \frac{11}{2}u_3'' = \frac{1}{h^2}\left(\frac{31}{45}u_1 - \frac{19}{110}u_2 - \frac{339}{110}u_3 + \frac{1933}{396}u_4 - \frac{40}{11}u_5 + \frac{96}{55}u_6 - \frac{479}{990}u_7 + \frac{13}{220}u_8\right) \tag{1.40}$$

and when $i = N-1$

$$\frac{2}{11}u_{N-2}'' + u_{N-1}'' = \frac{1}{h^2}\left(\frac{31}{45}u_N - \frac{19}{110}u_{N-1} - \frac{339}{110}u_{N-2} + \frac{1933}{396}u_{N-3} - \frac{40}{11}u_{N-4} + \frac{96}{55}u_{N-5} - \frac{479}{990}u_{N-6} + \frac{13}{220}u_{N-7}\right) \tag{1.41}$$

Combining Eq.(1.36),Eq.(1.40) and Eq.(1.41) into matrix form, the equations for approximating the second order derivatives can be expressed as

$$A_2 U'' = B_2 U + K_2 \tag{1.42}$$

where

$$A_2 = \begin{bmatrix} 1 & \frac{2}{11} & & & & \\ \frac{2}{11} & 1 & \frac{2}{11} & & & \\ & \frac{2}{11} & 1 & \frac{2}{11} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{2}{11} & 1 & \frac{2}{11} \\ & & & & \frac{2}{11} & 1 \end{bmatrix}_{(N-2)\times(N-2)} \qquad K_2 = \frac{1}{h^2}\begin{bmatrix} \frac{31u_1}{45} \\ \frac{3u_1}{44} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{3u_N}{44} \\ \frac{31u_N}{45} \end{bmatrix}_{(N-2)\times 1}$$

$$B_2 = \frac{1}{h^2}\begin{bmatrix} -\frac{19}{110} & -\frac{339}{110} & \frac{1933}{396} & -\frac{40}{11} & \frac{96}{55} & -\frac{479}{990} & \frac{13}{220} & \\ \frac{12}{11} & -\frac{51}{22} & \frac{12}{11} & \frac{3}{44} & & & & \\ \frac{3}{44} & \frac{12}{11} & -\frac{51}{22} & \frac{12}{11} & \frac{3}{44} & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \frac{3}{44} & \frac{12}{11} & -\frac{51}{22} & \frac{12}{11} & \frac{3}{44} \\ & & & & \frac{3}{44} & \frac{12}{11} & -\frac{51}{22} & \frac{12}{11} \\ & \frac{13}{220} & -\frac{479}{990} & \frac{96}{55} & -\frac{40}{11} & \frac{1933}{396} & -\frac{339}{110} & -\frac{19}{110} \end{bmatrix}_{(N-2)\times(N-2)}$$

and

$$U = [u_1, u_2, u_3, \ldots, u_N] \tag{1.43}$$

The equation for approximating the second derivatives is given by

$$U'' = E_2 U + H_2 \tag{1.44}$$

where

$$E_2 = A_2^{-1} B_2, \qquad H_2 = A_2^{-1} K_2$$

1.4 Spectral method

Spectral methods are powerful tools for solving differential equations if the physical domain is simple and the solution is smooth. They are significantly more accurate than related numerical methods such as finite differences and finite elements. Spectral methods approximate functions by means of truncated series of orthogonal functions (polynomials). Example of such orthogonal functions are the Fourier series which is used for periodic problems, Chebyshev, Legendre polynomials, etc. The speed of convergence is one of the great advantages of spectral methods. The methods became famous in the 1970s [77, 12].

The Chebyshev spectral method approximate functions by means of truncated series of Chebyshev orthogonal polynomials. The Chebyshev polynomials $T_N(x)$ of order N are defined as

$$T_N(x) = \cos(N \cos^{-1}(x)), \qquad N \in \mathbb{N}. \tag{1.45}$$

The Chebyshev interpolation $u_N(x)$ of a function $u(x)$ at $x = x_i$ is defined by

$$u_N(x) = \sum_{i=0}^{N} u(x_i) L_i(x), \tag{1.46}$$

where the collocation points x_i are chosen to be the extrema of $T_N(x)$:

$$\{x_i\} = \left\{ \cos\left(\frac{\pi i}{N}\right) \right\}_{i=0}^{N}, \qquad x \in [-1,1]$$

which are the Chebyshev-Gauss-Lobatto points. This choice is made from the simple reason that in Lagrangian interpolation, if the interpolation points are taken to be the zeros of the polynomial, the error is minimized.

$L_i(x)$, i = 0,1,...,N are Lagrange polynomials of order N based on the Chebyshev-Gauss-Lobatto points defined as

$$L_i(x) = \frac{(-1)^{i+1}(1-x^2)T_N'(x)}{\bar{c}_i N^2 (x - x_i)}, \qquad i = 0,1,\ldots,N \tag{1.47}$$

where $\bar{c}_0 = \bar{c}_N = 2, \bar{c}_i = 1$ for $i = 1,2,\ldots,N-1$.

The derivatives of the approximate solution at the collocation points are computed as

$$\frac{d^n}{dx^n} u(x_i) = \sum_{j=0}^{N} u(x_i) L^{(n)}(x_j) = \sum_{j=0}^{N} D_{ij}^{(n)} u(x_i), \tag{1.48}$$

where $D_{ij}^{(n)} = L^{(n)}(x_i)$ is an $(N+1) \times (N+1)$ Chebyshev differentiation matrix for $i, j = 0,1,\ldots,N$.

The first order Chebyshev derivative matrix at the collocation points is given by [12, 77]

$$D_{ij} = D_{ij}^{(1)} = \begin{cases} \dfrac{c_i(-1)^{j+i}}{c_j(x_j - x_i)}, & i \neq j \\ -\dfrac{x_i}{2(1-x_i^2)}, & (i=j) \neq 0,N \\ \dfrac{2N^2+1}{6}, & i,j = 0 \\ -\dfrac{2N^2+1}{6}, & i,j = N \end{cases} \tag{1.49}$$

Extending the spectral collocation idea to partial differential equations (Bivariate Chebyshev spectral method), the Chebyshev polynomials $T_N(\chi)$ of order N are defined as

$$T_N(\chi) = \cos(N\cos^{-1}(\chi)), \qquad N \in \mathbb{N}. \tag{1.50}$$

The Chebyshev interpolation $u_N(\chi,\tau)$ of a function $u(\chi,\tau)$ at $\chi = \chi_i$ and $\tau = \tau_j$ is defined by

$$u_N(\chi,\tau) = \sum_{j=0}^{M}\sum_{i=0}^{N} u_{ji} L_j(\tau) L_i(\chi) \tag{1.51}$$

where the collocation points χ_i and τ_j are chosen to be the extrema of T_N:

$$\{\chi_i\} = \left\{\cos\left(\frac{\pi i}{N}\right)\right\}_{i=0}^{N}, \qquad \{\tau_j\} = \left\{\cos\left(\frac{\pi j}{M}\right)\right\}_{j=0}^{M}. \tag{1.52}$$

which are the Chebyshev-Gauss-Lobatto points. $L_i(\chi)$, $i = 0,1,\ldots,N$, are Lagrange polynomials of order N based on the Chebyshev-Gauss-Lobatto points defined as

$$L_i(\chi) = \frac{(-1)^{i+1}(1-\chi^2)T_N'(\chi)}{\bar{c}_i N^2(\chi-\chi_i)}, \qquad i = 0,1,\ldots,N \tag{1.53}$$

where $\bar{c}_0 = \bar{c}_N = 2, \bar{c}_i = 1$ for $i = 1,2,\ldots,N-1$. The polynomials $L_j(\tau)$ are defined in a similar manner.

The nth order space and time derivatives of the approximate solution at the collocation points are computed as

$$\frac{\partial^n}{\partial\chi^n}u(\chi_i,\tau_j) = \sum_{q=0}^{M}\sum_{p=0}^{N} u(\chi_p,\tau_q)L_q(\tau_j)\frac{d^n L_p(\chi_i)}{d\chi^n} = \sum_{p=0}^{N} D_{ip}^n u(\chi_p,\tau_j) = \mathbf{D}^n\mathbf{U}_j, \tag{1.54}$$

$$\frac{\partial^n}{\partial\tau^n}u(\chi_i,\tau_j) = \sum_{q=0}^{M}\sum_{p=0}^{N} u(\chi_p,\tau_q)\frac{d^n L_q(\tau_j)}{d\tau^n}L_p(\chi_i) = \sum_{q=0}^{M} d_{jq}^n u(\chi_i,\tau_q), \tag{1.55}$$

where $D_{ip} = \frac{dL_p(\chi_i)}{d\chi}$ is an $(N+1)\times(N+1)$ Chebyshev differentiation matrix for $i,p = 0,1,\ldots,N$, and $d_{jq} = \frac{dL_q(\tau_j)}{d\tau}$ is an $(M+1)\times(M+1)$ Chebyshev differentiation matrix for $j,q = 0,1,\ldots,M$,

The first order Chebyshev derivative matrix at the collocation points is given by [12, 77]

$$D_{ip} = D_{ip}^{(1)} = \begin{cases} \dfrac{c_i(-1)^{p+i}}{c_p(\chi_p - \chi_i)}, & i \neq p \\ -\dfrac{\chi_i}{2(1-\chi_i^2)}, & (i=p) \neq 0, N \\ \dfrac{2N^2+1}{6}, & i,p=0 \\ -\dfrac{2N^2+1}{6}, & i,p=N \end{cases} \tag{1.56}$$

Similarly, we can compute the first order Chebyshev differentiation matrix d_{jq}.

1.5 Multi-domain approach

In this section, we describe the development of a multi-domain approach. The multi-domain approaches allow better conditioned matrices and larger step sizes than single domain computations. Also, an important effect of the multiple domain approaches is that they enhance the accuracy of the approximation significantly. In such methods, the given domain is divided into two or more sub-domains and the problem solved in each sub-domain with appropriate interface conditions connecting the solution across the sub-domain boundaries.

We first decompose the interval of integration $\Omega = [0,T]$ into non-overlapping intervals $\Omega_n = [t_{n-1}, t_n]$ where $n = 1,2,3,\ldots,P$, where $t_0 = 0$ and $t_F = T$. The main idea of the multi-domain approach is that of determining the solution of equation differential equation independently on each sub-interval, one at a time, beginning at the initial condition. The given initial condition is considered to be the left boundary of the time interval and used to compute the solution in the first sub-interval. The computed solution at the right hand boundary of the first interval is used as an initial condition in the subsequent second sub-interval to computed solution in the second sub-interval. The same process is repeated until the last sub-interval. The process where the solutions in different intervals are matched along their common boundary is called patching. The practical application of the multi-domain approach is illustrated by solving the Chaotic and hyperchaotic systems, nonlinear evolution partial differential equations and the generalized Kuramoto-Sivashinsky equation using higher order CFDS or spectral collocation with chebyshev functions.

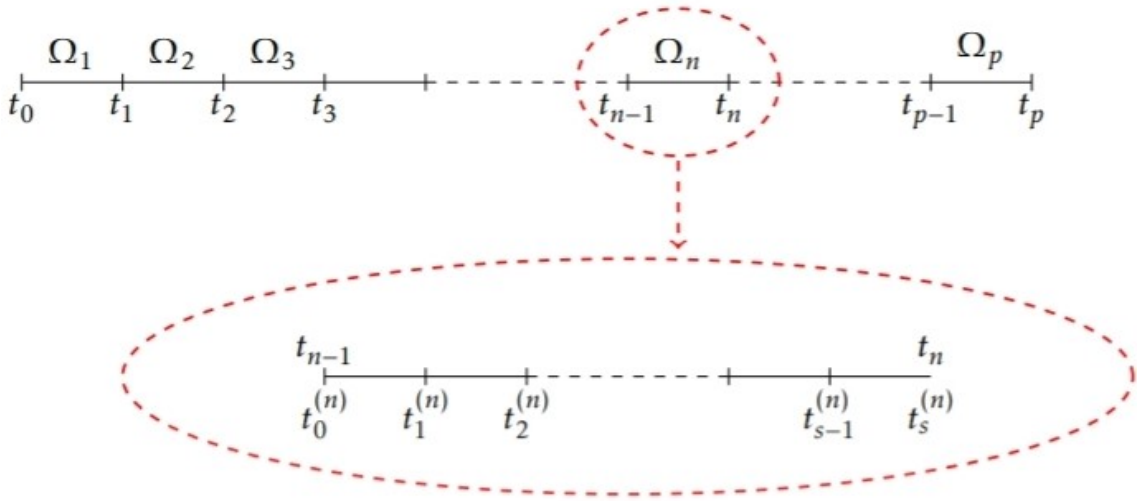


Fig. 1.1 The multi-domain grid

1.6 Dissertation outline

This dissertation consists of five chapters:

Chapter 2 In this chapter, we present a new application of higher order compact finite differences to solve nonlinear initial value problems exhibiting chaotic behavior. The method involves dividing the domain of the problem into multiple sub domains, with each sub domain integrated using higher order compact finite difference schemes. The nonlinearity is dealt with by using a Gauss-Seidel like relaxation. The method is therefore referred to as the multi-domain compact finite difference relaxation method (MD-CFDRM). In this new application, the MD-CFDRM is used to solve famous chaotic systems and hyperchaotic systems.

Chapter 3 We investigate a new application of higher order compact finite differences to solve nonlinear evolution partial differential equations. The method involves dividing the domain of the problem into multiple sub-domains, with each sub-domain integrated using higher order compact finite difference schemes. The nonlinearity of the evolution partial differential equations is dealt with by using the quasi-linearization technique. The method is therefore referred to as the multi-domain compact finite difference quasi-linearization method (MD-CFDQLM). In this work, the method is used to solve nonlinear evolution partial differential equations, namely, the Fisher's equation, Burgers-Fisher equation, Burgers-Huxley equation and the coupled Burgers' equations over a large time domain.

Chapter 4 We solve the Kuramoto-Sivashinsky (KS) equation which is a well known partial differential equation exhibiting chaos. Chaotic systems are characterized by rapidly and sharply changing solutions. As a result, computing their solutions is a bit tricky especially for large domains. In this chapter, we use the multi-domain bivariate spectral quasi-linearization method to solve different examples of the KS equations. The method uses the idea of domain decomposition to handle the continuously changing solutions.

Chapter 5 Conclusion of the work in this dissertation.

Chapter 2

Compact finite difference relaxation method for chaotic and hyperchaotic initial value systems

2.1 Introduction

Chaotic behavior can be observed in a variety of systems such as electrical circuits, lasers, fluid dynamics, mechanical devices, time evolution of the magnetic field of celestial bodies, population growth, and many other areas of scientific application. This kind of behavior was first observed by Lorenz [55] in 1963 on a system of ordinary differential equations modelling weather phenomena. Later on Rössler [70] observed hyperchaotic behavior on the ordinary differential equations for modeling chemical reactions. The difference between the two is that, chaotic systems consist of only one positive Lyapunov exponent whereas hyperchaotic systems have at least two positive Lyapunov exponents. Hyperchaotic systems generally have more complex dynamical behaviours than the ordinary chaotic systems.

Chaotic systems are characterized by high sensitivity to small perturbation on initial data and rapidly changing solutions. Such rapid variations in the solution pose tremendous problems to a number of numerical approximations. This often requires using a large number of grid points. However, that leads to large memory requirements. Also, for methods like the pseudo-spectral method the approximations can exhibit spurious oscillations which can lead to nonlinear instabilities if the number of grid points is very big. A number of researchers have tried to circumvent this problem by considering domain decomposition techniques, in which the domain of the problem is divided into two or more sub-domains. Multi-domain approaches allow better conditioned matrices and larger step sizes than single domain

computations. Also, an important effect of the multiple domain approaches is that they enhance the accuracy of the approximation significantly. In such methods, the given domain is divided into two or more sub-domains and the problem solved in each sub-domain with appropriate interface conditions connecting the solution across the sub-domain boundaries.

There is quite a substantial number of articles in the literature where the idea of domain decomposition has been applied specifically to chaotic and hyperchaotic systems. Examples of semi-analytical multi-domain methods applied to chaotic and hyperchaotic systems include the multi-stage Adomian decomposition method [3, 37, 5], multistage homotopy analysis method [6, 1], multi-stage differential transformation method [28, 32, 63], multi-stage variational iteration method [9, 33, 34], multistage homotopy perturbation methods [16, 17, 15, 79]. The analytical nature of these methods and their application in many subintervals makes the whole computation process tedious and time-consuming.

Recently, Motsa et al [61, 62] implemented the multi-domain approach on the Chebyshev spectral collocation method which is completely numerical. They used this to solve a few examples of chaotic and hyperchaotic systems. Their approach was very efficient and easy to implement and gave remarkable results. Their work has motivated the current project. In this work, instead of using the Chebyshev spectral method, we use higher order compact finite difference schemes (CFDS) to discretize in each subinterval. The advantage of the higher order CFDS over the standard second order finite difference, is that they give high accuracy on coarser grids with greater computational efficiency [44]. When compared to spectral methods, compact schemes are more flexible in terms of application to complex geometries and boundary conditions. CFDS have largely been applied to solve partial differential equations e.g. Burger's equation [73, 84], Navier-Stokes equation [75] , Korteweg-de Vries equation [52], Black-Scholes equation [30], and many more [8, 72, 74]. Before applying the CFDS, we deal with the nonlinearity of the chaotic and hyperchaotic systems by using a Gauss-Seidel relaxation approach. So, we refer to the method as the multi-domain compact finite difference relaxation method (MD-CFDRM). We examine it's applicability on 3 chaotic systems and 2 hyperchaotic chaotic systems. The numerical results are compared with Motsa et al's multi-domain spectral relaxation method (MSRM) [62, 61].

2.2 Multi-domain compact finite difference relaxation method - MD-CFDRM

In this section, we give a brief description of the numerical scheme. We employ the multi-domain compact finite relaxation method (MD-CFDRM) for the solution of common chaotic

and hyperchaotic systems governed by nonlinear systems of first order IVPs. The MD-CFDRM algorithm is based on Gauss-Seidel type of relaxation that decouples and linearises the system and the use of compact finite difference method to solve the linearised equations in a sequential manner. We consider chaotic and hyperchaotic systems of the form

$$\dot{x}_r(t) = \sum_{k=1}^{m} \alpha_{r,k} x_k(t) + f_r[x_1(t), x_2(t), \ldots, x_m(t)], \tag{2.1}$$

subject to the initial conditions

$$x_r(0) = x_r^*, \;\; r = 1, 2, \ldots, m \tag{2.2}$$

The scheme compute the solution of equation (2.1) in a sequence of equal subintervals that makes the entire interval.We first decompose the interval of integration $\Omega = [0, T]$ into non-overlapping intervals $\Omega_i = [t_{i-1}, t_i]$ where $i = 1, 2, 3, \ldots, F$, where $t_0 = 0$ and $t_F = T$. Equation (2.2) is used as the initial condition for obtaining the solution in the first sub-interval $[t_0, t_1]$.The solution of equation (2.1) is computed in the first interval $[t_0, t_1]$ and is labeled as $x_r^1(t)$, and $x_r^i(t)$ be the solutions in the subsequent sub-intervals Ω_i $(i = 2, \ldots, F)$. The value of the solution at last node of the first interval $\Omega_1 = (t_0, t_1)$, given by $x_r^1(t_1)$ is used as an initial condition in the second sub-interval. Then we use the continuity condition between neighbouring sub-intervals to obtain the initial conditions for solving (2.1) in the rest of the Ω_i sub-intervals. Thus, in each interval $[t_{i-1}, t_i]$ we must solve

$$\dot{x}_r^i = g_r + \alpha_{r,r} x_r^i + (1 - \delta_{r,k}) \sum_{k=1}^{m} \alpha_{r,k} x_k^i + f_r[x_1^i, x_2^i, \ldots, x_m^i] \tag{2.3}$$

subject to

$$x_r^i(t_{i-1}) = x_r^{i-1}(t_{i-1}) \tag{2.4}$$

where δ_{rk} is the Kronecker delta. As mentioned earlier, the main idea behind the MD-CFDRM scheme is decoupling the system of nonlinear IVPs using the Gauss-Seidel idea of decoupling systems of algebraic equations. The proposed MD-CFDRM iteration scheme for

the solution in the interval $\Omega_i = [t_{i-1}, t_i]$ is given as

$$\dot{x}^i_{1,s+1} - \alpha_{1,1}x^i_{1,s+1} = g_1 + \sum_{k=2}^{m} \alpha_{1,k}x^i_{k,s} + f_1[x^i_{1,s}, \ldots, x^i_{n,s}] \tag{2.5}$$

$$\dot{x}^i_{2,s+1} - \alpha_{2,2}x^i_{2,s+1} = g_2 + \sum_{\substack{k=1 \\ k\neq 2}}^{m} \alpha_{2,k}x^i_{k,s} + f_2[x^i_{1,s+1}, \ldots, x^i_{n,s}] \tag{2.6}$$

$$\vdots$$

$$\dot{x}^i_{m,s+1} - \alpha_{m,m}x^i_{m,s+1} = g_m + \sum_{k=1}^{m-1} \alpha_{m,k}x^i_{k,s} + f_m[x^i_{1,s+1}, \ldots, x^i_{m-1,s+1}, x^i_{m,s}] \tag{2.7}$$

subject to the initial conditions

$$x^i_{r,s+1}(t_{i-1}) = x^{i-1}_r(t_{i-1}), \;\; r = 1,2,3,\ldots,m \tag{2.8}$$

$$x^i_{r,0}(t) = \begin{cases} x^*_r & if \;\; i = 1, \\ x^{i-1}_r(t_{i-1}) & if \;\; 2 \leq i \leq F \end{cases} \tag{2.9}$$

We use the sixth order compact finite difference schemes to solve (2.5 - 2.7) on each interval. The sixth order CFDS for approximating the first derivative is given by

$$\frac{1}{3}\dot{x}_{i-1} + \dot{x}_i + \frac{1}{3}\dot{x}_{i+1} = \frac{14}{9}\frac{x_{i+1} - x_{i-1}}{2h} + \frac{1}{9}\frac{x_{i+2} - x_{i-2}}{4h}, \tag{2.10}$$

The approximations of the first derivative at the end points are given by the following:

$$\begin{aligned} \dot{x}_2 + \frac{1}{3}\dot{x}_3 =& \frac{1}{h}(-\frac{7}{45}x_1 - \frac{17}{12}x_2 + \frac{83}{36}x_3 - \frac{11}{9}x_4 + \frac{2}{3}x_5 \\ & - \frac{37}{180}x_6 + \frac{1}{36}x_7) \end{aligned} \tag{2.11}$$

$$\begin{aligned} \frac{1}{3}\dot{x}_{N-2} + \dot{x}_{N-1} + \frac{1}{3}\dot{x}_N =& \frac{1}{h}(\frac{451}{180}x_N - \frac{1003}{180}x_{N-1} + \frac{20}{3}x_{N-2} - \frac{55}{9}x_{N-3} \\ & + \frac{125}{36}x_{N-4} - \frac{67}{60}x_{N-5} + \frac{7}{45}x_{N-6}) \end{aligned} \tag{2.12}$$

$$\begin{aligned} \dot{x}_{N-1} + \frac{1}{3}\dot{x}_N =& \frac{1}{h}(\frac{35}{36}x_N - \frac{7}{12}x_{N-1} + \frac{7}{36}x_{N-2} - x_{N-3} \\ & + \frac{7}{12}x_{N-4} - \frac{7}{36}x_{N-5} + \frac{1}{36}x_{N-6}) \end{aligned} \tag{2.13}$$

Combining (2.10 - 2.13) into matrix form, we have

$$\mathbf{A}\dot{\mathbf{X}} = \mathbf{B}\mathbf{X} + \mathbf{K} \tag{2.14}$$

where

$$A = \begin{bmatrix} 1 & \frac{1}{3} & & & & \\ \frac{1}{3} & 1 & \frac{1}{3} & & & \\ & \frac{1}{3} & 1 & \frac{1}{3} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{1}{3} & 1 & \frac{1}{3} \\ & & & & \frac{1}{3} & 1 \end{bmatrix}_{(N-1)\times(N-1)} \qquad K = \frac{1}{h}\begin{bmatrix} -\frac{7}{45}x_1 \\ -\frac{1}{36}x_1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}_{(N-1)\times 1}$$

$$B = \frac{1}{h}\begin{bmatrix} -\frac{17}{12} & \frac{83}{36} & -\frac{11}{9} & \frac{2}{3} & -\frac{37}{180} & \frac{1}{36} & \\ -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} & & & \\ -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ & \frac{7}{45} & -\frac{67}{60} & \frac{125}{36} & -\frac{55}{9} & \frac{20}{3} & -\frac{1003}{180} & \frac{451}{180} \\ & \frac{1}{36} & -\frac{7}{36} & \frac{7}{12} & -1 & \frac{7}{36} & -\frac{7}{12} & \frac{35}{36} \end{bmatrix}_{(N-1)\times(N-1)}$$

From 2.14 we have

$$\dot{\mathbf{X}} = \mathbf{E}\mathbf{X} + \mathbf{H} \tag{2.15}$$

where

$$\mathbf{E} = \mathbf{A}^{-1}\mathbf{B}, \quad \mathbf{H} = \mathbf{A}^{-1}\mathbf{K} \tag{2.16}$$

$X^i_{t,s+1} = [x^i_{r,s+1}(t^i_0), x^i_{r,s+1}(t^i_1), \ldots, x^i_{r,s+1}(t^i_N)]$ are the vector functions at the grid points t^i. Applying the CFDS in (2.5 - 2.7) gives

$$\mathbf{A}_r\mathbf{X}^j_{r,s+1} = \mathbf{R}^i_r, \quad \mathbf{X}^i_{r,s+1}(t^{i-1}_N) = \mathbf{X}^{i-1}_r(t^{i-1}_N), \qquad r = 1,2,\ldots,m \tag{2.17}$$

$$\mathbf{A}_r = \mathbf{E} - \alpha_{r,r}\mathbf{I}, \tag{2.18}$$

$$\mathbf{R}_1^i = g_1 + \sum_{k=2}^{m} \alpha_{1,k}\mathbf{X}_{k,s}^i + f_1[\mathbf{X}_{1,s}^i, \ldots, \mathbf{X}_{m,s}^i] - \mathbf{H} \tag{2.19}$$

$$\mathbf{R}_2^i = g_2 + \alpha_{2,1}\mathbf{X}_{1,s+1}^i + \sum_{k=3}^{m} \alpha_{2,k}\mathbf{X}_{k,s}^i + f_2[\mathbf{X}_{1,s+1}^i, \mathbf{X}_{2,s}^i, \ldots, \mathbf{X}_{m,s}^i] - \mathbf{H} \tag{2.20}$$

$$\vdots$$

$$\mathbf{R}_m^i = g_m + \sum_{k=1}^{m-1} \alpha_{m,k}\mathbf{X}_{k,s+1}^i + f_m[\mathbf{X}_{1,s+1}^i, \ldots, \mathbf{X}_{m-1,s+1}^i, \mathbf{X}_{m,s}^i] - \mathbf{H} \tag{2.21}$$

f_r is a diagonal matrix of size $(N-1)\times(N-1)$ and $\mathbf{I}$ is identity matrix of order $N-1$, for $r = 1,2,\ldots,m$. $\mathbf{g}_r$ is g_r multiplied by a vector of ones of size $(N-1)\times 1$. Thus, starting from the initial approximation (2.9), the recurrence formula

$$\mathbf{X}_{r,s+1}^j = \mathbf{A}_r^{-1}\mathbf{R}_r^i, \quad r = 1,2,\ldots,m \tag{2.22}$$

can be used to find the solution $x_r^i(t)$ in the interval $[t_{i-1}, t_i]$. The solution approximating $x_i(t)$ in the entire interval $[a,b] = [t_0, t_F]$ is given by

$$x_r(t) = \begin{cases} x_r^1(t) & , \ t \in [t_0, t_1] \\ x_r^2(t) & , \ t \in [t_1, t_2] \\ \vdots & \\ x_r^F(t) & , \ t \in [t_{F-1}, t_F] \end{cases} \tag{2.23}$$

2.3 Numerical examples

In this section, we apply the proposed MD-CDFRM to systems of IVPs with chaotic and hyperchaotic behavior to illustrate its effectiveness. To demonstrate that the method is as an appropriate tool for solving complex dynamical systems, we consider the chaotic Lorenz, Chen, the Rikitake systems, the hyperchaotic Chua and Robinovich-Fabrikant system. The results obtained are compared to results obtained by the Multi-domain spectral relaxation method (MSRM)[62, 61].

Example 1: Lorenz system

The nonlinear differential equations that describe the Lorenz system [55] is a dynamical

system which is commonly used to explore chaotic behavior. The set of autonomous differential equations is given by

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1), \\ \dot{x}_2 = -x_1 x_3 + b x_1 - x_2, \\ \dot{x}_3 = x_1 x_2 - c x_3, \end{cases} \tag{2.24}$$

where a, b, c are all greater than zero. These system of differential equations were derived by Lorenz [55] in the modeling of two dimensional fluid cell between two parallel plates at different temperatures. In this example, the parameters $\alpha_{r,k}$, g_r and f_r are defined as

$$\begin{gathered} \alpha_{1,1} = -a, \quad \alpha_{1,2} = a, \quad \alpha_{2,1} = b, \\ \alpha_{2,2} = -1, \quad \alpha_{3,3} = -c, \\ f_2 = -x_1 x_3, \quad f_3 = x_1 x_2 \end{gathered} \tag{2.25}$$

with all other $\alpha_{r,k}$ and $g_r = f_r = 0$ for $r, k = 1, 2, 3$. The Lorenz system was solved using $a = 10, b = 28, c = 8/3$ with initial conditions $x_1(0) = 1, x_2(0) = 5, x_3(0) = 10$.

Example 2: Chen system In this example we consider the Chen dynamical system [13]. This is a three-dimensional system of ordinary differential equations with quadratic nonlinearities, defined as

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1), \\ \dot{x}_2 = (c - a)x_1 - x_1 x_3 + c x_2, \\ \dot{x}_3 = x_1 x_2 - b x_3. \end{cases} \tag{2.26}$$

In this example, the parameters $\alpha_{r,k}$, g_r and f_r are defined as

$$\alpha_{1,1} = -a, \;\; \alpha_{1,2} = a, \;\; \alpha_{2,1} = c - a, \;\; \alpha_{2,2} = c,$$

$$\alpha_{3,3} = -b \;\; f_2 = -x_1 x_3, \;\; f_3 = x_1 x_2$$

with all other $\alpha_{r,k}$ and $g_r = f_r = 0$ for $r, k = 1, 2, 3$. The Chen system was solved using the parameters $a = 35, b = 3, c = 28$ and the initial conditions $x_1(0) = -10, \;\; x_2(0) = 0, \;\; x_3(0) = 37$.

Example 3: Rikitake system Here, we consider the Rikitake system [69, 81] which is a model that attempts to explain the reversal of Earth's magnetic field. This system describes the currents of two coupled dynamo disks. The governing equations are

$$\begin{cases} \dot{x}_1 = -bx_1 + x_2x_3, \\ \dot{x}_2 = -bx_2 + x_1(x_3 - a), \\ \dot{x}_3 = 1 - x_1x_2. \end{cases} \tag{2.27}$$

In this example, the parameters $\alpha_{r,k}$, g_r and f_r are defined as

$$\alpha_{1,1} = -b, \quad \alpha_{2,1} = -a, \quad \alpha_{2,2} = -b, \quad f_1 = x_2x_3,$$
$$f_2 = x_1x_3, \quad f_3 = -x_1x_2, \quad g_3 = 1 \tag{2.28}$$

with all other $\alpha_{r,k}$ and $g_r = f_r = 0$ for $r,k = 1,2,3$. The system was solved for the parameters $a = 5, b = 2$ and the initial conditions $x_1(0) = 7, x_2(0) = 11, x_3(0) = 15$ in [69, 81].

Example 4: Hyperchaotic Chua system The Chua system was originally proposed for an electric circuit by Leon Chua in 1983 [46]. The system is a set of equations with a smooth nonlinearity given by

$$\begin{cases} \dot{x}_1 = b(x_2 - ax_1^3 - (1+c)x_1) \\ \dot{x}_2 = x_1 - x_2 + x_3 \\ \dot{x}_3 = -\beta x_2 - \gamma x_3 \end{cases} \tag{2.29}$$

Based on the Chua oscillator, Rech and Albuquerque [68] constructed a new four-dimensional system by introducing a fourth variable x_4 which is an adequate feedback controller to the third equation in system (2.29), to obtain

$$\begin{cases} \dot{x}_1 = b(x_2 - ax_1^3 - (1+c)x_1) \\ \dot{x}_2 = x_1 - x_2 + x_3 \\ \dot{x}_3 = -\beta x_2 - \gamma x_3 + x_4 \\ \dot{x}_4 = -sx_4 + x_2x_3 \end{cases} \tag{2.30}$$

When $a = 0.03, b = 30, c = -1.2, \beta = 50, \gamma = 0.32, s = 0.1060$, the system (2.30) has two positive Lyapunov exponents and hence exhibits a hyperchaotic behavior [14]. The hyperchaotic system (2.30) was solved for the initial conditions $x_1(0) = 3, x_2(0) = 1, x_3(0) = 6, x_4(0) = 1$.

In this example, the parameters $\alpha_{r,k}$, g_r and f_r are defined as

$$\alpha_{1,1} = -b(1+c), \quad \alpha_{1,2} = b, \quad \alpha_{2,1} = 1, \quad \alpha_{2,2} = -1,$$

$$\alpha_{3,2} = -\beta, \quad \alpha_{3,3} = -\gamma$$

$$\alpha_{3,4} = 1, \quad \alpha_{4,1} = -s, \quad f_1 = -bax_1^3, \quad f_4 = x_2x_3$$

with all other $\alpha_{r,k}$ and $g_r = f_r = 0$
for $r,k = 1,2,3,4$.

Example 5: Hyperchaotic Rabinovich-Fabrikant System The Rabinovich-Fabrikant system models the dynamical behavior arising from the modulation instability in a non-equilibrium dissipative medium [20, 56]. The system was introduced by Rabinovich and Fabrikant [67]. The Rabinovich-Fabrikant equation possesses multiple chaotic attractors. The system is described by the following set of equations:

$$\begin{cases} \dot{x}_1 = x_2(x_3 - 1 + x_1^2) + ax_1, \\ \dot{x}_2 = x_1(3x_3 + 1 - x_1^2) + ax_2, \\ \dot{x}_3 = -2x_3(b + x_1x_2), \end{cases} \tag{2.31}$$

with parameters $a,b > 0$. Luo et al [20] reported that different chaotic behaviors are observed for different values of a and b. In this example, the parameters $\alpha_{r,k}$, g_r and f_r are defined as

$$\alpha_{1,1} = a, \quad \alpha_{1,2} = -1, \quad \alpha_{2,1} = 1, \quad \alpha_{2,2} = a,$$

$$\alpha_{3,3} = -2b,$$

$$f_1 = x_2x_3 + x_2x_1^2, \quad f_2 = 3x_1x_3 + x_1^3, \quad f_3 = -2x_1x_2x_3.$$

with all other $\alpha_{r,k}$ and $g_r = f_r = 0$ for $r,k = 1,2,3$. The hyperchaotic systems (2.31) was solved for $a = 0.1, b = 0.98$ with initial conditions $x_1(0) = 0.5, x_2(0) = 6, x_3(0) = 1.1$.

2.4 Results and discussion

In this section, we present the results obtained from implementing the multi-domain compact finite difference relaxation method (MD-CFDRM) to the five examples given in the previous section. To show the performance of the proposed method, we present graphical and tabular

comparisons with the multistage spectral relaxation method (MSRM). We also show plots of the phase portraits of each example. The phase portrait is a plot of a vector field which qualitatively shows how the solutions to the equations will go from a given starting point. In generating all the results presented in this work, it was found that $N = 10$ grid points in each interval (t_{i-1}, t_i) was sufficient to give good accuracy. The MD-CFDRM algorithm was run repeatedly in each interval until the norm of the difference between successive iterations was less than 10^{-6}. The results were validated by computing the residual at each time level.

Table 2.1 Numerical solution for the Lorenz system compared with the MSRM results [62].

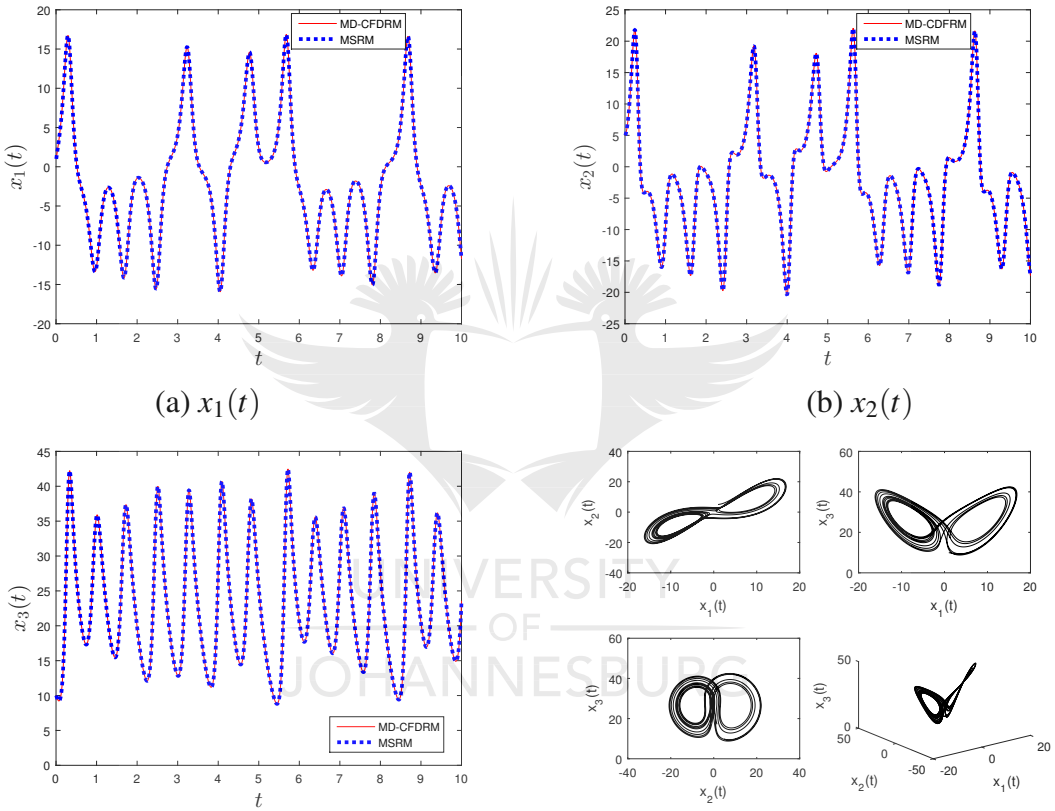
	$x_1(t)$		$x_2(t)$		$x_3(t)$	
t	MD-CFDRM	MSRM	MD-CFDRM	MSRM	MD-CFDRM	MSRM
2	-1.444359	-1.444359	-1.074977	-1.074977	19.517057	19.517057
4	-14.675080	-14.675080	-20.189107	-20.189107	29.063361	29.063362
6	-2.883028	-2.883028	-4.763557	-4.763557	20.355558	20.355558
8	-2.679253	-2.679252	1.429476	1.429476	27.105659	27.105659
10	-12.026645	-12.026646	-17.520281	-17.520281	24.300154	24.300159

Table 2.2 Numerical solution for the Chen system compared with the MSRM results [62].

	$x_1(t)$		$x_2(t)$		$x_3(t)$	
t	MD-CFDRM	MSRM	MD-CFDRM	MSRM	MD-CFDRM	MSRM
1	-15.904923	-15.904923	-13.162222	-13.162222	32.090048	32.090048
2	16.318160	16.318160	6.369432	6.369432	40.655948	40.655948
3	-10.705249	-10.705249	-8.474057	-8.474057	30.502592	30.502592
4	3.771507	3.771509	12.524291	12.524292	37.032302	37.032301
5	-1.751032	-10.751032	2.618147	2.618147	25.328889	25.328889

Table 2.3 Numerical solution for the Rikitake system compared with the MSRM results [62]

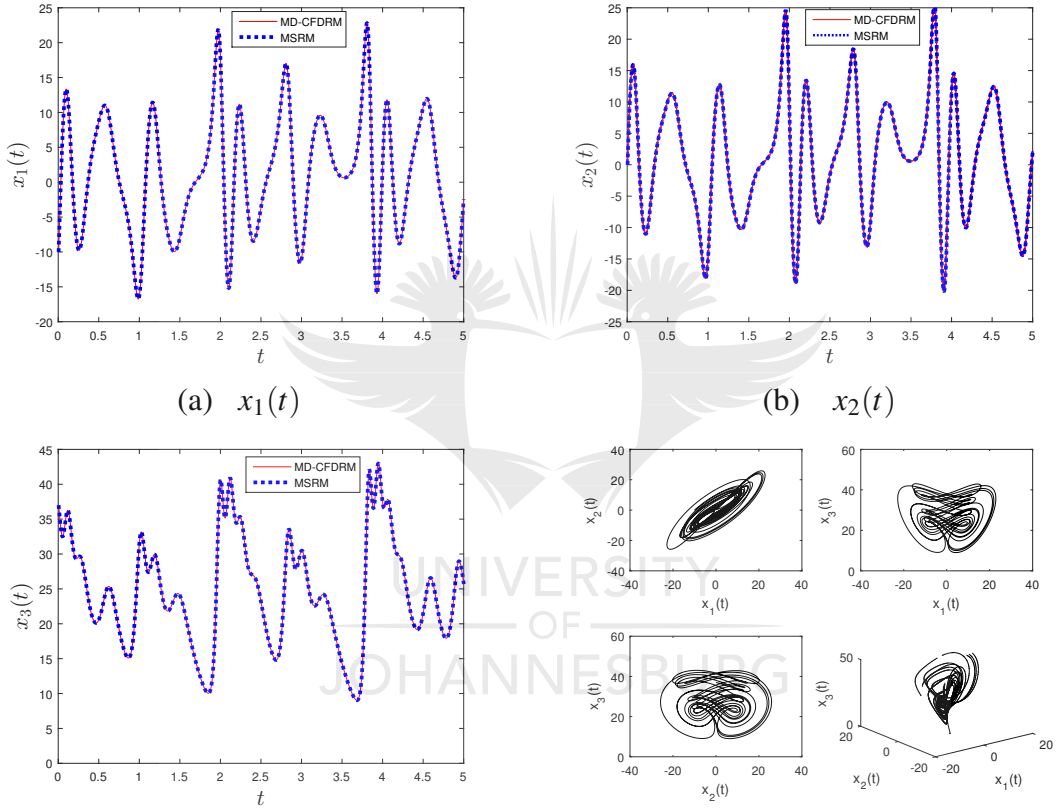
	$x_1(t)$		$x_2(t)$		$x_3(t)$	
t	MD-CFDRM	MSRM	MD-CFDRM	MSRM	MD-CFDRM	MSRM
2	0.982780	0.982780	0.000756	0.000756	3.663515	3.663515
4	-0.061809	-0.061809	-0.018351	-0.018351	5.695371	5.695371
6	-0.641786	-0.641786	-0.372575	-0.372575	7.641621	7.641621
8	-0.219972	-0.219972	0.249986	0.249986	4.871418	4.871418
10	0.439362	0.439362	0.220305	0.220305	6.827688	6.827688



(a) $x_1(t)$ (b) $x_2(t)$

(c) $x_3(t)$ (d) Phase portraits

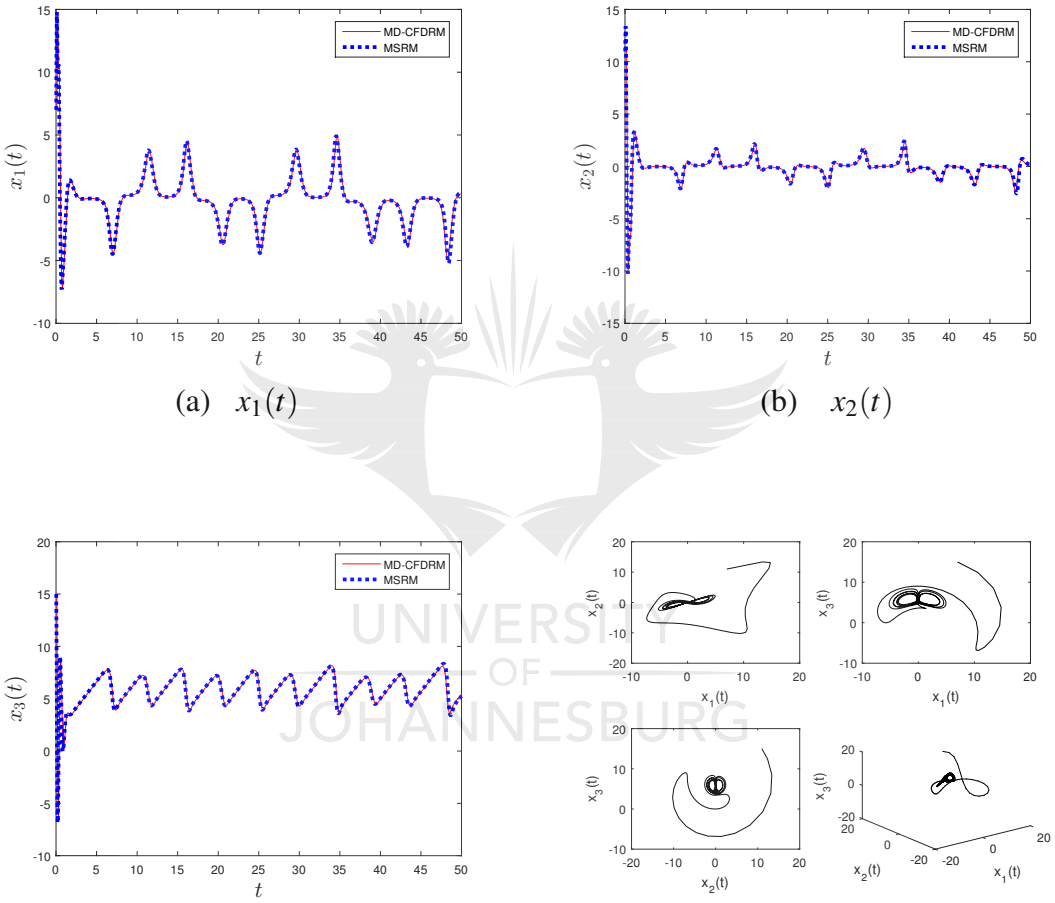
Fig. 2.1 Lorenz system



(a) $x_1(t)$ (b) $x_2(t)$

(c) $x_3(t)$ (d) Phase portraits

Fig. 2.2 Chen system



(a) $x_1(t)$ (b) $x_2(t)$

(c) $x_3(t)$ (d) Phase portraits

Fig. 2.3 Rikitake system

We remark that the accuracy of the MD-CFDRM can also be improved by increasing the number of iterations of the MD-CFDRM algorithm in each interval Ω_i.

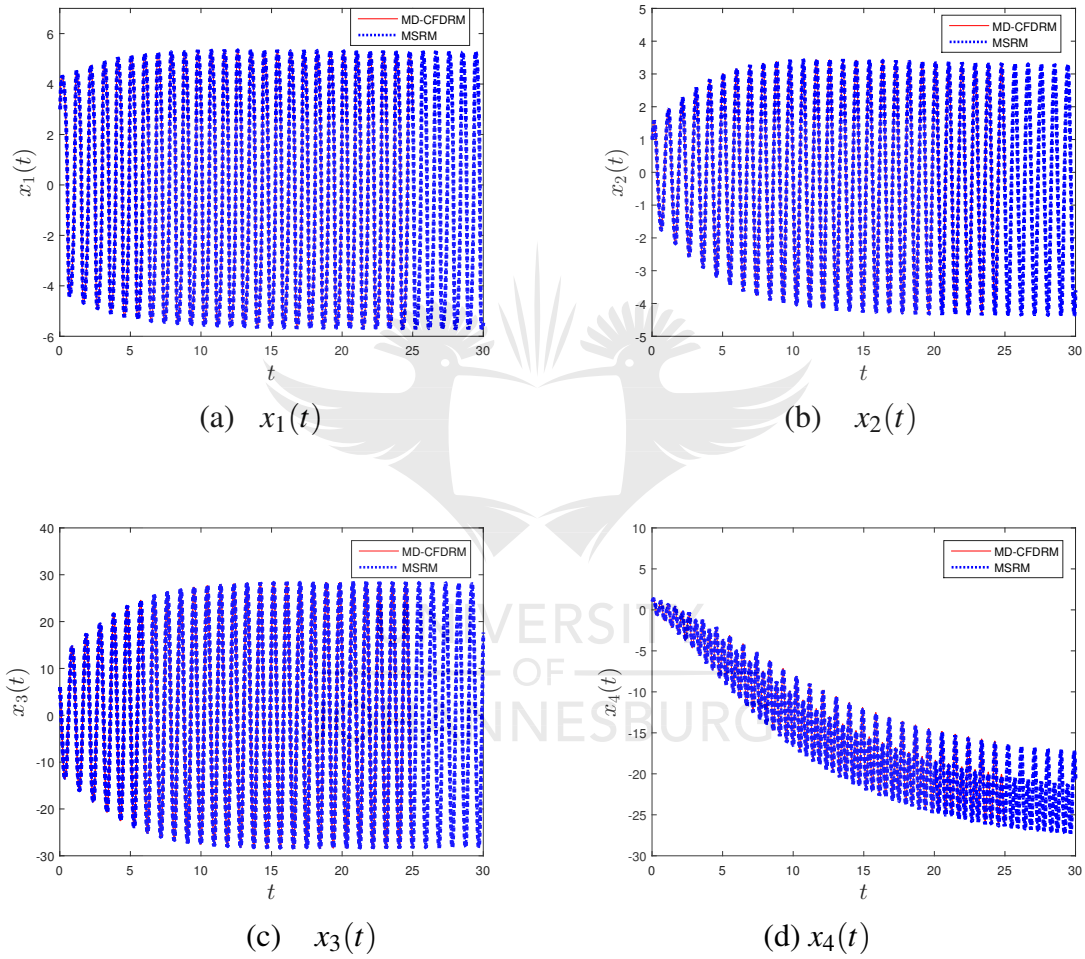
Tables 2.1 - 2.3 show the comparison of the MD-CFDRM with the MSRM results reported in [62] for the Lorenz, Chen and Rikitake systems at selected values of time. The two sets of results are in good agreement. This fact is also shown graphically in Figures 2.1- 2.3. The phase portraits in Figures 2.1d, 2.2d and 2.3d are similar to the ones reported by Motsa et. al. [62].

Table 2.4 Comparison of the numerical solution of the hyperchaotic Chua system obtained by the MD-CFDRM and MSRM.

	$x_1(t)$		$x_2(t)$	
t	MD-CFDRM	MSRM	MD-CFDRM	MSRM
2	-0.170293	-0.170293	1.551695	1.551695
4	3.990219	3.990219	2.525985	2.525985
6	5.159513	5.159513	3.058536	3.058536
8	4.648366	4.648366	1.76664	1.76664
10	2.583704	2.583704	-0.980654	-0.980654
	$x_3(t)$		$x_4(t)$	
	MD-CFDRM	MSRM	MD-CFDRM	MSRM
2	11.014651	11.014651	-1.493298	-1.493298
4	7.011997	7.011997	-3.597274	-3.597274
6	-6.752382	-6.752382	-7.371016	-7.371016
8	-22.334344	-22.334344	-16.055827	-16.055827
10	-27.076591	-27.076591	-22.167142	-22.167142

Table 2.5 Comparison of the numerical solution of Rabinovich-Fabrikant equations obtained by the MD-CFDRM and MSRM for $a = 0.1$ and $b = 0.98$.

	$x_1(t)$		$x_2(t)$		$x_3(t)$	
t	MD-CFDRM	MSRM	MD-CFDRM	MSRM	MD-CFDRM	MSRM
10	-0.193770	-0.193770	5.046955	5.046955	0.964454	0.964454
20	0.030179	0.030179	-1.020136	-1.020136	0.000192	0.000192
30	0.878727	0.878727	-1.845389	-1.845389	0.399523	0.399523
40	-1.029824	-1.029824	1.771524	1.771524	0.000093	0.000093
50	1.037551	1.037551	-0.878634	-0.878634	0.000091	0.000091
60	1.034005	1.034005	0.277127	0.277127	0.002862	0.002862
70	0.798540	0.798540	-0.876128	-0.876128	0.457949	0.457949
80	1.026191	1.026191	-1.979857	-1.979857	0.000876	0.000876



(a) $x_1(t)$ (b) $x_2(t)$

(c) $x_3(t)$ (d) $x_4(t)$

Fig. 2.4 Chua system

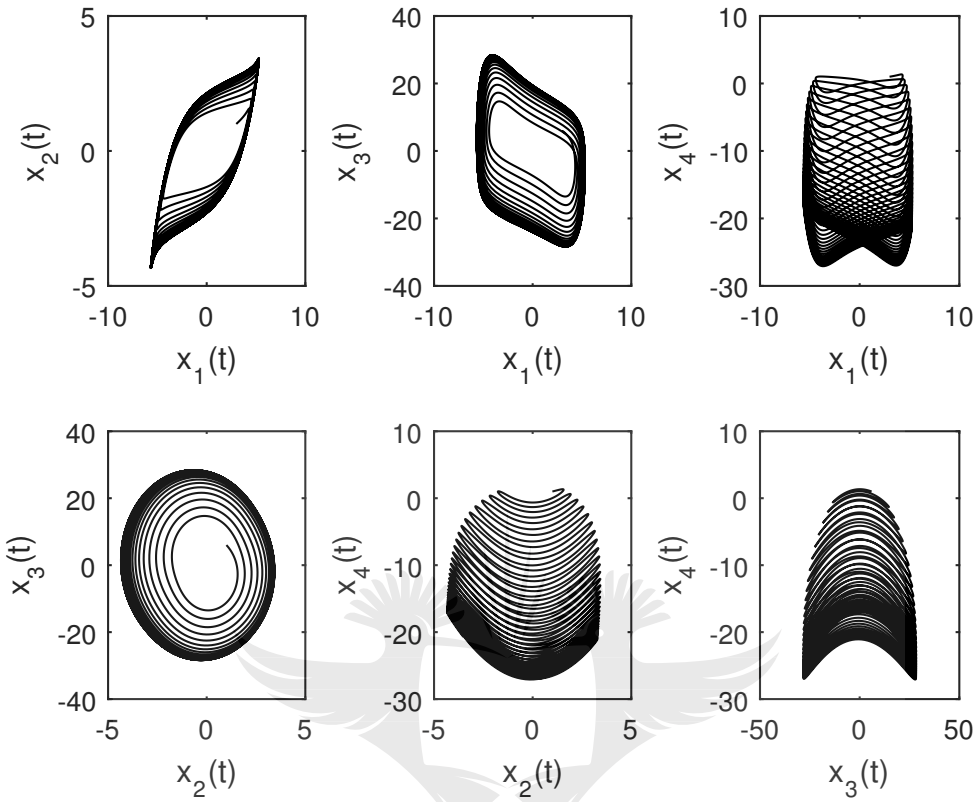


Fig. 2.5 Chua phase portraits

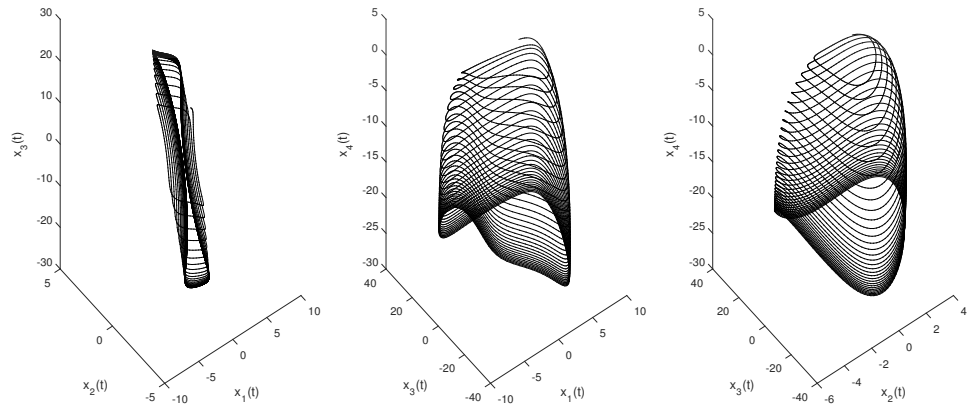
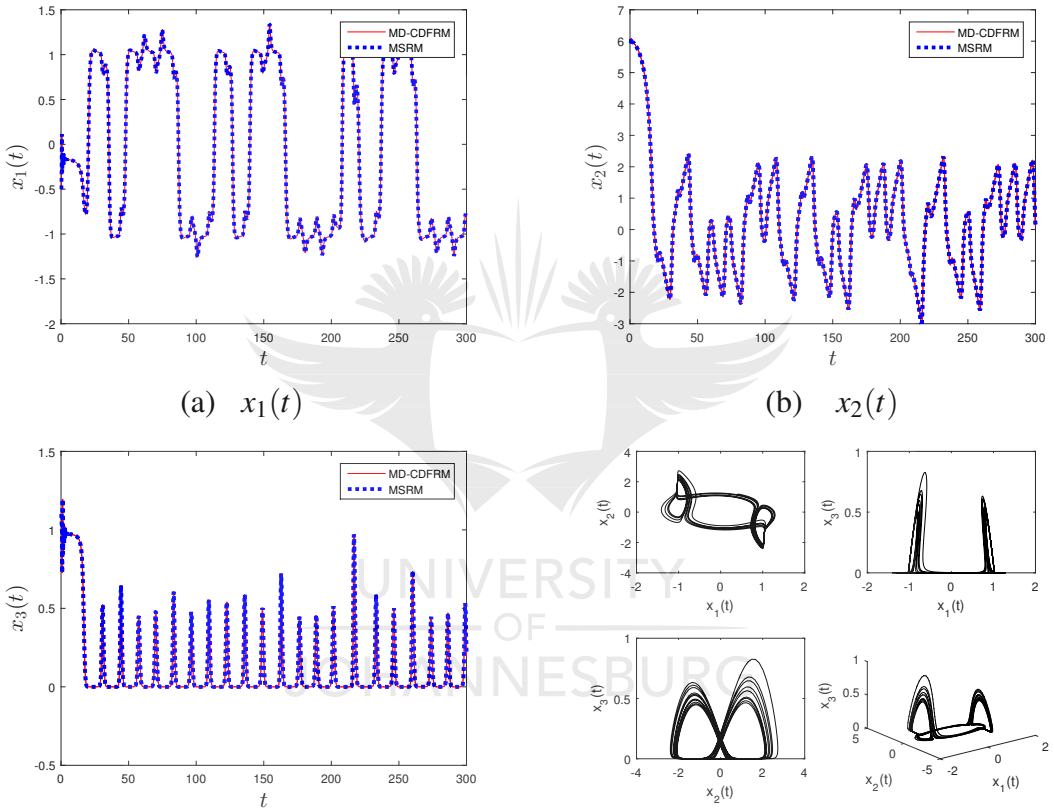


Fig. 2.6 Chua phase portraits

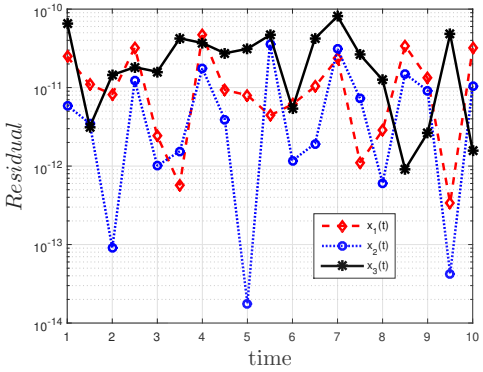


(a) $x_1(t)$ (b) $x_2(t)$

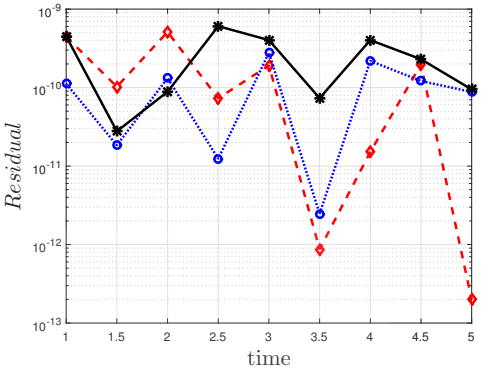
(c) $x_3(t)$ (d) Phase portraits

Fig. 2.7 Rabinovich-Fabrikant system

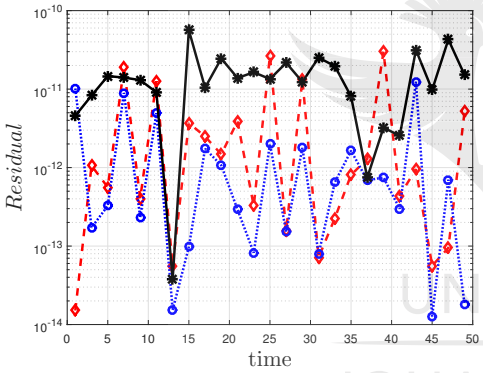
For the hyperchaotic cases, the results are given in Tables 2.4 and 2.5. Again the MD-CFDRM and MSRM [61] yield comparable results. Figures 4 and 7 also depict similar observations. The phase portraits are shown in Figures 2.5 and 2.6 for the Chua system and Figure 2.7d for the Rabinovich-Fabrikant system



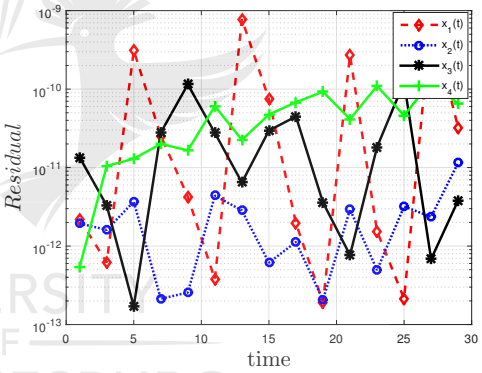
(a) Lorenz system



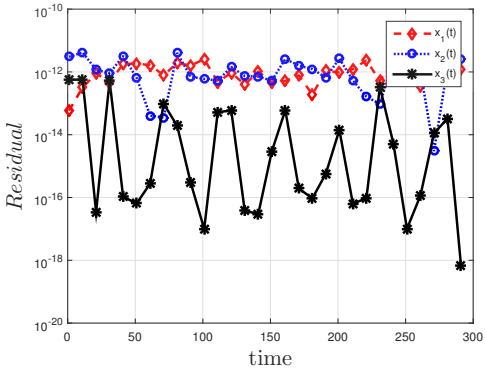
(b) Chen system



(c) Rikitake system



(d) Chua system



(e) Rabinovich-Fabrikant system

Fig. 2.8 Residuals computed at different time levels

The accuracy of the results is determined by computing the residual at each time level. The graphs of the residual against time, for all the examples, are shown in Figure 2.8. It can be seen that the residual varies between $10^{-9} - 10^{-15}$ which shows good accuracy.

2.5 Conclusion

In this work, we have presented the multi-domain compact finite difference method (MD-CFDRM) to solve nonlinear chaotic and hyperchaotic systems. The method is based on blending gauss-Siedel relaxation method and the sixth-order compact finite difference schemes. We used the MD-CFDRM to solve dynamical systems like the chaotic and hyperchaotic sytems. The accuracy of the proposed method was confirmed by comparing tabulated results of the numerical solutions against the MSRM [62, 61] solutions. Graphical results for the time series solutions and corresponding phase portraits of the governing chaotic and hyperchaotic systems confirmed that the MD-CFDRM results were in agreement with the MSRM. The numerical experiments confirms that the MD-CFDRM is highly accurate. We also remark it is computationally efficient and reliable method for solving complex dynamical systems with chaotic and hyperchaotic behavior. The method can easily be extended to general classes of nonlinear IVPs systems arising from a wide variety of other dynamical system applications.

Chapter 3

Multi-Domain Compact finite difference quasi-linearization method for nonlinear evolution partial differential equations

3.1 Introduction

Naturally occurring phenomena and their respective dynamics can be captured accurately using nonlinear partial differential equations(NPDEs). Nonlinear evolution partial differential equations are useful for modeling many naturally occurring phenomena. These are equations arising in a number of fields of science and engineering. They are used to describe many complex nonlinear settings in applications such as vibration and wave propagation, fluid mechanics, plasma physics, quantum mechanics, nonlinear optics, solid state physics, chemical kinematics, physical chemistry, population dynamics, and many other areas of mathematical modeling (see[2, 11]). It is however difficult to obtain analytic solutions of NPDEs due to their nonlinearity and complexities over large time domains. The development of numerical solutions to solve such problems continues to be an active area of research.

Many researchers who prefer finite difference methods to approximate solutions of differential equations are now using higher order compact finite difference schemes (CFDS) as a substitute of the conventional second order finite differences [27, 30, 52, 73, 75]. The CFDS are highly accurate and computationally efficient. This is because they require few grid points to achieve high accuracy. Various CFDS for application such as evaluating higher order derivatives, interpolating and filtering were presented by Lele [51]. The CFDS have been used to solve the Shrödinger's equations [23, 44, 82, 75], Burger's equations [53] and

other equations. Sari and Guraslan [74] used a sixth-order compact finite difference method to solve the one-dimensional sine-Gordon equation.

A number of researchers use the CFDS for space variables and combine them with another numerical technique for the time variable when solving time-dependent PDEs [26, 84, 52, 73, 74]. Dlamini and Khumalo [25] used the CFDS in both space and time in order to improve the accuracy and computational speed in numerical simulations. The numerical method gave accurate results for smaller time domains. In this paper we propose to extend the work of Dlamini and Khumalo [25] by using the multi-domain approach for the time domain. This leads to significant improvements in the accuracy of the method for large time domains. Linear forms of the evolution partial differential equations are obtained by using quasi-linearization method developed by Bellman and Kalaba [10]. The sixth order CFDS applied both in space and time variables of the linearized equations. The method is therefore called multi-domain compact finite difference quasi-linearization method (MD-CFDQLM). The accuracy and reliability of the method is tested by solving a number of nonlinear evolution equations, namely Fishers, Burgers-Fisher, Burgers-Huxley and coupled Burgers equations.

3.2 Multi-domain compact finite difference quasi-linearization method

In this section, we give a brief description of the numerical method of solution and illustrate how it is used to solve nonlinear evolution partial differential equations. Without loss of generality, we consider an *nth* order NPDE of the form:

$$\frac{\partial U}{\partial t} = G\left(U, \frac{\partial U}{\partial x}, \frac{\partial^2 U}{\partial x^2} \ldots, \frac{\partial^n U}{\partial x^n}\right) \tag{3.1}$$

with the physical region $\{(t,x)|t \in [T_0, T_F], x \in [a,b]\}$ The constant n denotes the order of differentiation and G is a function of $U(x,t)$ and its spatial derivatives.

The multi-domain approach assumes that the time interval can be decomposed into P non-overlapping sub-intervals. Let $t \in \omega$, where $\omega = [T_0, T_F]$, be the time interval where the solution of the non-linear PDE exist. The set of intervals are defined as

$$\omega_l = (t_{l-1}, t_l),\ l = 1,2,\ldots,P \ \ with\ \ T_0 = t_0 < t_1 < t_2 <, \ldots, < t_P = T_F \tag{3.2}$$

The main idea of the multi-domain approach is that of determining the solution of equation (3.1) independently on each sub-interval, one at a time, beginning at the initial condition. The given initial condition is considered to be the left boundary of the time interval. The given initial condition is used to compute the solution in the first sub-interval. The computed solution at the right hand boundary of the first interval is used as an initial condition in the subsequent second sub-interval. The computed solution in the first interval is then used to compute the solution in the second sub-interval. The same process is repeated until the last sub-interval. The process where the solutions in different intervals are matched along their common boundary is called patching. The patching condition requires that

$$u^{(l)}(x,t_{l-1}) = u^{(l-1)}(x,t_{l-1}),\ x \in [a,b] \tag{3.3}$$

where $u^{(l)}(x,t)$ denotes the solution of equation (3.1) at each sub-interval ω_l with $1 \leq l \leq P$. Therefore, in each sub-interval, we are required to solve the nonlinear parabolic equation

$$\frac{\partial u^l}{\partial t} = G\left(u^l, \frac{\partial u^l}{\partial x}, \frac{\partial^2 u^l}{\partial x^2} \ldots, \frac{\partial^n u^l}{\partial x^n}\right),\ t \in [T_0, T_F],\ x \in [a,b] \tag{3.4}$$

subject to

$$u^{(l)}(x,t_{l-1}) = u^{(l-1)}(x,t_{l-1}), \tag{3.5}$$

where $l = 1,2,\ldots,P$. To solve (3.4) in each interval we first use the quasi-linearization technique to linearize (3.1), then solve the derivatives using the higher order compact finite difference schemes. Equation (3.4) can be expressed in the form:

$$H[u^l_{(x,0)}, u^l_{(x,1)}, u^l_{(x,2)}, \ldots, u^l_{(x,n)}] - u^l_{(t,1)} = 0, \tag{3.6}$$

where $u^l_{(x,n)}$ denote the nth partial derivative of $u(x,t)$ with respect to x in the lth sub-interval. Similarly, $u^l_{(t,1)}$ the first partial derivative with respect to t in the lth sub-interval and H is the nonlinear operator.If we assume that the difference $u^{(l)}_{(x,0,s+1)} - u^{(l)}_{(x,0,s)}$ and all its space derivatives is small, then we can approximate the nonlinear operator H using the linear terms of the Taylor series and thus

$$H[u^l_{(x,0)}, u^l_{(x,1)}, u^l_{(x,2)}, \ldots, u^l_{(x,n)}] \approx H[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}]$$

$$+ \sum_{k=0}^{n} \frac{\partial H}{\partial u^l_{(x,k)}} (u^{(l)}_{(x,k,s+1)} - u^{(l)}_{(x,k,s)}) \tag{3.7}$$

where s and $s+1$ denote previous and current iterations respectively.

Let

$$\frac{\partial H}{\partial u^l_{(x,k)}}\left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right]$$
$$= \Omega^{(l)}_{(k,s)}[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}] \tag{3.8}$$

Therefore equation (3.7) can be expressed as

$$H[u^l_{(x,0)}, u^l_{(x,1)}, u^l_{(x,2)}, \ldots, u^l_{(x,n)}] \approx H[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}]$$
$$+ \sum_{k=0}^{n} \Omega^{(l)}_{(k,s)}[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}]u^l_{(x,k,s+1)}$$
$$- \sum_{k=0}^{n} \Omega^{(l)}_{(k,s)}[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}]u^l_{(x,k,s)} \tag{3.9}$$

Let

$$R^{(l)}_s[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}] = \sum_{k=0}^{n} \Omega^{(l)}_{(k,s)}[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}]u^l_{(x,k,s)}$$
$$- H[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}] \tag{3.10}$$

Equation (3.8) can be expressed as

$$H[u^l_{(x,0)}, u^l_{(x,1)}, u^l_{(x,2)}, \ldots, u^l_{(x,n)}] \approx \sum_{k=0}^{n} \Omega^{(l)}_{(k,s)}[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}]u^l_{(x,k,s+1)}$$
$$- R^{(l)}_s[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}] \tag{3.11}$$

Substituting equation (3.11) into equation (3.6), we get linearized form of equation (3.4) as follows:

$$\sum_{k=0}^{n} \Omega^{(l)}_{k,s} u^{(l)}_{(x,k,s+1)} - u^{(l)}_{(t,1,s+1)} = R^{(l)}_s[u^l_{(x,0,s)}, u^l_{(x,1,s)}, u^l_{(x,2,s)}, \ldots, u^l_{(x,n,s)}] \tag{3.12}$$

Using the CFDS to evaluate (3.12) at each grid point (x_i, t_j), we get

$$\sum_{k=0}^{n} \mathbf{\Omega}_{k,s} E_{(x,k)} U^{(l)}_{\cdot,j,s+1} - \sum_{k=1}^{N_t} e_{j,k} U^{(l)}_{\cdot,k,s+1} = R^{(l)}_s - \sum_{k=1}^{n} \mathbf{\Omega}_{k,s} H_{(x,k)} \tag{3.13}$$

for $j = 1,2,\ldots,N_t$, where $\mathbf{\Omega}_{k,s}$ is a diagonal matrix given by:

$$\mathbf{\Omega}_{k,s} = \begin{bmatrix} \Omega_{k,s}(x_2,t_j) & & & \\ & \Omega_{k,s}(x_1,t_j) & & \\ & & \ddots & \\ & & & \Omega_{k,s}(x_{N_x-1},t_j) \end{bmatrix} \tag{3.14}$$

and

$$U_{\cdot,j}^{(l)} = [u^{(l)}(x_2,t_j), u^{(l)}(x_3,t_j), \ldots, u^{(l)}(x_{N_x-1},t_j)]$$

Since the initial condition is known, then we express equation (3.13) as

$$\sum_{k=0}^{n} \mathbf{\Omega}_{k,s} E_{(x,k)} U_{\cdot,j,s+1}^{(l)} - \sum_{k=2}^{N_t} e_{j,k} U_{\cdot,k,s+1}^{(l)} = R_s^{(l)} - \sum_{k=1}^{n} \mathbf{\Omega}_{k,s} H_{(x,k)} - e_{j,1} U_{\cdot,1}^{(l)}, \qquad j = 2,3,\ldots,N_t \tag{3.15}$$

Equation (3.15) can be expressed as the following $(N_t-1)(N_x-1) \times (N_t-1)(N_x-1)$ matrix system

$$\begin{bmatrix} A_{2,2} & A_{2,3} & \ldots & A_{2,N_t} \\ A_{3,2} & A_{3,3} & \ldots & A_{3,N_t} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t,2} & A_{N_t,3} & \ldots & A_{N_t,N_t} \end{bmatrix} \begin{bmatrix} U_{\cdot,2}^{(l)} \\ U_{\cdot,3}^{(l)} \\ \vdots \\ U_{\cdot,N_t}^{(l)} \end{bmatrix} = \begin{bmatrix} B_2^{(l)} \\ B_3^{(l)} \\ \vdots \\ B_{N_t}^{(l)} \end{bmatrix}, \tag{3.16}$$

where

$$A_{i,i} = \sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{E}_{(x,k)} - \mathbf{e}_{i,i} I, \quad i = 2,3,\ldots,N_t \tag{3.17}$$

$$A_{i,j} = -\mathbf{e}_{i,j} I, \ \textit{when} \ \ i \neq j \quad i,j = 2,3,\ldots,N_t \tag{3.18}$$

$$B_j = \mathbf{R}_s^{(l)} - \sum_{k=1}^{n} \mathbf{\Omega}_{k,s} \mathbf{H}_{(x,k)} - \mathbf{e}_{j,1} U_{s,1}^{(l)}, \quad j = 2,3,\ldots,N_t \tag{3.19}$$

E,H, **R** and **e** are the Sixth order compact schemes defined in chapter (1), I is the identity matrix of size $(N_x-1) \times (N_x-1)$ and $E_{(x,0)} = I$. Solving equation (3.15) gives $u^{(l)}(x_i,t_j)$ as the approximate solution for equation (3.4) at each sub-interval.

3.3 Numerical Experiments

In this section, we apply the proposed MD-CDFQLM to well-known nonlinear PDEs of the form (3.1) with exact solutions. In order to determine the level of accuracy of the MD-

CFDQLM approximation solution, at a particular time level, in comparison with the exact solution we report maximum error and compare the MD-CFDQLM with CFDQLM [27] results. In this work,we consider nonhomogeneous Dirichlet boundary conditions. The space and time domains are given by $[a,b] = [0,5]$ and $[t_0,T] = [0,10]$ respectively for most of the numerical experiments. We choose a $T = 10$ to show the accuracy of the algorithm over a large time domains.

Example 1: Fisher's equation

We consider the Fisher's equation which represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. The equation is given by

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u(1-u), \tag{3.20}$$

subject to the initial condition

$$u(x,0) = \frac{1}{\left(1+e^{(\sqrt{\frac{\alpha}{6}}x)}\right)^2}, \tag{3.21}$$

and exact solution (Wazwaz and Gorguis [80])

$$u(x,t) = \frac{1}{\left(1+e^{(\sqrt{\frac{\alpha}{6}}x-5\alpha t/6)}\right)^2}, \tag{3.22}$$

where α is a constant (set to be $\alpha = 1$ in this study). The Fisher equation represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. The Fisher's equation has been solved by Olmos et al. [64], Hariharan et al. [39] using nonstandard finite differences, spectral collocation and Haar wavelet method respectively. For this example, the appropriate nonlinear operator H are chosen as

$$H(u) = u'' + \alpha u - \alpha u^2 \tag{3.23}$$

The primes denote differentiation with respect to x. We use $[a,b] = [0,5]$ and $[t_0,T] = [0,10]$ as our x and t domains, respectively. We linearize the nonlinear operator H by expanding using the Taylor series expansion. We assume that the difference $u_{s+1}^{(l)} - u_s^{(l)}$ and it's derivatives is very small.

Example 2: Burgers-Fisher equation

We consider the generalized Burgers-Fisher equation [35]

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta}), \tag{3.24}$$

subject to the initial condition

$$u(x,0) = \Big[\frac{1}{2} + \frac{1}{2}\tanh\Big(\frac{-\alpha\delta}{2(\delta+1)}x\Big)\Big]^{\frac{1}{\delta}} \tag{3.25}$$

and exact solution

$$u(x,t) = \Big[\frac{1}{2} + \frac{1}{2}\tanh\Big(\frac{-\alpha\delta}{2(\delta+1)}\Big[x - \Big(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}\Big)t\Big]\Big)\Big]^{\frac{1}{\delta}} \tag{3.26}$$

where α, β and δ are parameters. In this work, the parameters are chosen to be $\alpha = \beta = \delta = 1$. The Burgers-Fisher equation has been solved by Javidi et al. [43], Moghimi and Hejazi [60], Golbabai and Javidi [36] using spectral collocation method, spectral domain decomposition and Homotopy analysis method respectively. The nonlinear operator H is chosen as

$$H(u) = u'' + u - uu' - u^2 \tag{3.27}$$

We use $[a,b] = [0,5]$ and $[t_0,T] = [0,10]$ as our x and t domains, respectively. We linearize the nonlinear operator H by expanding using the Taylor series expansion. We assume that the difference $u_{s+1}^{(l)} - u_s^{(l)}$ and it's derivatives is very small.

Example 3: Burgers-Huxley equation

We consider the generalized Burgers-Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta})(u^{\delta} - \gamma), \tag{3.28}$$

where $\alpha, \beta \geq 0$ are constant parameters, δ is a positive integer (set to be $\delta = 1$ in this study) and $\gamma \in (0,1)$. The exact solution subject to the initial condition

$$u(x,0) = \frac{1}{2} - \frac{1}{2}\tanh\Big[\frac{\beta}{r-\alpha}x\Big], \tag{3.29}$$

is

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\Big[\frac{\beta}{r-\alpha}(x-ct)\Big], \tag{3.30}$$

where

$$r = \sqrt{\alpha^2 + 8\beta}, \quad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4} \tag{3.31}$$

The Burgers-Huxley has been solved by Batiha et al. [9], Darvishi et al. [21], Dehghan et al. [22] using the VIM, spectral collocation and finite differences, respectively. In this example, the nonlinear operator H is chosen as

$$H(u) = u'' - \beta\gamma u - \alpha u u' + \beta(1+\gamma)u^2 - \beta u^3 \tag{3.32}$$

Example 4: Consider the Coupled Burgers equations

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - 2u\frac{\partial u}{\partial x} + \frac{\partial (uv)}{\partial x} = 0 \tag{3.33}$$

$$\frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial x^2} - 2v\frac{\partial v}{\partial x} + \frac{\partial (vu)}{\partial x} = 0 \tag{3.34}$$

with the initial conditions

$$u(x,0) = v(x,0) = sin(x) \tag{3.35}$$

and the exact solutions

$$u(x,t) = v(x,t) = e^{-t} sin(x) \tag{3.36}$$

The nonlinear operator H is chosen as

$$H(u) = u'' - 2uu' + (uv)' \tag{3.37}$$

$$H(v) = u'' - 2vv' + (uv)' \tag{3.38}$$

We use $[a,b] = [c,d] = [0,5]$ and $[t_0,T] = [0,10]$ as our x and t domains, respectively where $[a,b]$ are nodes for u and $[c,d]$ are nodes for v.

3.4 Results and Discussion

In this section, we present and discuss the numerical results. To establish the accuracy of the proposed method, we compute the maximum error E_{N_x} which is the maximum difference between the approximate solution and the exact solution at each time level, that is,

$$E_{N_x} = \max_k |u_a(x_k,t) - u_e(x_k,t)|, : 0 \leq k \leq N_x \tag{3.39}$$

where $u_a(x_k,t)$ is the approximate solution and $u_e(x_k,t)$ is the exact solution at the time level t. We analyze the effects of dividing the domain into sub-domains on the accuracy and computational time. This is done by solving each example with $p = 1$ and $p = 10$ (p is the number of sub-domains). The method is just the compact finite difference quasi-linearization method (CFDQLM) when $p = 1$. Dlamini and Khumalo [25] introduced the CFDQLM to solve the same examples considered in this work but they only considered small time domains. They obtained highly accurate results. In this work we deal with large time domains. We compare the results for $p = 1$ and $p = 10$ and display results in tables. In all cases we used $N_t = 10$ and varied the values of N_x.

Table 3.1 Maximum error estimates E_{N_x} for the Fishers equation, with $N_t = 10$.

	CFDQLM		MD-CFDQLM	
t/N_x	20	50	20	30
1.0	4.805e-004	4.807e-004	4.733e-009	1.237e-010
2.0	1.132e-004	1.132e-004	8.466e-009	1.677e-010
3.0	4.259e-005	4.262e-005	5.192e-009	3.030e-011
4.0	1.542e-005	1.543e-005	1.090e-009	1.862e-011
5.0	3.168e-006	3.169e-006	9.989e-010	1.387e-011
6.0	1.289e-006	1.289e-006	1.039e-010	3.344e-012
7.0	5.391e-007	5.395e-007	8.829e-011	1.242e-013
8.0	3.798e-007	3.798e-007	2.194e-011	2.095e-013
9.0	7.504e-007	7.517e-007	3.861e-012	4.741e-014
10.0	4.099e-006	4.109e-006	1.218e-012	2.753e-014
CPU time:	0.123795s	0.205661s	0.116480s	0.138220s

In Table 3.1, the maximum errors for the Fisher equation are displayed. We observe much better accuracy for the MD-CFDQLM than the CFDQLM. The computational time is also slightly lower for the MD-CFDQLM. This is because it achieves high accuracy on remarkably small number of grid points Using $N_x = 20$ for the MD-CFDQLM produces more accurate results for the CFDQLM with $N_x = 20$ and $N_x = 50$. For the CFDQLM increasing N_x does not improve the solution as it can be seen from Table 3.1.

Results of the Burgers-Fisher equation are shown in Table 3.2. Again the MD-CFDQLM shows superiority in terms of accuracy and computational speed when compared to the CFDQLM. The same trend is observed for the results of the Burgers-Huxley which are shown in Table 3.3. The same observations are seen in Figure 3.4 which shows the maximum errors for each of the equations considered. It can be seen that MD-CFDQLM gives highly

Table 3.2 Maximum error estimates E_{N_x} for the Burgers-Fisher equation, with $N_t = 10$.

	CFDQLM		MD-CFDQLM	
t/N_x	20	50	20	30
1.0	4.498e-004	4.508e-004	1.069e-008	8.357e-010
2.0	1.569e-004	1.569e-004	6.020e-009	1.812e-010
3.0	6.663e-005	6.665e-005	1.564e-009	9.970e-011
4.0	1.869e-005	1.869e-005	3.333e-010	4.445e-011
5.0	5.022e-006	5.022e-006	7.224e-011	1.768e-012
6.0	1.646e-006	1.647e-006	1.540e-011	1.128e-012
7.0	8.277e-008	8.300e-008	2.664e-012	2.128e-013
8.0	9.022e-007	9.029e-007	6.029e-013	5.673e-014
9.0	2.356e-006	2.358e-006	1.482e-013	3.320e-014
10.0	1.074e-005	1.074e-005	3.586e-014	1.621e-014
CPU time:	0.025197s	0.421462s	0.015197s	0.074305s

accurate solutions with errors of at least 10^{-8} for the values of t considered. Another remarkable observation is that the method gives accurate results even for large time domains. A lot of methods loose significant accuracy when t is very big.

Table 3.3 Maximum error estimates E_{N_x} for the Burgers-Huxley equation, with $N_t = 10$.

	CFDQLM		MD-CFDQLM	
t/N_x	40	50	30	40
1.0	6.565e-006	6.570e-006	7.169e-008	6.711e-010
2.0	2.215e-006	2.214e-006	1.083e-007	1.008e-009
3.0	8.862e-007	8.868e-007	8.132e-008	1.865e-009
4.0	1.629e-007	1.628e-007	4.554e-008	5.667e-010
5.0	3.330e-008	3.336e-008	3.202e-008	3.244e-010
6.0	5.831e-008	5.835e-008	4.444e-008	7.128e-010
7.0	2.665e-007	2.665e-007	2.979e-008	3.439e-010
8.0	4.883e-007	4.887e-007	1.196e-007	1.629e-009
9.0	1.133e-006	1.133e-006	3.779e-008	1.353e-009
10.0	5.361e-006	5.360e-006	1.088e-007	1.391e-009
CPU time:	0.236870s	0.423499s	0.187576s	0.218691s

Figure 3.2 shows good agreement between the exact and approximation solution for the proposed method. Table 3.4 shows the maximum errors for the coupled Burgers' system. Even for the system of equations the MD-CFDQLM gives more accurate results and is more

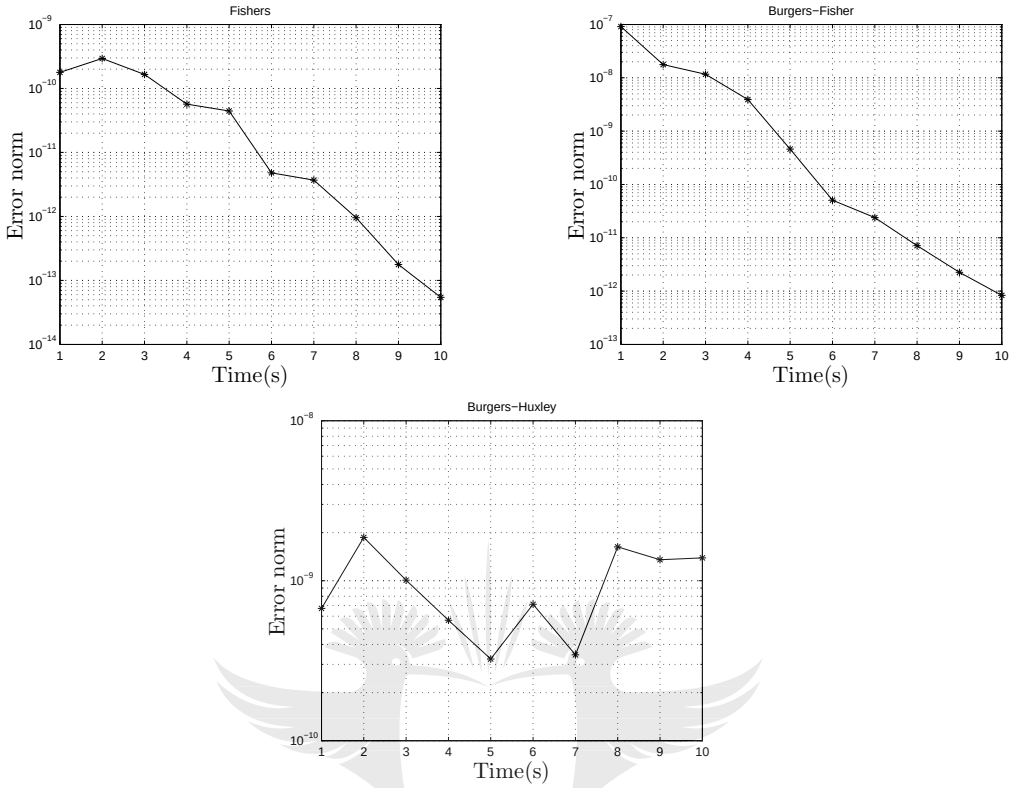


Fig. 3.1 Error norms E_{N_x} at different time levels

Table 3.4 Maximum error estimates E_{N_x} for the coupled Burgers system, with $N_t = 10$.

	CFDQLM	$N_x = 40$	MD-CFDQLM	$N_x = 20$
t/N_x	$u(x,t)$	$v(x,t)$	$u(x,t)$	$v(x,t)$
1.0	1.04232e-001	1.04232e-001	8.85687e-008	8.85687e-008
2.0	3.83453e-002	3.83453e-002	4.74018e-008	4.74018e-008
3.0	1.41068e-002	1.41068e-002	2.56153e-008	2.56153e-008
4.0	5.18979e-003	5.18979e-003	1.25629e-008	1.25629e-008
5.0	1.90935e-003	1.90935e-003	5.78707e-009	5.78707e-009
6.0	7.02498e-004	7.02498e-004	2.56926e-009	2.56926e-009
7.0	2.58493e-004	2.58493e-004	1.10716e-009	1.10716e-009
8.0	9.51340e-005	9.51340e-005	4.66891e-010	4.66893e-010
9.0	3.50247e-005	3.50247e-005	1.93682e-010	1.93683e-010
10.0	2.81025e-007	2.81025e-007	7.93165e-011	7.93174e-011
CPU time:	0.914139s	0.914139s	0.800893s	0.800893s

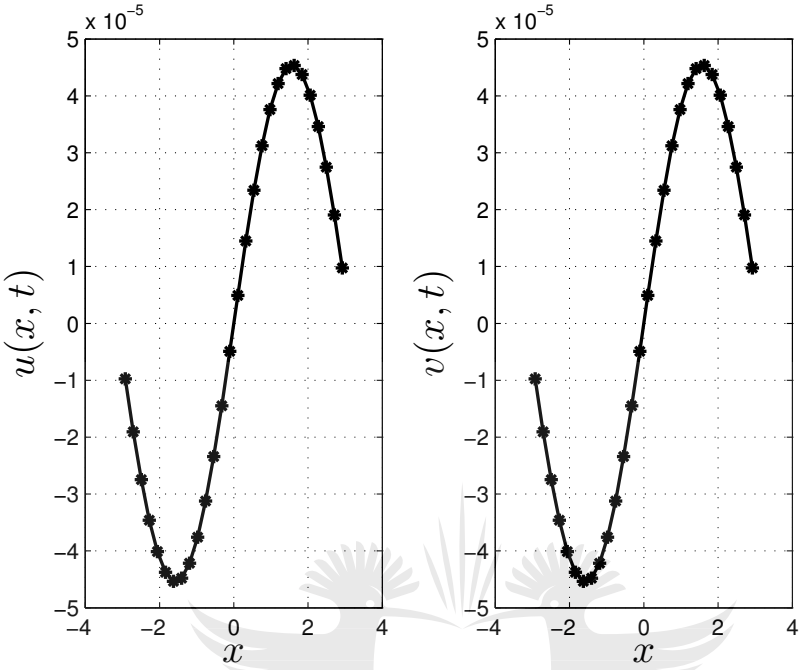


Fig. 3.2 Exact (dots)and MD-CFDQLM approximate (line) solution for the coupled Burgers system

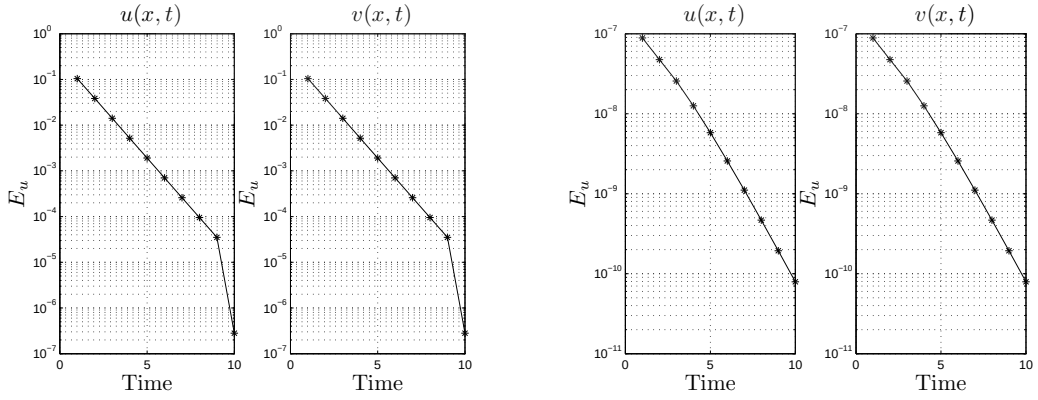


Fig. 3.3 Error norms (E_{N_x}) for the (a) CFDQLM and the (b) MD-CFDQLM at different time levels

efficient in terms of speed than the CFDQLM. This fact is also portrayed graphically in Figure 3.3, where we see the maximum error ranging between 10^{-7} and 10^{-11}.

3.5 Conclusion

In this paper, we have extended and implemented for the first time a new approach named the multi-domain compact finite difference quasi-linearization method (MD-CDFQLM) for solving nonlinear parabolic PDEs. The method uses compact finite difference schemes in both space and time. The time domain is also divided into small sub-domains and the equation solved in each sub-domain. As a result the method is able to handle problems with large time-domains. This idea was tested on Fisher's, Burgers-Fisher, Burgers-Huxley, and Burgers systems. The MD-CFDQLM method gave more accurate results than the CFDQLM [25] over large domains.

Chapter 4

Numerical solutions of the generalized Kuramoto-Sivashinsky equation using Multi-Domain Bivariate Pseudospectral Method

4.1 Introduction

In this chapter, we consider the generalized Kuramoto-Sivashinsky (GKS) equation

$$u_t + uu_x + \alpha u_{xx} + \beta u_{xxx} + \gamma u_{xxxx} = 0, \quad x \in [a,b], t \in [0,T] \tag{4.1}$$

where α, β and γ are constants.

For $\beta = 0$, the equation is called the Kuramoto-Sivashinsky (KS) equation.

$$u_t + uu_x + \alpha u_{xx} + \gamma u_{xxxx} = 0, \quad x \in [a,b], t \in [0,T] \tag{4.2}$$

The KS equation is a nonlinear evolution partial differential equation with applications in various physical phenomena, such as plasma and chemical reaction dynamics [18, 47], reaction diffusion systems [47], long waves on thin films and on the interface between two viscous fluids [40], two-phase flows in cylindrical pipes [66] and many more [38, 42, 47, 54, 58, 19, 76].

A number of numerical schemes have been used to compute solutions of the KS equation. These include the local discontinuous Galerkin method [83], tanh function method [41], homotopy analysis method [48], inverse scattering method [29], homogeneous balance

method [31], Lattice Boltzmann method [49], cubic B-spline finite difference-collocation method [50], quintic B spline collocation method [59], a higher-order finite element approach [7], finite difference discretization [4], fourth-order singly diagonally implicit Runge-Kutta method [24], etc.

The KS equation exhibit chaotic behaviour under certain conditions. Rapidly changing solutions and high sensitivity to small perturbations of the initial data are some of the characteristics of chaotic differential equations. These characteristics pose some problems when it comes to numerical simulations especially for large spatial and temporal domains. This is because, for large domains, the computed numerical solutions may fail to converge even when the step size is decreased, thus compromising the accuracy and efficiency of the numerical simulations. It often requires using a large number of grid points. However, that leads to large memory requirements. Also, for methods like the pseudo-spectral method the approximations can exhibit spurious oscillations which can lead to nonlinear instabilities if the number of grid points is very big. Because of the success over short time intervals, modifications have been made to a number of numerical methods to be able to deal with chaotic behaviour over large intervals. Such modifications involve domain decomposition techniques, in which the domain of the problem is divided into two or more sub-domains. Multi-domain approaches allow better conditioned matrices and larger step sizes than single domain computations. Basically, the domain of the problem is divided into two or more sub-domains and the problem solved in each sub-domain with appropriate interface conditions connecting the solution across the sub-domain boundaries.

Motsa et al [62, 61] have used the idea of domain decomposition to solve chaotic and hyperchaotic ordinary differential equations. They used the Chebyshev spectral method to discretize the time variable over multiple intervals making the entire domain of the problem. The method was able to capture rapidly changing solutions even for large time intervals. In this work we extend Motsa et al's approach to chaotic partial differential equations, with the KS equation as an example. We use the ideas of Magagula et al [57] by employing the Chebyshev spectral method to discretize on both space and time and using the multi-domain approach in the time variable. This leads to significant improvements in the accuracy of the method for large values of time t. Before applying the Chebyshev spectral method the KS equation is linearized using the quasi-linearization technique introduced by Bellman and Kalaba [10]. Results show that for small values of t, the single domain spectral method gives accurate results, but the accuracy deteriorates as t increases. On the other hand, the multi-domain approach produces accurate results even for large t.

4.2 Multi-domain bivariate spectral quasi-linearisation method (MD-BSQLM)

In this section we detail the multi-domain spectral quasi-linearization method and illustrate how it is used to solve the GKS equation. The method is a combination of 3 elements which are the bivariate Chebyshev spectral method, multi-domain approach and quasi-linearization technique. The following subsections describe each of these elements and how they combine to form the MD-BSQLM and how it is used to solve the GKS equation

4.2.1 Quasi-linearization

In this section we illustrate how we use the quasi-linearization technique to linearize any 4^{th} order parabolic PDE of the form

$$u_t = H\Big[u_{(x,0)}, u_{(x,1)}, u_{(x,2)}, u_{(x,3)}, u_{(x,4)}\Big], \tag{4.3}$$

where H is a nonlinear function of $u(x,t)$ and it's space derivatives.If we assume that the difference $u_{s+1} - u_s$ and all its space derivatives is small where s and $s+1$ denote previous and current iterations, respectively, the nonlinear function H can be approximated by using the linear terms of the Taylor series and thus H

$$\begin{aligned} H\Big[u_{(x,0)}, u_{(x,1)}, u_{(x,2)}, u_{(x,3)}, u_{(x,4)}\Big] \approx{}& H\Big[u_{(x,0,s)}, u_{(x,1,s)}, u_{(x,2,s)}, u_{(x,3,s)}, u_{(x,4,s)}\Big] \\ &+ \sum_{k=0}^{4} \frac{\partial H}{\partial u_{(x,k)}}\Big(u_{(x,k,s+1)} - u_{(x,k,s)}\Big) \end{aligned} \tag{4.4}$$

Let

$$\frac{\partial H}{\partial u_{(x,k)}}\Big[u_{(x,0,s)}, u_{(x,1,s)}, u_{(x,2,s)}, u_{(x,3,s)}, u_{(x,4,s)}\Big] = \Omega_{(k,s)}\Big[u_{(x,0,s)}, u_{(x,1,s)}, u_{(x,2,s)}, u_{(x,3,s)}, u_{(x,4,s)}\Big] \tag{4.5}$$

Therefore equation (4.4) can be expressed as

$$\begin{aligned} H\Big[u_{(x,0)}, u_{(x,1)}, u_{(x,2)}, u_{(x,3)}, u_{(x,4)}\Big] \approx{}& H\Big[u_{(x,0,s)}, u_{(x,1,s)}, u_{(x,2,s)}, u_{(x,3,s)}, u_{(x,4,s)}\Big] \\ &+ \sum_{k=0}^{4} \Omega_{(k,s)}\Big[u_{(x,0,s)}, u_{(x,1,s)}, u_{(x,2,s)}, u_{(x,3,s)}, u_{(x,4,s)}\Big] u_{(x,k,s+1)} \\ &- \sum_{k=0}^{4} \Omega_{(k,s)}\Big[u_{(x,0,s)}, u_{(x,1,s)}, u_{(x,2,s)}, u_{(x,3,s)}, u_{(x,4,s)}\Big] u_{(x,k,s)} \end{aligned} \tag{4.6}$$

Let

$$R_s\Big[u_{(x,0,s)},u_{(x,1,s)},u_{(x,2,s)},u_{(x,3,s)},u_{(x,4,s)}\Big] = \sum_{k=0}^{4}\Omega_{(k,s)}\Big[u_{(x,0,s)},u_{(x,1,s)},u_{(x,2,s)},u_{(x,3,s)},u_{(x,4,s)}\Big]u_{(x,k,s)} - H\Big[u_{(x,0,s)},u_{(x,1,s)},u_{(x,2,s)},u_{(x,3,s)},u_{(x,4,s)}\Big] \tag{4.7}$$

Therefore, equation (4.5) can be expressed as

$$H\Big[u_{(x,0)},u_{(x,1)},u_{(x,2)},u_{(x,3)},u_{(x,4)}\Big] \approx \sum_{k=0}^{4}\Omega_{(k,s)}\Big[u_{(x,0,s)},u_{(x,1,s)},u_{(x,2,s)},u_{(x,3,s)},u_{(x,4,s)}\Big]u_{(x,k,s+1)} - R_s\Big[u_{(x,0,s)},u_{(x,1,s)},u_{(x,2,s)},u_{(x,3,s)},u_{(x,4,s)}\Big] \tag{4.8}$$

Substituting Eq. (4.8) into Eq. (4.3), we get

$$u_{(t,1,s+1)} + \sum_{k=0}^{4}\Omega_{k,s}u_{(x,k,s+1)} = R_s \tag{4.9}$$

4.2.2 Multi-Domain Approach

Before solving the linearized form of the KS equation, we first decompose the interval of integration $0 \le t \le T$ into non-overlapping intervals defined as

$$\omega_l = [t_{l-1}, t_l],\ \ l = 1,2,3,...,F$$

where

$$0 = t_0 \le t_1 \le t_2 \le \ldots \le t_F = T$$

The objective of the multi-domain approach is to solve for u in Eq. (4.9) in each of the sub-intervals. The initial condition is used for obtaining the solution in the first sub-interval $[t_0,t_1]$. After that, we use the continuity condition between neighbouring sub-intervals to obtain the initial conditions for the subsequent intervals. Basically, the solution at the last point of each interval becomes the initial condition for the next interval. The process where the solutions in different intervals are matched along their common boundary is called patching. The patching condition requires that

$$u^{(l)}(x,t_{l-1}) = u^{(l-1)}(x,t_{l-1}),\ x \in [a,b] \tag{4.10}$$

where $u^{(l)}(x,t)$ denotes the solution of Eq.(4.9) at each sub-interval ω_l with $1 \leq l \leq P$. The given physical regions, $t^l \in [t_{l-1}, t_l]$, is converted to the region $\tau \in [-1,1]$ using the linear transformation

$$t = \frac{1}{2}(t_l - t_{l-1})\tau + \frac{1}{2}(t_l + t_{l-1})$$

and $x \in [a,b]$ is converted to the region $\chi \in [-1,1]$ using the linear transformation

$$x = \frac{1}{2}(b-a)\chi + \frac{1}{2}(b+a).$$

Therefore,in each sub-interval, we are required to solve

$$u^l_{(\tau,1,s+1)} + \sum_{k=0}^{4} \Omega^l_{k,s} u^l_{(\chi,k,s+1)} = R^l_s \tag{4.11}$$

subject to

$$u^{(l)}(\chi, t_{l-1}) = u^{(l-1)}(\chi, \tau_{l-1}),\ \chi \in [a,b] \tag{4.12}$$

4.2.3 Application to the generalized Kuramoto-Sivashinsky equation

The linearized version of the GKS equation is

$$u^l_{(\tau,1,s+1)} + \sum_{k=0}^{4} \Omega^l_{k,s} u^l_{(\chi,k,s+1)} = R^l_s \tag{4.13}$$

with

$$\begin{aligned}
\Omega_{0,s} &= u_{(\chi,1,s)}, \\
\Omega_{1,s} &= u_{(\chi,0,s)}, \\
\Omega_{2,s} &= \alpha, \\
\Omega_{3,s} &= \beta, \\
\Omega_{4,s} &= \gamma, \\
R_s &= u_{(\chi,0,s)} u_{(\chi,1,s)}
\end{aligned}$$

From Eqs. (1.54) and (1.55), Eq. (4.13) at the collocation points (x_i, t_j) becomes

$$\sum_{q=0}^{N_t} \hat{d}_{jq} U^{(l)}_{s+1,q} + \sum_{k=0}^{4} \Omega^{(l)}_{k,s} \hat{\mathbf{D}}^k U^{(l)}_{s+1,j} = R^{(l)}_s \tag{4.14}$$

for $j = 0,1,2,\ldots,N_t$, where $\hat{\mathbf{D}} = \frac{2}{b-a}, \hat{d_{jq}} = \frac{2}{T}$ and

$$\Omega_{k,r} = \begin{bmatrix} \Omega_{k,s}(\chi_0,\tau_j) & & & \\ & \Omega_{k,s}(\chi_1,\tau_j) & & \\ & & \ddots & \\ & & & \Omega_{k,s}(\chi_{N_x},\tau_j) \end{bmatrix} \tag{4.15}$$

Since the initial condition is known, then we express Eq.(4.14) as

$$\sum_{q=0}^{N_t-1} \hat{d}_{jq} U_{s+1,q}^{(l)} + \sum_{k=0}^{4} \Omega_{k,s}^{(l)} \hat{\mathbf{D}}^k U_{s+1,j}^{(l)} = R_j^{(l)} \tag{4.16}$$

where

$$R_j^{(l)} = R_s^{(l)} - \hat{d}_{jN_t} U_{N_t}^{(l)}, \quad \text{for } j = 0,1,2,\ldots,N_t-1$$

Eq.(4.16) can be expressed as the following $M(N+1) \times M(N+1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,3} & \ldots & A_{0,M-1} \\ A_{1,0} & A_{1,1} & A_{1,2} & \ldots & A_{1,M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{M-1,1} & A_{M-1,2} & \ldots & A_{M-1,M-1} \end{bmatrix} \begin{bmatrix} U_0^{(l)} \\ U_1^{(l)} \\ \vdots \\ U_{M-1}^{(l)} \end{bmatrix} = \begin{bmatrix} R_0^{(l)} \\ R_1^{(l)} \\ \vdots \\ R_{M-1}^{(l)} \end{bmatrix} \tag{4.17}$$

where

$$A_{i,i} = \sum_{k=0}^{4} \Omega_{k,s} \hat{\mathbf{D}}^k + d_{i,i} I,$$
$$A_{i,j} = \hat{d}_{i,j} I, \quad \text{when } i \neq j$$

and $\mathbf{I}$ is the identity matrix of size $(N+1) \times (N+1)$. Solving Eq.(4.16) gives $u^l(\chi_i,\tau_j)$ in each interval l.

4.3 Numerical experiments

In this section, we give computational results for examples of the KS equation solved using the MD-BSQLM. In order to determine the level of accuracy of the MD-BSQLM approximate solution, at a particular time level, in comparison with the exact solution, we report maximum error which is defined by

$$E_N = \max_r \{|u^{(l)}(x_r,t_j) - \hat{u}^{(l)}(x_r,t_j)|, : 0 \le r \le N\}, \tag{4.18}$$

where $\hat{u}^l(x_r,t_j)$ is the approximate solution at each sub-interval and $u^l(x_r,t_j)$ is the exact solution at the time level t.

We varied the number of sub-domains, P, between $P = 1, 10$ and 20. $P = 1$ implies that the domain of the problem is not decomposed. This is included to compare with the multi-domain approach and see the effect of the domain decomposition on the accuracy of the MD-BSQLM for $t \ge 1$. Setting $P = 1$(single domain) show better accuracy as for $t \le 1$ and loses accuracy as time domain increase. Increasing the number of sub-domains(P) improves the accuracy of the numerical solution as time t increases. The results are displayed in tables and graphs as shown in the examples below.

Example 1:

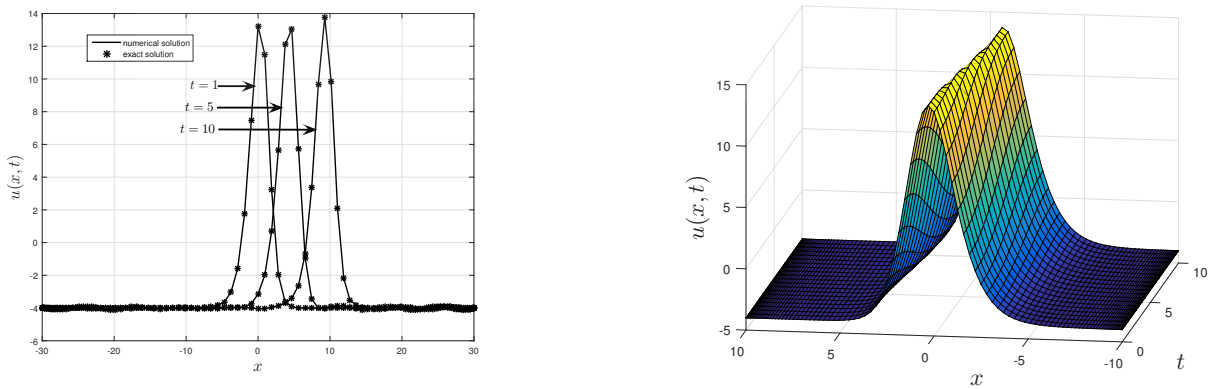
In this example, we consider the GKS equation, with $\alpha = \gamma = 1$ and $\beta = 4$.

$$u_t + uu_x + u_{xx} + 4u_{xxx} + u_{xxxx} = 0, \quad t > 0 \tag{4.19}$$

The exact solution is [45]

$$u(x,t) = 11 + 15\tanh(\theta) - 15\tanh^2(\theta)15\tanh^3(\theta)$$

with $\theta = -\frac{1}{2}x + t$ the solution was evaluated at $t = 0$, as the initial condition, and the boundary functions from the exact solution on the interval $[-10, 10]$.



(a) (b)

Fig. 4.1 The physical behaviour of Example 4.3 in (a) two-dimensions and (b) three-dimensions for $t \le 10$

Table 4.1 Results of MD-BSQLM method for Example 4.3.

Time	t = 1	t = 5	t = 10
P = 1	2.360e-007	2.807e+002	3.340e+003
P = 10	1.3229e-007	2.3576e-007	2.7376e-007
P = 20	1.7568e-007	6.7083e-007	2.7590e-007

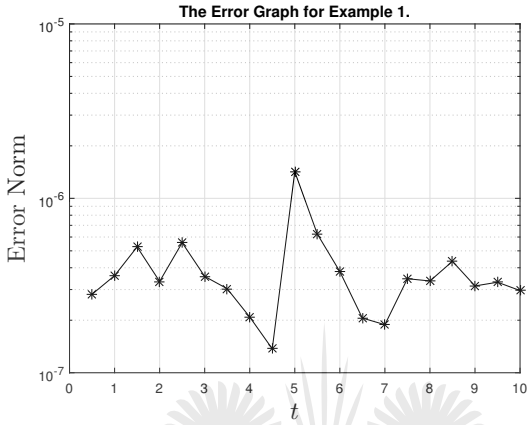


Fig. 4.2 Error for Example 4.3 at different values of t

Figure 4.1 shows the graphical representation of the results of Example 4.3 in 2 dimensions and 3 dimensions. In Figure 4.1a a comparison between the numerical (MD-BSQLM) and the exact solution for $t = 1,5$ and 10 is displayed. A good agreement between the two solutions is observed. Table 4.1 gives maximum errors for different values of P (P is the number of sub-domains). We observe that the single domain is only accurate for small values of t. As t is increased it significantly loses accuracy. The multi-domain, $P = 10$ and 20, is able to retain accuracy as t is increased. The error at each time level is shown by Figure 4.2.

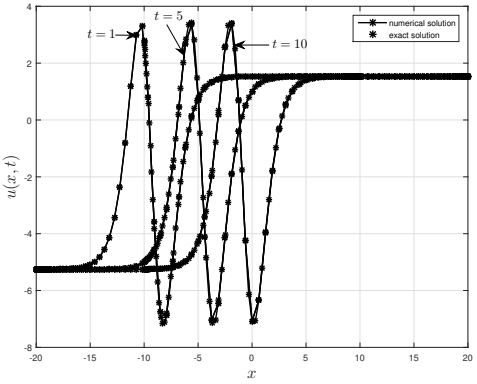
Example 2: In this example, we consider the GKS equation, with $\alpha = 2,\ \gamma = 1$ and $\beta = 0$.

$$u_t + uu_x + 2u_{xx} + u_{xxxx} = 0, \quad t > 0 \tag{4.20}$$

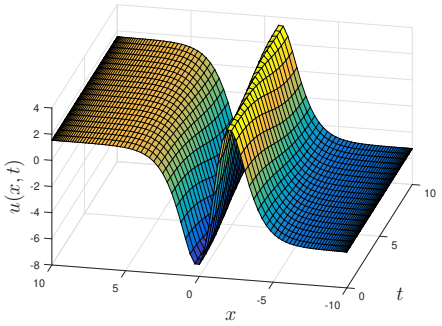
The exact solution is [45]

$$u(x,t) = -\frac{1}{K} + \frac{60}{19}K(-38\gamma K^2 + \alpha)\tanh(\theta) + 120\gamma K^3 \tanh^3(\theta).$$

where $\theta = Kx + t$ and $K = (1/2)\sqrt{11\alpha/19\gamma}$. Similar to the previous example, we extract the required boundary functions from the exact solution on the interval $[-10, 10]$.



(a)



(b)

Fig. 4.3 The physical behaviour of Example 4.3 in (a) two-dimensions and (b) three-dimensions for $t \leq 10$

Table 4.2 Results of the MD-BSQLM for Example 4.3

Time	t = 1	t = 5	t = 10
P = 1	2.741e-008	5.334e+005	3.093e+006
P = 10	2.0323e-007	3.7119e-005	1.2489e-004
P = 20	1.3926e-007	3.7000e-005	1.2396e-004

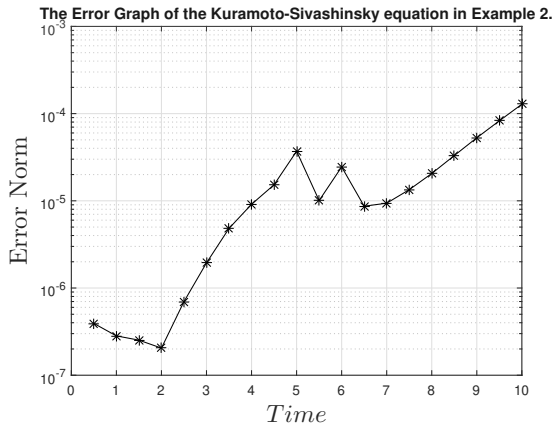


Fig. 4.4 Error for Example 4.3 at different values of t

The comparison between the exact and the MD-BSQLM results for Example 4.3 is depicted by Figure 4.3a for $t = [1,5,10]$. The two results are comparable. Figure 4.3b shows the 3 dimensional solution obtained by the MD-BSQLM. The results for single domain is only accurate for small values of t. As t is increased the method loses accuracy. The multi-domain approach also loses accuracy a little bit in this example but still provides good enough results with a large t. This observation can be seen in Table 4.2 which shows the number of subdomains, P, being varied between $1, 10$ and 20. Figure 4.4 shows the error at each time level.

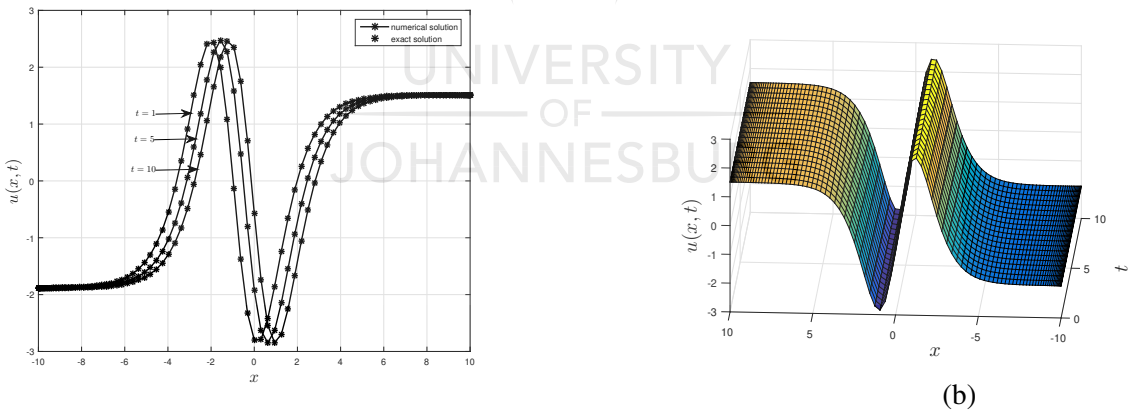
Example 3: In this example, we consider the GKS equation, with $\alpha = 1,\ \gamma = 0.5$ and $\beta = 0$.

$$u_t + uu_x + u_{xx} + 0.5u_{xxxx} = 0, \quad t > 0 \tag{4.21}$$

The exact solution is [45]

$$u(x,t) = -\frac{0.1}{K} + \frac{60}{19}K(-38\gamma K^2 + \alpha)\tanh(\theta) + 120\gamma K^3 \tanh^3(\theta),$$

where $\theta = Kx + 0.1t$ and $K = (1/2)\sqrt{11\alpha/19\gamma}$. Again, we extract the required boundary functions from the exact solution on the interval $[-10, 10]$.



(a) (b)

Fig. 4.5 The physical behaviour of Example 4.3 in (a) two-dimensions and (b) three-dimensions for $t \leq 10$

The results for Example 4.3 are shown in Table 4.3 and Figures 4.5 and 4.6. The same trend as with the previous examples is observed. The error at each time level is displayed in Figure 4.6.

Table 4.3 Results of the MD-BSQLM for Example refE3r.

Time	t = 1	t = 5	t = 10	t = 20
P = 1	9.397e-008	1.160e-008	4.908e-008	4.913e+000
P = 10	4.2401e-008	3.1848e-008	3.8943e-008	3.0538e-007
P = 20	1.0491e-007	4.6799e-008	4.5888e-008	3.3325e-007

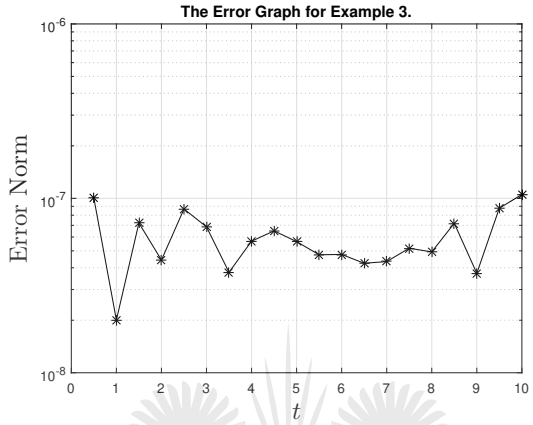


Fig. 4.6 Error for Example 4.3 at different values of t

Example 4: In this example, we consider the GKS equation, with $\alpha = \gamma = 1$ and $\beta = 4$.

$$u_t + uu_x + u_{xx} + 4u_{xxx} + u_{xxxx} = 0, \quad t > 0 \tag{4.22}$$

The exact solution is [45]

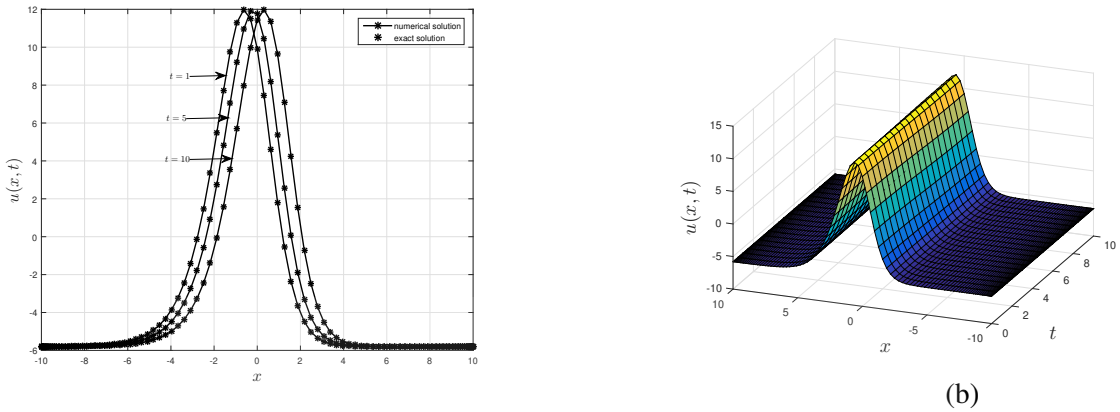
$$u(x,t) = 9 + 2c + 15\tanh(\theta) - 15\tanh^2(\theta) - 15\tanh^3(\theta)$$

with $\theta = -\frac{1}{2}x + ct$. Similar to the previous examples, we extract the required boundary functions from the exact solution on the interval $[-10, 10]$.

Table 4.4 Results the MD-BSQLM for Example 4.3.

Time	t = 1	t = 5	t = 10	t=20
P = 1	3.658e-007	1.971e-006	2.936e-006	1.341e+001
P = 10	1.6078e-007	6.6995e-007	7.3393e-007	2.9149e-006
P = 20	1.3819e-007	4.7150e-007	2.7603e-006	4.7322e-007

Again for Example 4.3 the results are similar to the previous results, as shown by Table 4.4 and Figures 4.7 and 4.8. The error at each time level is displayed in Figure 4.8.



(a) (b)

Fig. 4.7 The physical behaviour of Example 4.3 in (a) two-dimensions and (b) three-dimensions for $t \leq 10$

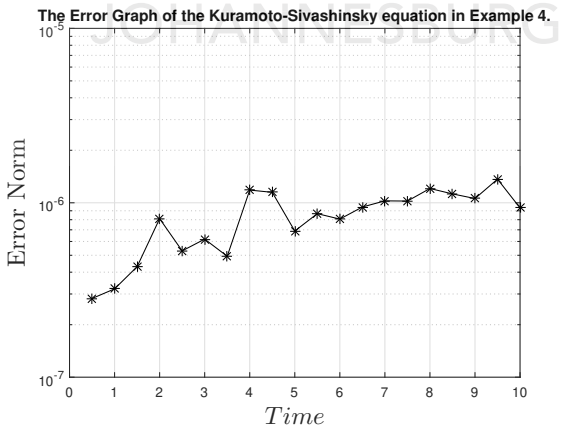


Fig. 4.8 Error for Example 4.3 at different values of t

Example 5: In this example, we consider the KS equation

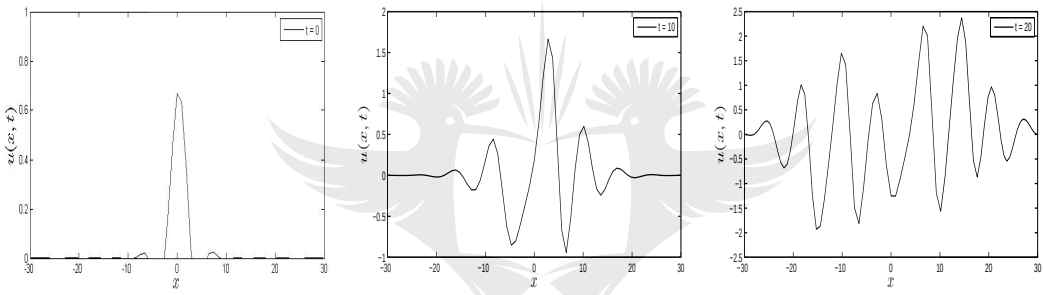
$$u_t + uu_x + u_{xx} + u_{xxxx} = 0, \tag{4.23}$$

where is the simplest nonlinear partial differential exhibiting the chaotic behavior over a finite spatial domain. To find the approximate solution of Eq.(4.23) we use the Gaussian initial condition

$$u(x,0) = e^{-x^2}$$

with the boundary conditions

$$u(a,t) = 0,\ u(b,t) = 0,\ u_x(a,t) = 0,\ u_x(b,t) = 0.$$



(a) $t = 0$ (b) $t = 10$ (c) $t = 20$

Fig. 4.9 The solitary wave propagation of the Kuramoto-Sivashinsky equation at $t = 0$, $t = 10$, and $t = 20$

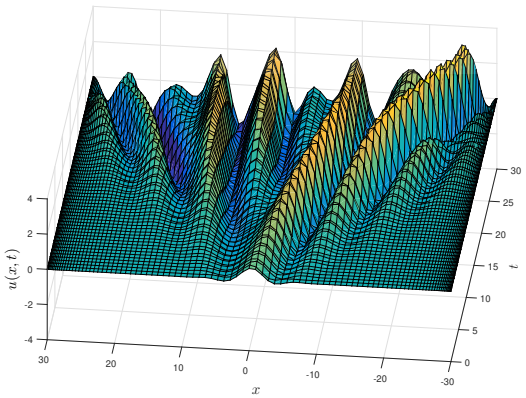


Fig. 4.10 The physical behaviour of Example 4.3 in (a) two-dimensions and (b) three-dimensions for $t \leq 10$

Figure 4.9 shows the chaotic nature of the solution of Example 4.3. It can be seen that as the value of t is increased the chaotic behaviour becomes extreme, which is consistent with the results of [78]. Figure 4.10 shows the 3 dimensional plot.

4.4 Conclusion

In this work, we have successfully used the multi-domain bivariate spectral quasi-linearization method to solve a chaotic partial differential equation called the Kuramoto-Sivashinsky equation. The method is based on a domain decomposition in time, a bivariate Lagrange interpolation and a quasi-linearization technique. The results show that the multi-domain approach is effective in handling chaotic PDEs with large time domains.

Chapter 5

Conclusion

5.1 Summary of the work

In this dissertation, we proposed a new numerical technique for solving nonlinear differential equations. The new numerical technique is called A multi-domain compact finite difference relaxation method (MD-CFDRM). We then implemented the multi-domain technique into the sixth-order compact finite difference quasi-linearization method by extending the work done by Dlamini [25] and we call the method the Multi-domain compact finite difference quasi-linearization method (MD-CFDQLM). We also extend the application pseudospectral methods that use spectral collocation independently in space and time to solve the famous nonlinear evolution partial differential equations that exhibit chaotic behaviour under certain conditions and the method is referred to as Multi-domain bivariate spectral quasi-linearization method (MD-BSQLM). We tested the applicability of these methods to different nonlinear differential equations and compared their results to results to those found in literature.

In Chapter **2**, the multi-domain compact finite difference relaxation method was then applied to nonlinear systems of initial value problems that exhibit chaotic behaviour (chaotic and hyperchaotic systems). Chaotic and hyperchaotic systems are characterized by high sensitivity to small perturbation on initial data and rapidly changing solutions. Such rapid variations in the solution pose tremendous problems to a number of numerical approximations. This often requires using a large number of grid points. It was shown that the MD-CFDRM is more accurate than some traditional numerical methods and reliable method for solving complex dynamical systems with chaotic and hyperchaotic behavior, computationally efficient and robust.

In Chapter **3**, we have extended and implemented for the first time a new approach named the multi-domain compact finite difference quasi-linearization method (MD-CDFQLM) for solving nonlinear parabolic PDEs. The objective was to extend the application of higher order finite difference schemes and improve accuracy of the schemes when solving nonlinear evolution partial differential equations over large time interval. This idea was tested on Fisher's, Burgers-Fisher, Burgers-Huxley, and Burgers systems. It was shown that the MD-CFDQLM is more accurate and reliable when solving nonlinear partial differential equations over large time interval.

In Chapter **4**, the multi-domain bivariate spectral quasi-linearization method (MD-BSQLM) was used to solve a chaotic partial differential equation called the Kuramoto-Sivashinsky equation. The MD-BSQLM proves to be effective in handling chaotic PDEs with large time domains.

5.2 Future work

In this work, we have opened up many new ideas and solve various nonlinear differential equations. We have solved differential equations mainly used to model differential equations arising from fluid dynamics, and parabolic nonlinear evolution equations. We could possibly extend the use of multi-domain technique to solve differential equations arising from financial mathematics and other fields. And also extend the work to solve hyperbolic and elliptic differential equations presented in general forms.

References

[1] A K Alomari, M S M Noorani, R. N. (2010). Homotopy approach for the hyperchaotic chen system. *Physica Scripta*, 81(4):045005.

[2] Abdollah, B. and Abazari, R. (2011). Exact solutions for non-linear schrödinger equations by differential transformation method. 35:37–51.

[3] Abdulaziz, O., Mohd Noor, N. F., Hashim, I., and Noorani, M. (2008). Further accuracy tests on adomian decomposition method for chaotic systems. 36:1405–1411.

[4] Akrivis, G. D. (1992). Finite difference discretization of the kuramoto-sivashinsky equation. *Numerische Mathematik*, 63(1):1–11.

[5] Al-Sawalha, M. M., Noorani, M., and Hashim, I. (2009). On accuracy of adomian decomposition method for hyperchaotic rössler system. *Chaos, Solitons and Fractals*, 40(4):1801–1807.

[6] Alomari, A., Noorani, M., and Nazar, R. (2009). Adaptation of homotopy analysis method for the numeric-analytic solution of chen system. *Communications in Nonlinear Science and Numerical Simulation*, 14(5):2336–2346.

[7] Anders, D., Dittmann, M., and Weinberg, K. (2012). A higher-order finite element approach to the kuramoto-sivashinsky equation. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 92:599–607.

[8] BASTANI. M, S. D. P. (2012). A highly accurate method to solve fisher's equation. *J Phys*, 78(4):335.

[9] Batiha, B., Noorani, M. S. M., Hashim, I., and Ismail, E. S. (2007). The multistage variational iteration method for a class of nonlinear system of odes. *Physica Scripta*, 76(4):388.

[10] Bellman, R. E. and Kalaba., R. E. (1965). Quasilinearization and nonlinear boundary value problems. *Cambridge University Press, Elsevier, New York*, 52(380):212–212.

[11] Borhanifar, A. and Abazari, R. (2010). Numerical study of nonlinear schrödinger and coupled schrödinger equations by differential transformation method. *Optics Communications*, 283(10):2026 – 2031.

[12] Canuto, C.; Hussaini, M. Y. Q. A. Z. T. A. (1987). Spectral methods in fluid dynamics. berlin. *Applied Mathematics and Mechanics*, 69(7):214–214.

[13] Chen, G. and Dong, X. (1993). From chaos to order – perspectives and methodologies in controlling chaotic nonlinear dynamical systems. *Int. J. Bifurcation Chaos Appl. Sci. Eng.*, 3(6):1363–1409.

[14] Cheng, H., Zhou, J., and Wu, Q. (2011). Adaptive synchronization of coupled hyperchaotic chua systems. In *2011 Chinese Control and Decision Conference (CCDC)*, pages 143–148.

[15] Chowdhury, M. and Hashim, I. (2009). Application of multistage homotopy-perturbation method for the solutions of the chen system. *Nonlinear Analysis: Real World Applications*, 10(1):381 – 391.

[16] Chowdhury, M., Hashim, I., and Momani, S. (2009). The multistage homotopy-perturbation method: A powerful scheme for handling the lorenz system. *Chaos, Solitons and Fractals*, 40(4):1929 – 1937.

[17] Chowdhury, M. S. H., Hashim, I., Momani, S., and Rahman, M. M. (2012). Application of multistage homotopy perturbation method to the chaotic Genesio system. *Abstr. Appl. Anal.*, 2012:10.

[18] Cohen, B., Krommes, J., Tang, W., and Rosenbluth, M. (1976). Non-linear saturation of the dissipative trapped-ion mode by mode coupling. *Nuclear Fusion*, 16(6):971.

[19] D. M. Rademacher, J. and Wittenberg, R. (2006). Viscous shocks in the destabilized kuramoto-sivashinsky equation. *Journal of Computational and Nonlinear Dynamics*, 1.

[20] Danca, M.-F. and Chen, G. (2004). Bifurcation and chaos in a complex model of dissipative medium. *Int. J. Bifurcation Chaos Appl. Sci. Eng.*, 14(10):3409–3447.

[21] Darvishi, M., Kheybari, S., and Khani, F. (2008). Spectral collocation method and darvishi's preconditionings to solve the generalized burgers-huxley equation. *Communications in Nonlinear Science and Numerical Simulation*, 13(10):2091 – 2103.

[22] Dehghan, M., Saray, B. N., and Lakestani, M. (2012). Three methods based on the interpolation scaling functions and the mixed collocation finite difference schemes for the numerical solution of the nonlinear generalized burgers-huxley equation. *Mathematical and Computer Modelling*, 55(3):1129 – 1142.

[23] Dehghan, M. and Taleei, A. (2010). A compact split-step finite difference method for solving the nonlinear schrödinger equations with constant and variable coefficients. *Computer Physics Communications*, 181(1):43 – 51.

[24] Deng, D. and Pan, T. (2015). A fourth-order singly diagonally implicit runge-kutta method for solving one-dimensional burgers' equation. *IAENG International Journal of Applied Mathematics*, 45:327–333.

[25] Dlamini, P. and Khumalo, M. (2017). A new compact finite difference quasilinearization method for nonlinear evolution partial differential equations. *Open Mathematics*, 15.

[26] Dlamini, P., Motsa, S., and Khumalo, M. (2013a). Higher order compact finite difference schemes for unsteady boundary layer flow problems. vol 2013.

[27] Dlamini, P., Motsa, S., and Khumalo, M. (2013b). On the comparison between compact finite difference and pseudospectral approaches for solving similarity boundary layer problems. 2013.

[28] Do, Y. and Jang, B. (2012). Enhanced multistage differential transform method: Application to the population models. *Abstract and Applied Analysis*, 2012, Special Issue.

[29] Drazin, P. G.; Johnson, R. S. (1989). Solitons: An introduction. *ZAMM - Journal of Applied Mathematics and Mechanics*, 70(8):340–340.

[30] Düring, B., Fournié, M., and Jüngel, A. (2003). High-order compact finite difference schemes for a nonlinear Black-Scholes equation. *Int. J. Theor. Appl. Finance*, 6(7):767–789.

[31] Fan, E. and Zhang, H. (1998). A note on the homogeneous balance method. *Physics Letters A*, 246(5):403 – 406.

[32] Freihat, A. and Momani, S. (2012). Adaptation of differential transform method for the numeric-analytic solution of fractional-order Rössler chaotic and hyperchaotic systems. *Abstr. Appl. Anal.*, 2012:13.

[33] Goh, S., Noorani, M., and Hashim, I. (2009a). Efficacy of variational iteration method for chaotic genesio system - classical and multistage approach. *Chaos, Solitons and Fractals*, 40(5):2152 – 2159.

[34] Goh, S., Noorani, M., Hashim, I., and Al-Sawalha, M. (2009b). Variational iteration method as a reliable treatment for the hyperchaotic rössler system. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):363–371.

[35] Golbabai, A. and Javidi, M. (2009a). A spectral domain decomposition approach for the generalized burger's-fisher equation. *Chaos, Solitons and Fractals*, 39(1):385 – 392.

[36] Golbabai, A. and Javidi, M. (2009b). A spectral domain decomposition approach for the generalized burger's-fisher equation. *Chaos, Solitons and Fractals*, 39(1):385 – 392.

[37] Gonzalez-Parra, G., Arenas, A. J., and Jodar, L. (2009). Piecewise finite series solutions of seasonal diseases models using multistage adomian method. *Communications in Nonlinear Science and Numerical Simulation*, 14(11):3967 – 3977.

[38] Grimshaw, R. and Hooper, A. (1991). The non-existence of a certain class of travelling wave solutions of the kuramoto-sivashinsky equation. *Physica D: Nonlinear Phenomena*, 50(2):231 – 238.

[39] Hariharan, G., Kannan, K., and Sharma, K. (2009). Haar wavelet method for solving fisher's equation. *Applied Mathematics and Computation*, 211(2):284 – 292.

[40] Hooper, A. P. and Grimshaw, R. (1985). Nonlinear instability at the interface between two viscous fluids. *The Physics of Fluids*, 28(1):37–45.

[41] Huibin, L. and Kelin, W. (1990). Exact solutions for two nonlinear equations. i. *Journal of Physics A: Mathematical and General*, 23(17):3923.

[42] Hyman, J. M. and Nicolaenko, B. (1986). The kuramoto-sivashinsky equation: A bridge between pde's and dynamical systems. *Physica D: Nonlinear Phenomena*, 18(1):113 – 126.

[43] Javidi, M. (2006). Spectral collocation method for the solution of the generalized burger-fisher equation. *Applied Mathematics and Computation*, 174(1):345 – 352.

[44] Kalita, J. C., Dalal, D. C., and Dass, A. K. (2002). A class of higher order compact schemes for the unsteady two-dimensional convection-diffusion equation with variable convection coefficients. *Int. J. Numer. Methods Fluids*, 38(12):1111–1131.

[45] Khater, A. and Temsah, R. (2008). Numerical solutions of the generalized kuramoto-sivashinsky equation by chebyshev spectral collocation methods. *Computers and Mathematics with Applications*, 56(6):1465 – 1472.

[46] Kocic LM, Gegovska-Zajkova S, K. S. (2010). *On Chua dynamical system*. Ser A. Appl. Math. Inform. and Mech.

[47] Kuramoto, Y. and Tsuzuki, T. (1976). Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress of Theoretical Physics*, 55(2):356–369.

[48] Kurulay, M., Secer, A., and Ali Akinlar, M. (2013). A new approximate analytical solution of kuramoto - sivashinsky equation using homotopy analysis method. 7.

[49] Lai, H. and Ma, C. (2009). Lattice boltzmann method for the generalized kuramoto-sivashinsky equation. *Physica A: Statistical Mechanics and its Applications*, 388(8):1405 – 1412.

[50] Lakestani, M. and Dehghan, M. (2012). Numerical solutions of the generalized kuramoto-sivashinsky equation using b-spline functions. *Applied Mathematical Modelling*, 36(2):605 – 617.

[51] Lele, S. K. (1992). Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16 – 42.

[52] Li, J. and Visbal, M. R. (2006). High-order compact schemes for nonlinear dispersive waves. *J. Sci. Comput.*, 26(1):1–23.

[53] Liao, W. (2008). An implicit fourth-order compact finite difference scheme for one-dimensional burgers' equation. *Applied Mathematics and Computation*, 206(2):755 – 764.

[54] Liu, X. (1991). Gevrey class regularity and approximate inertial manifolds for the kuramoto-sivashinsky equation. *Physica D: Nonlinear Phenomena*, 50(1):135 – 151.

[55] Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141.

[56] Luo, X., Small, M., Danca, M.-F., and Chen, G. (2007). On a dynamical system with multiple chaotic attractors. *Int. J. Bifurcation Chaos Appl. Sci. Eng.*, 17(9):3235–3251.

[57] Magagula, V., Motsa, S., and Sibanda, P. (2016). A multi-domain bivariate pseudospectral method for evolution equations. page 1750041.

[58] Michelson, D. (1986). Steady solutions of the kuramoto-sivashinsky equation. *Physica D: Nonlinear Phenomena*, 19(1):89 – 111.

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

[60] Moghimi, M. and Hejazi, F. S. (2007). Variational iteration method for solving generalized burger-fisher and burger equations. *Chaos, Solitons and Fractals*, 33(5):1756 – 1761.

[61] Motsa, S., Dlamini, P., and Khumalo, M. (2012). Solving hyperchaotic systems using the spectral relaxation method. *Abstract and Applied Analysis*, vol 2012.

[62] Motsa, S. S., Dlamini, P., and Khumalo, M. (2013). A new multistage spectral relaxation method for solving chaotic initial value systems. *Nonlinear Dynamics*, 72(1):265–283.

[63] Odibat, Z., Bertelle, C., Aziz-Alaoui, M., and Duchamp, G. (2010). A multi-step differential transform method and application to non-chaotic or chaotic systems. *Computers and Mathematics with Applications*, 59:1462–1472.

[64] Olmos, D. and Shizgal, B. D. (2006). A pseudospectral method of solution of fisher's equation. *Journal of Computational and Applied Mathematics*, 193(1):219 – 242.

[65] Orszag, S. A. (1980). Spectral methods for problems in complex geometries. *Journal of Computational Physics*, 37(1):70 – 92.

[66] Papageorgiou, D. T., Maldarelli, C., and Rumschitzki, D. S. (1990). Nonlinear interfacial stability of core-annular film flows. *Physics of Fluids A: Fluid Dynamics*, 2(3):340–352.

[67] Rabinovich MI, F. A. (1979). Stochastic self-modulation of waves in nonequlibrium media. *Physica D: Nonlinear Phenomena*, 77:617–629.

[68] Rech, P. C. and Albuquerque, H. A. (2009). A hyperchaotic Chua system. *Int. J. Bifurcation Chaos Appl. Sci. Eng.*, 19(11):3823–3828.

[69] Rikitake, T. (1958). Oscillations of a system of disk dynamos. *Mathematical Proceedings of the Cambridge Philosophical Society*, 54:89 – 105.

[70] Rössler, O. (1976). An equation for continuous chaos. *Physics Letters A*, 57(5):397 – 398.

[71] Ryabenkii, S. G. V. (1987). *Difference schemes: An introduction to the underlying theory*. Elsevier Science Publishers, New York.

[72] Sari, M. (2009). Solution of the porous media equation by a compact finite difference method. *Math. Probl. Eng.*, 2009:13.

[73] Sari, M. and Gürarslan, G. (2009). A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation. *Appl. Math. Comput.*, 208(2):475–483.

[74] Sari, M. and Gürarslan, G. (2011). A sixth-order compact finite difference method for the one-dimensional sine-Gordon equation. *Int. J. Numer. Methods Biomed. Eng.*, 27(7):1126–1138.

[75] Shah, A., Yuan, L., and Khan, A. (2010). Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier-Stokes equations. *Appl. Math. Comput.*, 215(9):3201–3213.

[76] Sivashinsky, G. I. (1983). Instabilities, pattern formation, and turbulence in flames. *Annual Review of Fluid Mechanics*, 15(1):179–199.

[77] Trefethen, L. N. (2000). *Spectral Methods in MatLab*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[78] Uddin, M., Haq, S., and ul Islam, S. (2009). A mesh-free numerical method for solution of the family of kuramoto-sivashinsky equations. *Applied Mathematics and Computation*, 212(2):458 – 469.

[79] Wang, S. and Yu, Y. (2012). Application of multistage homotopy-perturbation method for the solutions of the chaotic fractional order systems. *Int. J. Nonlinear Sci.*, 13(1):3–14.

[80] Wazwaz, A.-M. and Gorguis, A. (2004). An analytic study of fisher's equation by using adomian decomposition method. *Applied Mathematics and Computation*, 154(3):609 – 620.

[81] Xiang-Jun, Wu; Jing-Sen, L. G.-R. (2008). Chen Chaos synchronization of Rikitake chaotic attractor using the passive control technique. *Nonlinear Dynam*, 53(1-2):45–53.

[82] Xie, S.-S., Li, G.-X., and Yi, S. (2009). Compact finite difference schemes with high accuracy for one-dimensional nonlinear schrödinger equation. *Computer Methods in Applied Mechanics and Engineering*, 198(9):1052 – 1060.

[83] Xu, Y. and Shu, C.-W. (2006). Local discontinuous galerkin methods for the kuramoto-sivashinsky equations and the ito-type coupled kdv equations. *Computer Methods in Applied Mechanics and Engineering*, 195(25):3430 – 3447. Discontinuous Galerkin Methods.

[84] Zhang, P.-G. and Wang, J.-P. (2012). A predictor-corrector compact finite difference scheme for Burgers' equation. *Appl. Math. Comput.*, 219(3):892–898.