THE NUMERICAL SOLUTIONS OF SOME LINEAR AND NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS USING TRIGONOMETRIC B-SPLINE BASIS FUNCTIONS

A

Thesis

Submitted to

For the award of

DOCTOR OF PHILOSOPHY (Ph.D.)

in

MATHEMATICS

by

VARUN JOSHI

(Registration No.: 41400073)

Under the Supervision of

Dr. Geeta Arora

Lovely Faculty of Technology and Sciences

Lovely Professional University

Punjab

May 2018

Declaration of Authorship

I, Varun Joshi, declare that this thesis titled, 'The Numerical Solutions of Some Linear and Nonlinear Partial Differential Equations Using Trigonometric B-Spline Basis Functions' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Certificate

The thesis titled **"The Numerical Solutions of Some Linear and Nonlinear Partial Differential Equations Using Trigonometric B-Spline Basis Functions"** submitted by **Mr. Varun Joshi** for the award of the degree of Doctor of Philosophy has been carried out under my supervision at the Department of Mathematics, Lovely Professional University, Punjab, India. The matter presented in this thesis is original and has not been submitted in any other University or Institute for award of any degree or diploma. The work is comprehensive, complete and fit for evaluation.

Dr. Geeta Arora
Assistant Professor,
Department of Mathematics,
Lovely Professional University,
Punjab, India.

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

Abstract

Lovely Faculty of Technology and Sciences
Department of Mathematics

Doctor of Philosophy

by VARUN JOSHI

This thesis aims to validate and establish the trigonometric B-spline basis function to compute the numerical solution of linear and nonlinear partial differential equations. These differential equations are the powerful tool to express various phenomena in science, for instance, quantum mechanics, electromagnetic fields, fluid flow diffusion, and the physical laws of structural mechanics. Because of having many applications in different disciplines, differential equations have been solved in the literature using various analytical and numerical methods. During the mathematical modeling of the problems, most of the time, the modeled differential equation is not so easy to solve. Hence to obtain an analytical or exact solution for that problem is a challenging task. This results in the requirement of advanced numerical methods which can be used to get an accurate numerical solution. So in this research work, two hybrid numerical schemes are proposed with trigonometric B-spline basis function to compute the numerical solution of time dependent partial differential equations. Various linear and nonlinear partial differential equations are considered for the numerical solution. To solve the second order equations like Fisher's equation, sine-Gordon equation, Burgers' equation, Schrödinger equation, and telegraph equation, basis function of the third degree is used with collocation and differential quadrature methods. For third and fourth order equations like Korteweg-de Vries, extended Fisher-Kolmogorov, and Kuramoto-Sivashinsky equation, fifth degree basis function is used with differential quadrature method in one as well as in two dimensions. To demonstrate the applicability and robustness of the proposed schemes different error norms like L_2, L_∞, RMS and GRE are computed and the results are presented in this work. The stability of the proposed schemes is also discussed using eigenvalues.

This thesis contains seven chapters. Chapter 1 is introductory in nature, in this brief introduction and literature survey are given for trigonometric B-spline basis functions, collocation, and differential quadrature method. Furthermore, some prerequisites like stability conditions for the system and different error norm formulas are also given in this chapter.

In Chapter 2, Fisher's equation in one dimension is solved with trigonometric B-spline collocation method, which is in the following form:

$$\frac{\partial u(x,t)}{\partial t} = \nu \frac{\partial^2 u(x,t)}{\partial x^2} + \rho f(u(x,t)),\ x \in (-\infty, \infty),\ t > 0 \tag{1}$$

where t and x are the time and spatial coordinate respectively. $\nu > 0$ represents the diffusion coefficient, $\rho > 0$ is the reaction factor and $f(u(x,t))$ is the nonlinear reaction term. $f(u(x,t))$ represents a nonlinear development rate and diminishes as u expands, and vanishes when $u = 1$. It compares the development of a populace u when there is a point of confinement $u = 1$ on the population's span the habitat can bolster. In the event, if $u > 1$ then $f(u) < 0$, i.e. the population diminishes when u is more noteworthy than limiting value. This interpretation suggests that the habitat can support a certain maximum population so that

$$0 \leq u(x,0) \leq 1 \ for\ x \in \mathbb{R} \tag{2}$$

Due to the existence of Fisher's equation in many contexts, the nonlinear term $f(u)$ is referred to different terminology based on its role. The term $f(u)$ alludes to a source or reaction term that articulate to the confinement-fatality process in a biological background. And if a chemically responding substance is diffusing through a medium, then its focus $u(x,t)$ fulfills mathematical equation (2), where $f(u)$ represent the rate of expansion of the substance because of the synthetic response. Different issues depicted by equation (1) incorporate the spread of creature or plant populace and the advancement of neutron populace in an atomic reactor, where $f(u)$ articulates to the net development rate. A part of this chapter is published in Ain Shams Engineering Journal.

In Chapter 3, numerical solution of second order partial differential equations in one dimension is obtained by using the cubic trigonometric differential quadrature

method. In this chapter, four partial differential equations: sine-Gordon, Burgers', Schrödinger, and telegraph equation are solved. These equations have many applications in numerous fields of science and engineering. For instance, it appears in the application of nonlinear optics, the stability of fluid motion, motion of pendulum attached to a stretched string, investigation of turbulence portrayed by the connection of the two inverse impacts of convection and dispersion etc. A part of this research work is published in the two different journals namely Alexandria Engineering Journal and Indian Journal of Science and Technology.

Chapter 4 is an extension of chapter three in the form of space dimensions. In this chapter, all the four equations are solved for the numerical solution in two dimensions. A part of this research work is published in Alexandria Engineering Journal.

Chapter 5 concentrated on the numerical solution of the generalized nonlinear fourth order partial differential equation:

$$u_t + \epsilon u u_x + \alpha u_{xx} + \mu u_{xxx} + \beta u_{xxxx} + f(u) = 0. \tag{3}$$

This type of nonlinear partial differential equation arises in many applications of theoretical, engineering and environmental sciences. Because of the involvement of nonlinearity and high order of derivatives, it is very difficult to find the exact solution of this equation, hence solving this equation numerically is a good option. This general fourth order partial differential equation leads to some important equations such as Korteweg-de Vries, extended Fisher-Kolmogorov, and Kuramoto-Sivashinsky equation which contain the nonlinear terms with higher order derivatives. Due to the presence of high order derivatives in these equations, cubic trigonometric B-spline basis function is not sufficient for the computation, hence in this chapter quintic trigonometric B-spline differential quadrature method is used to solve these equations.
In Chapter 6 considered equation for numerical solution is the extended Fisher-Kolmogorov partial differential equation in two dimensions:

$$\begin{aligned} u_t + \gamma(u_{xxxx} + u_{yyyy}) - (u_{xx} + u_{yy}) + f(u) &= g(x, y, t), \\ \text{or} \qquad\qquad\qquad\qquad\qquad\qquad & \\ u_t + \gamma\Delta^2 u - \Delta u + f(u) &= g(x, y, t), \end{aligned} \tag{4}$$

with initial and boundary conditions

$$u(x, y, t) = u_0(x, y),\ (x, y) \in [a, b] \times [c, d]$$

$$u = 0, \Delta u = 0, \text{ on boundary}$$

where γ is a positive constant and $f(u) = (u^3 - u)$. To solve this equation quintic trigonometric differential quadrature method is used to compute the numerical solution. This equation has applications in various phenomena such as in pattern formation, spatiotemporal chaos in bi-stable systems and in phase transition near a Lifshitz point.

In Chapter 7, conclusions are drawn based on the present study and future research work in this direction is suggested.

Acknowledgements

First of all, I would like to thank the Almighty for granting perseverance. I would like to express my gratitude to Dr. Geeta Arora, Assistant Professor, Department of Mathematics, Lovely Professional University, Phagwara, for her patient, guidance, and support throughout this work. I was truly very fortunate to have the opportunity to work with her as a student. It was both an honor and a privilege to work with her. She also provides help in technical writing and presentation style and I found this guidance to be extremely valuable. I take this opportunity to express my sincere thanks to Mr. Lavish Kansal, Department of Academic Affairs, Dr. Ramesh Katta, Department of Mathematics, and Prof. Sangha, Content Development Cell, Lovely Professional University, Phagwara, for their valuable support and help without which it would not have been possible for me to complete this work.

I am also thankful to all my teachers, and all my friends who devoted their valuable time and helped me in all possible ways towards successful completion of this work. I do not find enough words with which I can express my feeling of thanks to the entire faculty, and staff of Department of Mathematics, Lovely Professional University, Phagwara, for their help, inspiration and moral support which went a long way in the successful completion of my work. I thank all those who have contributed directly or indirectly to this work.

Lastly, and more importantly, I would like to thank my family for their years of unyielding love and encouragement. They have always wanted the best for me and I admire my parent's and wife's determination and sacrifice to put me through Ph.D.

Contents

Declaration of Authorship iii

Certificate v

Abstract vii

Acknowledgements xi

List of Figures xvii

List of Tables xxiii

1 Introduction 1
- 1.1 Introduction . . . 1
- 1.2 Idea of spline . . . 5
  - 1.2.1 Definition of spline . . . 5
  - 1.2.2 B-spline basis function . . . 6
- 1.3 Collocation method . . . 12
- 1.4 Differential quadrature method . . . 13
  - 1.4.1 Computation of the weighting coefficients(a_{ij}) for cubic trigonometric B-spline function . . . 15
  - 1.4.2 Computation of the weighting coefficients(a_{ij}) for quintic trigonometric B-spline function . . . 17
  - 1.4.3 Computation of the weighting coefficients for high order derivatives . . . 18
- 1.5 Strong stability-preserving time-stepping Runge-Kutta (SSP-RK43) scheme . . . 20
- 1.6 Stability of the numerical schemes . . . 21
- 1.7 Error norms and order of convergence . . . 23

2 Numerical solution of Fisher's equation using cubic trigonometric collocation method 25
- 2.1 Introduction . . . 25

2.2 Solution of Fisher's equation 28
2.3 The initial vector C^0 29
2.4 Stability of the numerical scheme 30
2.5 Convergence analysis 31
2.6 Numerical experiment and discussion 34
2.7 Summary 36

3 Numerical solution of second order partial differential equations using cubic trigonometric differential quadrature method 47
3.1 Introduction to sine-Gordon equation 47
3.1.1 Methods proposed to solve sine-Gordon equation type equation 48
3.1.2 Implementation of numerical scheme on sine-Gordon equation 49
3.1.3 Stability of the scheme 50
3.1.4 Numerical Experiment and Discussion 50
3.1.5 Summary 52
3.2 Introduction to Burgers' equation 57
3.2.1 Implementation of the numerical scheme on Burgers' equation 60
3.2.2 Stability of the scheme 61
3.2.3 Numerical results and discussion 61
3.2.4 Summary 63
3.3 Introduction of Schrödinger equation 66
3.3.1 Implementation of Numerical Scheme 71
3.3.1.1 Method 1 71
3.3.1.2 Method 2 72
3.3.2 Stability of the scheme 72
3.3.3 Numerical experiment and discussion 73
3.3.4 Summary 75
3.4 Introduction to telegraph equation 80
3.4.1 Implementation of the numerical scheme on telegraph equation 84
3.4.2 Stability of the scheme 85
3.4.3 Numerical experiments and discussions 86
3.4.4 Summary 88

4 Numerical solution of second order partial differential equations in two dimensions using cubic trigonometric differential quadrature method 97
4.1 Partial differential equations in two dimensions 97
4.1.1 Sine-Gordon equation in two dimensions 98
4.1.2 Burgers' equation in two dimensions 98
4.1.3 Nonlinear Schrödinger equation in two dimensions 99
4.1.4 Linear telegraph equation in two dimensions 99
4.1.5 Implementation of numerical scheme to partial differential equations 100
4.2 Stability of the scheme 101
4.3 Numerical solution and discussion 102

4.4 Summary 106

5 Numerical solution of fourth order partial differential equations using quintic trigonometric differential quadrature method 131
5.1 Introduction 131
5.1.1 Korteweg-de Vries (KdV) equation 132
5.1.2 Kuramoto-Sivashinsky (KS) equation 133
5.1.3 Extended Fisher-Kolmogorov equation 133
5.1.4 Implementation of the numerical scheme 134
5.2 Stability of the scheme 135
5.3 Numerical results and discussion 135
5.4 Summary 140

6 Numerical solution of extended Fisher-Kolmogorov equation in two dimensions using quintic trigonometric differential quadrature method 159
6.1 Introduction 159
6.2 Implementation of the numerical scheme 160
6.3 Stability of the scheme 160
6.4 Numerical results and discussion 161
6.5 Summary 162

7 Conclusion 167

Bibliography 173

Research publications/Conferences/Workshops attended 195

List of Figures

1.1 Stability region when the obtained system has complex eigenvalues 22

2.1 Eigenvalues of matrix M with 20 grid points 37
2.2 Eigenvalues of matrix M with 40 grid points 37
2.3 Eigenvalues of matrix M with 60 grid points 42
2.4 Traveling wave solutions of example 2.1 for $\rho = 1$ and $N = 41$ at $t = 0.001, 0.002, 0.003, 0.004, 0.005$ 43
2.5 Time dependent profiles versus x of example 2.2 for $\rho = 1$ and N= 41 at time $t = 1, 2, 3, 4, 5$ 44
2.6 Time dependent profiles versus x of example 2.3 for $\rho = 1$ and N= 41 at time $t = 0.001, 0.004, 0.008, 0.012, 0.016$ 44
2.7 Traveling wave solutions of example 2.3 for $\rho = 1$ and $N = 410$ at $t = 0.001, 0.004, 0.008, 0.012, 0.016$ 45
2.8 Time dependent profiles versus x of example 2.3 for $\rho = 6$ and N= 41 at time $t = 0.0001, 0.0005, 0.001, 0.0015, 0.002$ 45
2.9 Traveling wave solutions of example 2.3 for $\rho = 6$ and $N = 41$ at $t = 0.0001, 0.0005, 0.001, 0.0015, 0.002$ 46

3.1 Eigenvalues of the matrix B for different partitions of the domain in one dimension 52
3.2 The traveling wave solution of example 3.1 at different time levels with $\Delta t = 0.0001$ and $h = 0.04$ 53
3.3 The traveling wave solution of example 3.2 at different time levels with $\Delta t = 0.0001$ and $h = 0.04$ 54
3.4 The traveling wave solution of example 3.3 at different time levels with $\Delta t = 0.0001$ and $N = 51$ 55
3.5 The traveling wave solution of example 3.4 at different time levels with $\Delta t = 0.0001$ and $N = 51$ 55
3.6 Eigenvalues of the matrix B for different partitions of the domain in one dimension 63
3.7 Traveling wave solutions of example 3.5 for time $t \leq 3.5$ with $\nu = 0.005$ and $\Delta t = 0.01$ 64
3.8 The contour plot of example 3.5 for time $t \leq 3.5$ with $\nu = 0.005$ and $\Delta t = 0.01$ 64
3.9 Traveling wave solutions of example 3.6 for time $t \leq 1$ with $\alpha = 1$ and $\Delta t = 0.01$ 66

3.10 Traveling wave solutions of example 3.7 for time $t \leq 3.5$ with $\alpha = 1$, $\nu = 0.002$ and $\Delta t = 0.002$ 67
3.11 The contour plot of example 3.7 for time $t \leq 3.5$ with $\alpha = 1$, $\nu = 0.002$ and $\Delta t = 0.002$ 68
3.12 Eigenvalues of the matrix B in method 1 for different partitions of the domain in one dimension 76
3.13 Eigenvalues of the matrix B in method 2 for different partitions of the domain in one dimension 76
3.14 The comparison of numerical and the exact solutions of example 3.8 for $N = 110$ at $t = 1$ with method 1 78
3.15 The comparison of numerical and the exact solutions of example 3.8 for $N = 100$ at $t = 1$ with method 2 79
3.16 The comparison of numerical and the exact solutions of example 3.9 for $N = 100$ at $t = 1$ with method 1 80
3.17 The comparison of numerical and the exact solutions of example 3.9 for $N = 100$ at $t = 1$ with method 2 81
3.18 The comparison of numerical and the exact solutions of example 3.10 for $N = 200$ at $t = 2$ with method 1 81
3.19 The comparison of numerical and the exact solutions of example 3.10 for $N = 200$ at $t = 2$ with method 2 82
3.20 The comparison of numerical and the exact solutions of example 3.11 for $N = 300$ at $t = 1$ with method 1 82
3.21 The comparison of numerical and the exact solutions of example 3.11 for $N = 350$ at $t = 1$ with method 2 83
3.22 Eigenvalues of the matrix B for different partitions of the domain in one dimension 88
3.23 The physical behavior of exact and numerical solutions of example 3.12 with $\alpha = 2, \beta = \sqrt{2}$ at different time levels using trigonometric B-spline with differential quadrature for $t \leq 2$ 89
3.24 The physical behavior of exact and numerical solutions of example 3.12 with $\alpha = 3, \beta = \sqrt{2}$ at different time levels using trigonometric B-spline with differential quadrature for $t \leq 2$ 89
3.25 The physical behavior of exact and numerical solutions of example 3.13 with $\triangle t = 0.0001$ and $h = 0.0125$ at different time levels using trigonometric B-spline with differential quadrature for $t \leq 5$ 90
3.26 The physical behavior of exact and numerical solutions of example 3.14 taking $\triangle t = 0.0001$ and $h = 0.025$ at different time levels using trigonometric B-spline with differential quadrature for $t < 1$ 90

4.1 Eigenvalues of the matrix B for sine-Gordon equation with different partitions of the domain in two dimensions 107
4.2 Eigenvalues of the matrix B for Burgers' equation with different partitions of the domain in two dimensions 108
4.3 Eigenvalues of the matrix B for Schrödinger equation with different partitions of the domain in two dimensions 108

4.4 Eigenvalues of the matrix B for Telegraph equation with different partitions of the domain in two dimensions 109
4.19 The surface plot of real and imaginary parts of the solution for example 4.7 at time $t = 1$ and $N = M = 30$ 109
4.5 The surface plot of example 4.1 at time $t = 1, 3, 7, 10$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 31$ 110
4.6 L_∞ of example 4.1 at different time $t = 1, 1.5, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 40, 20, 10$ 111
4.7 Errors and convergence with respect to the nodal spacing h with $N_x = N = N_y = M = 80, 40, 20$ for Example 4.1 at time $t = 1$ and $\Delta t = 0.001$ 111
4.8 The surface plot of example 4.2 at time $t = 1, 3, 6, 9$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 31$ 112
4.9 L_∞ of example 4.2 at different time $t = 1, 1.5, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 40, 20, 10$ 113
4.10 Errors and convergence with respect to the nodal spacing h with $N_x = N = N_y = M = 80, 40, 20$ for example 4.2 at time $t = 1$ and $\Delta t = 0.001$ 113
4.11 The surface plot of example 4.3 at time $t = 1, 5.6, 8.4, 11.2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 31$ 114
4.12 L_∞ of example 4.3 at different time $t = 1, 1.5, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 40, 20, 10$ 115
4.13 Errors and convergence with respect to the nodal spacing h with $N_x = N = N_y = M = 80, 40, 20$ for example 4.3 at time $t = 1$ and $\Delta t = 0.001$ 115
4.14 The physical behaviour of numerical solution in example 4.4 at time $t = 0.2, 0.6, 1, 1.4$ for $N = M = 50$ and $k = \Delta t = 0.01$ 116
4.15 The comparison of numerical and the exact solution for example 4.4 at time $t = 0.5$ for $N = M = 30$, $k = \Delta t = 0.0001$ and R=50 . . 116
4.16 The physical behaviour of numerical solution in example 4.5 at time $t = 0.25, 0.5, 0.75, 1$ for $N = M = 60$ and $k = \Delta t = 0.0001$ 117
4.17 The comparison of numerical and the exact solution for example 4.5 at time $t = 0.5$ for $N = M = 30$, $k = \Delta t = 0.0001$ and $R = 50$. 117
4.18 The surface plot of real and imaginary parts of the solution for example 4.6 at time $t = 1$ and $N = M = 30$ 118
4.20 The surface plot of real and imaginary parts of the solution for example 4.8 at time $t = 1$ and $N = M = 30$ 118
4.21 The surface and contour plot of example 4.9 at time $t = 1, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$ 119
4.22 The surface and contour plot of example 4.10 at time $t = 1, 3$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$ 120
4.23 The surface and contour plot of example 4.11 at time $t = 1, 3$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$ 121
4.24 The surface and contour plot of example 4.12 at time $t = 1, 3$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$ 122

5.1 Eigenvalues of the matrix B for KDV equation with different partitions of the domain in one dimension 141
5.2 Eigenvalues of the matrix B for extended Fisher's-Kolmogorov equation with different partitions of the domain in one dimension 141
5.3 Eigenvalues of the matrix B for KS equation with different partitions of the domain in one dimension 142
5.4 Comparison of the exact and numerical solutions of example 5.1 for time $t \leq 3$ with $\Delta t = 0.001$ and $N = 121$ 142
5.5 Surface and contour plot of the exact and numerical solution of example 5.1 for time $t \leq 3$ with $\Delta t = 0.001$ and $N = 121$ 143
5.6 Comparison of the exact and numerical solutions of example 5.2(a) for time $t \leq 6$ with $\Delta t = 0.0001$ and $N = 121$ 143
5.7 Surface and contour plot of the exact and numerical solution of example 5.2(a) for time $t \leq 6$ with $\Delta t = 0.0001$ and $N = 121$ 144
5.8 Comparison of the exact and numerical solutions of example 5.2(b) for time $t \leq 0.3$ with $\Delta t = 0.00001$ and $N = 121$ 144
5.9 Surface and contour plot of the exact and numerical solution of example 5.2(b) for time $t \leq 0.3$ with $\Delta t = 0.00001$ and $N = 121$ . . 145
5.10 Comparison of the exact and numerical solutions of example 5.2(c) for time $t \leq 1.5$ with $\Delta t = 0.0001$ and $N = 121$ 145
5.11 Surface and contour plot of the exact and numerical solution of example 5.2(c) for time $t \leq 1.5$ with $\Delta t = 0.0001$ and $N = 121$ . . . 146
5.12 Comparison of the exact and numerical solutions of example 5.2(d) for time $t \leq 10$ with $\Delta t = 0.001$ and $N = 121$ 146
5.13 Surface and contour plot of the exact and numerical solution of example 5.2(d) for time $t \leq 10$ with $\Delta t = 0.001$ and $N = 121$ 147
5.14 Comparison of the exact and numerical solutions of example 5.2(e) for time $t \leq 1$ with $\Delta t = 0.0001$ and $N = 171$ 147
5.15 Surface and contour plot of the exact and numerical solution of example 5.2(e) for time $t \leq 1$ with $\Delta t = 0.0001$ and $N = 171$ 148
5.16 Comparison of numerical solutions of example 5.6 for different time $t \leq 0.20$ with $\Delta t = 0.0001$, $\gamma = 0$ and $N = 251$ 148
5.17 Comparison of numerical solutions of example 5.6 for different time $t \leq 0.20$ with $\Delta t = 0.0001$, $\gamma = 0.0001$ and $N = 251$ 149
5.18 Comparison of numerical solutions of example 5.7 for different time $t \leq 4.50$ with $\Delta t = 0.001$, $\gamma = 0.0001$ and $N = 121$ 149
5.19 Comparison of numerical solutions of example 5.8 for different time $t \leq 4.50$ with $\Delta t = 0.001$, $\gamma = 0.0001$ and $N = 121$ 150
5.20 Comparison of numerical solutions of example 5.3 for different time $t \leq 4$ with $\Delta t = 0.001$, and $N = 201$ 151
5.21 Surface and contour plot of the exact and numerical solution of example 5.4 for different time $t \leq 4$ with $\Delta t = 0.001$, and $N = 201$. 151
5.22 Comparison of numerical solutions of example 5.4 for different time $t \leq 2$ with $\Delta t = 0.001$, and $N = 201$ 152

5.23 Surface and contour plot of the exact and numerical solution of example 5.4 for different time $t \leq 2$ with $\Delta t = 0.001$, and $N = 201$. 152
5.24 Numerical solution of example 5.5 for different times $t = 1, 5, 10, 20$ with $\Delta t = 0.001$, and $N = 201$ 153

6.1 Eigenvalues of the matrix B for an extended Fisher's-Kolmogorov equation with different partitions of the domain in two dimensions . 163
6.2 Surface plot of the numerical solution of example 6.1 for time $t \leq 1$ with $\Delta t = 0.00001$, $\gamma = 0.0001$ and $N = M = 31$ 163
6.3 Surface and contour plot of the numerical and the exact solutions of example 6.2 at time $t = 0.25$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$ 164
6.4 Surface and contour plot of the numerical and the exact solution of example 6.2 at time $t = 0.75$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$ 164
6.5 Surface and contour plot of the numerical and the exact solution of example 6.2 at time $t = 1$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$ 165
6.6 Surface and contour plot of the numerical and the exact solution of example 6.2 at time $t = 1.25$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$ 166

List of Tables

1.1 The values of cubic trigonometric B-spline basis functions and their derivatives at different node points 11
1.2 The values of quintic trigonometric B-spline basis functions and their derivative at different node points 11

2.1 Absolute errors in the present numerical solution of example 2.1 with $\rho = 1$, $a = 0.2$ and $\triangle t = 0.0001$ at different time levels 38
2.2 Absolute errors in the solution of example 2.2 for $N = 360$ and $\triangle t = 0.01$ at different time levels 39
2.3 Absolute errors in the solution of example 2.2 with $\rho = 1$, $N = 410$ and $\triangle t = 0.00001$ at different time levels 40
2.4 Absolute errors in the solution of example 2.2 with $\rho = 6$, $N = 410$ and $\triangle t = 0.000001$ at different time levels 41
2.5 L_2 and L_∞ errors in the solution of example 2.1 with $\rho = 2000$ and $\triangle t = 0.00001$ at different time levels 42
2.6 L_2 and L_∞ errors in the solution of example 2.2 with $N = 360$ and $\triangle t = 0.01$ at different time levels 42
2.7 L_2 and L_∞ errors in the solution of example 2.3 for $\rho = 1$ with $h = 0.0001$ and $\triangle t = 0.00001$ at different time levels 43
2.8 L_2 and L_∞ errors in the solution of example 2.3 for $\rho = 6$ with $h = 0.0001$ and $\triangle t = 0.000001$ at different time levels 46

3.1 Comparison of L_2, L_∞, and RMS errors in the MCTB-DQM solution of example 3.1 for $\Delta t = 0.0001$ and $h = 0.04$ at different time levels with the errors obtained by other researchers 53
3.2 Comparison of L_2, L_∞, and RMS errors in the MCTB-DQM solution of example 3.2 for $\Delta t = 0.0001$ and $h = 0.04$ at different time levels with the errors obtained by other researchers 54
3.3 Comparison of L_2, L_∞, and CPU time(s) in the MCTB-DQM solution of example 3.3 for $\Delta t = 0.0001$ and $N = 51$ at different time levels with the errors obtained by other researchers 56
3.4 Comparison of L_2, L_∞, and CPU time(s) in the MCTB-DQM solution of example 3.3 for m $\Delta t = 0.0001$ and $N = 101$ at different time levels with the errors obtained by other researcher 56
3.5 Error norms in the MCTB-DQM solution of example 3.4 for m $\Delta t = 0.0001$ and $N = 51$ at different time levels 57

3.6 Comparison of L_2 and L_∞ errors in the TMCB-DQM solution of example 3.5 for $\rho = 0.005$ at different time levels 65
3.7 L_∞ errors in the MTCB-DQM solution of example 3.6 at different time levels 69
3.8 Errors norms in the MTCB-DQM solution of example 3.7 at different time levels with $h = 0.02$, $\Delta t = 0.002$ and $\nu = 0.000666666$ . . . 70
3.9 Errors norms in the MTCB-DQM solution of example 3.7 at different time levels with $h = 0.02$, $\Delta t = 0.002$ and $\nu = 0.002$ 70
3.10 Comparison of errors obtained by applying both of the methods in example 3.8 for $h = 0.02$, $\Delta t = 0.0001$ at different time levels 77
3.11 Comparison of errors obtained by applying both of the methods in example 3.9 for $N = 100$, $\Delta t = 0.0001$ at different time levels 77
3.12 Comparison of errors obtained by applying both of the methods in example 3.10 for $N = 200$, $\Delta t = 0.0001$ at different time levels . . . 78
3.13 Comparison of errors obtained by applying both of the methods in example 3.11 for $N = 350$, $\Delta t = 0.0001$ at different time levels . . . 79
3.14 Comparison of L_2 and L_∞ errors at $t \leq 2$ with $h = 0.02$ and $\triangle t = 0.01$ 91
3.15 Comparison of L_2 and L_∞ errors at $t \leq 2$ with $h = 0.02$ and $\triangle t = 0.0001$ 92
3.16 Comparison of L_2 and L_∞ errors at different time levels ≤ 5 with the errors due to well known earlier schemes 93
3.17 Comparison of L_∞ errors in example 3.14 at different time levels $t \leq 1$ with the errors in the earlier schemes 94
3.18 Comparison L_2 errors in example 3.14 with the errors obtained by other scheme at different time levels $t \leq 1$ 95

4.1 L_∞ error in the MCTB-DQM solution of example 4.1 for $\Delta t = 0.001$ for different values of $N_x = N = N_y = M = 10, 20, 40$ at different time levels 119
4.2 L_∞, RMS, and L_2, errors in the MCTB-DQM solution of example 4.1 for $\Delta t = 0.001$ and for different values of $N_x = N = N_y = M = 20, 40, 80$ at t=1 123
4.3 L_∞ error in the MCTB-DQM solution of example 4.2 for $\Delta t = 0.001$ for different values of $N_x = N = N_y = M = 10, 20, 40$ at t=1 123
4.4 L_∞, RMS, and L_2, errors in the MCTB-DQM solution of example 4.2 for $\Delta t = 0.001$ and for different values of $N_x = N = N_y = M = 20, 40, 80$ at t=1 123
4.5 L_∞ error in the MCTB-DQM solution of example 4.3 for $\Delta t = 0.001$ for different values of $N_x = N = N_y = M = 10, 20, 40$ at different time levels 124
4.6 L_∞, RMS, and L_2, errors in the MCTB-DQM solution of example 4.3 for $\Delta t = 0.001$ and for different values of $N_x = N = N_y = M = 20, 40, 80$ at t=1 124
4.7 Errors norms in the MTCB-DQM solution of example 4.4 at different time levels with $N = M = 60$, and $\Delta t = 0.0001$ 124

4.8 Errors norms in the MTCB-DQM solution of example 4.5 at different time levels with $N = M = 60$, and $\Delta t = 0.0001$ 125
4.9 Comparison of errors obtained by applying both of the methods in example 4.6 for $N = M = 30$, $\Delta t = 0.0001$ at different time levels . 125
4.10 Comparison of errors obtained by applying both of the methods in example 4.7 for $h = 0.033$, $\Delta t = 0.0001$ at different time levels . . . 126
4.11 Comparison of errors obtained by applying both of the methods in example 4.8 for $N = 40$, $\Delta t = 0.0001$ at different time levels 127
4.12 Comparison of error norms obtained by present method in example 4.9 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels 127
4.13 Comparison of error norms obtained by present method in example 4.10 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels 128
4.14 Comparison of error norms obtained by present method in example 4.11 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels 128
4.15 Comparison of error norms obtained by present method in example 4.12 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels 129

5.1 Comparison of errors norms in the numerical solutions computed by present method for Example 5.1 at different time levels with $N = 121$, $\Delta t = 0.001$ 153
5.2 Comparison of errors norms in the numerical solutions computed by present method for example 5.2(a) at different time levels with $N = 121$, $\Delta t = 0.0001$ 154
5.3 Comparison of errors norms in the numerical solutions computed by present method for example 5.2(b) at different time levels with $N = 121$, $\Delta t = 0.00001$ 154
5.4 Comparison of errors obtained by applying the present methods in example 5.2(c) for $N = 121$, $\Delta t = 0.0001$ at different time levels . . 155
5.5 Comparison of errors norms in the numerical solutions computed by present method for example 5.2(d) at different time levels with $N = 121$, $\Delta t = 0.001$ 155
5.6 Comparison of errors norms in the numerical solution computed by present method for example 5.2(e) at different time levels with $N = 171$, $\Delta t = 0.0001$ 156
5.7 Comparison of errors norms in the numerical solution computed by present method for example 5.3 at different time levels with $N = 171$, $\Delta t = 0.0001$ 156
5.8 Comparison of errors norms in the numerical solution computed by present method for example 5.4 at different time levels with $N = 171$, $\Delta t = 0.0001$ 157

6.1 Error norms in the numerical solution computed by present method for example 6.2 at different time levels with $N = M = 41$, $\Delta t = 0.0001$ and $\gamma = 0.00001$ 165

6.2 Comparison of errors norms computed by present method for example 6.2 at time $t = 1$ with different values of N and M ($N = M = 30, 40, 50$) 166

Dedicated to...

My family and Sahib Ji

Chapter 1

Introduction

1.1 Introduction

Many of the mathematical models of engineering problems are expressed in terms of ordinary and partial differential equations. These differential equations are the powerful tool to express various phenomena in science, for instance, quantum mechanics, electromagnetic fields, fluid flow diffusion, and the physical laws of structural mechanics. Because of having many applications in different disciplines, differential equations have been solved in the literature using various analytical and numerical methods. During the mathematical modeling of the problems, most of the time the modeled differential equation is not so easy to solve. Hence to obtain an analytical or exact solution for that problem is a challenging task. This results in the requirement of advanced numerical methods which can be used to get an accurate numerical solution. So this research aims to investigate the execution of collocation method and differential quadrature technique with trigonometric B-spline basis functions to find the numerical solutions of some important linear and nonlinear time dependent partial differential equations.

In the last couple of years, the large number of numerical methods are proposed to find the solution of initial and boundary value problems. Some of the well-known methods include the finite difference (FD) method and finite element (FE) method. But collocation method is one of the developing well-known method developed for a variety of numerical problems to get the solution of different linear and nonlinear differential equations. It includes fulfilling a differential equation to some resilience at some chosen limited number of points, called collocation points.

Collocation method has a couple of unique favorable advantages over the FD and FE methods. One of the major advantages of collocation method over FD method includes the closed form of the numerical solution, which is also piecewise continuous. As compared to FE method, collocation method is easy to be applied to numerous problems including initial and boundary value problems.

Partial differential equations are broadly examined by many researchers in a previous couple of years due to an indispensable utilization of these equations in different fields of research by collocation method. Some of the work is as follows: cubic trigonometric B-spline is used with collocation to solve the hyperbolic equation by Abbas and Majid [1]. Botella [2] applied the collocation method to solve Navier-Stokes equation. Can and Dag [3] solved Burgers' equation with collocation B-spline method. Fernandes and Fairweather [4] used an ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems. A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines collocation is used by Govindarajan et al. [5]. Gupta and Kukreja [6] solved diffusion problems using cubic B-spline collocation method. Linear rational spline collocation for linear boundary value problems is given by Ideon and Oja [7]. A high order B-spline collocation method for linear boundary value problems is given by Jator and Sinkala [8]. The work of Kadalbajoo on collocation method is published for many well known equations [9–16]. The generalized Kuramoto-Sivashinsky equation is solved by Lakestani and Dehghan [17] using spline collocation method. Lang and Xu [18] used quintic B-spline collocation method for the second order mixed boundary value problem. The sub-diffusion equation is solved by orthogonal spline collocation method by Li et al. [19]. The kuramoto-sivashinsky equation is solved numerically by Mittal and Arora [20] using quintic B-spline collocation method. Cubic B-splines collocation method is also used to solve one dimension hyperbolic telegraph equation and convection-diffusion equation by Mittal [21, 22]. Euler-Bernoulli beam models are solved by Mohammadi [23] using sextic B-spline collocation method. Morinishi and Tamano [24] gave collocation method for compressible turbulent channel flow. An optimal B-spline collocation method for self-adjoint singularly perturbed boundary value problems is given by Rao and Kumar [25]. Siddiqi and Arshed [26] solved good Boussinesq equation by using quintic B-spline collocation method. Numerical solution of nonlinear parabolic partial differential equations with Neumann boundary conditions is given by Mittal and Jain [27].

Collocation method has also be extended and explored in various aspects to solve

fractional and integral differential equations such as fractional differential equations are solved by Li [28] using wavelet collocation method. Orthogonal spline collocation method is used by Yang et al. [29] for the two-dimensional fractional sub-diffusion equation. Fourth order partial integro-differential equations with a weakly singular kernel are solved by Zhang et al. [30]. Sahu and Ray [31] solved Fredholm integral equations of the second kind by a new approach involving semiorthogonal B-spline wavelet collocation method. Pedas and Tamme [32] solved nonlinear fractional differential equations by spline collocation method.

Another well known numerical method used to solve ordinary and partial differential equations is differential quadrature method (DQM), introduced by Bellman and Casti in the early 1970s [33, 34]. This method revisited in the later 1980s [35] and has been gradually emerging as a unique numerical solution technique for the initial and boundary value problems of physical and engineering sciences [36]. In fact, DQM can also be formulated via the polynomial-based collocation method, which is one of the popular method [37, 38]. As a numerical method, DQM has been applied successfully to solve differential equations existing in the fields of biosciences, transport processes, fluid mechanics, static and dynamic structural mechanics, static aeroelasticity, and lubrication mechanics. It has been established that DQM is simple and can yield highly accurate numerical solutions with minimal computational effort. To overcome some deficiency existing in the conventional DQM, differential quadrature-based element methods are also proposed which has extended the application range of the DQM in dealing with complex geometry and boundary conditions. The developed DQM has seemingly high potential as an alternative to the classical finite difference and FEMs [36]. Due to its attractive features of rapid convergence, high accuracy, and computational efficiency, DQM is now a well-known numerical method. The early developments of the DQM and its applications in general engineering up to the year of 1999 are well documented in a book written by Professor Chang Shu [39]. New developments on the DQ method and its applications to structural mechanics have been made since then. Although the progression of the development and application of DQM in the area of structural mechanics is clear from the past researches, but these have been scattered over many papers. In addition, a variety of different quadrature formulations by varying the degree of the polynomials, treatment of boundary conditions and employment of regular or irregular grid points also exist in the literature. The utmost requirement of DQM is the calculation of weighting

coefficient whose formulation was further improved by Quan and Chang [40, 41]. Various kinds of test functions such as spline functions, Lagrange interpolation polynomials, sinc function [40–42], etc. are successfully implemented to determine the weighting coefficients.

A lot of work is accounted by Korkmaz in literature for DQM like sinc differential Quadrature method [42], polynomial based differential quadrature method [43], cubic B-spline differential quadrature method [44] and quartic B-splines differential quadrature method [45]. Initially, Bellman [33, 34, 46] started the work on DQM and solve the inverse problem, long term integration, and nonlinear partial differential equations. After that Bellomo [47] gave the solution of nonlinear model in applied science. Bert [35–37] gave his contribution to exploring DQM with different problems. Geometrically nonlinear bending of orthotropic rectangular plates, Poisson, and convection-diffusion equations are solved by Chen [48–50]. Multivariable models are solved by the quadrature and cubature methods by Civan [51]. Multiquadric radial basis function with DQM is given by Nam and Thanh [52, 53] to solve the differential equations. Two-dimensional incompressible Navier-Stokes equations are solved by Shu and Richards [54] using generalized DQM, for further details on DQM we refer to [41, 54–56]. Dual reciprocity BEM is applied to transient elastodynamic problems with differential quadrature method in time by Tanaka and Chen [57]. Wu [58–60] solved the driven flow by linearized elimination, domain decomposition method for problems on a triangular domain and non-linear Burgers' equation using differential quadrature method. Three dimensional telegraphic equation is solved by Mittal and Dahiya [61] using B-spline DQM. A novel differential quadrature element method was suggested for vibration analysis of hybrid nonlocal Euler-Bernoulli beams by Wang [62].

As mentioned above, a lot of work has been represented in the literature using DQM due to its distinct properties. The first advantage of DQM over the collocation method includes accurate results even for the small number of knots and second, there is no need of linearization in DQM while solving nonlinear problems. In the present research work, trigonometric basis functions of the third and fifth degree have been used with the collocation and differential quadrature methods with an introduction to basis spline functions given in the coming sections.

1.2 Idea of spline

Let us consider a problem of fitting a polynomial passing through points whose function values are given. If the number of points are two, we get a linear polynomial. If data of 4 points is given, a cubic polynomial can be fitted and so on. Thus, the degree of polynomial goes on increasing with the increase in the number of data points. When the number of points are large, polynomial of higher degree exists, which is not easy to work with. So piecewise polynomial comes in to picture to resolve the issues of working with higher degree polynomials. A polynomial which approximates the function over some part of the domain is called piecewise polynomial. This approximation permits us to build precise approximation, but since a portion of the approximated polynomials is not smooth, so acquired function is not smooth at the point joining separate piecewise polynomials. Sometimes it may also possible that the graph of interpolant is not smooth because every polynomial is always continuous but it may or may not be differentiable on the interval. To resolve this problem splines functions are used. Splines are used to develop a piecewise polynomial which interplants the given information or the function values and also continuously differentiable to some degree.

1.2.1 Definition of spline

The spline is a gadget utilized by shipbuilders and draftsmen to draw a bend through pre-referred points (Knots) in a manner such that the curve as well as its slant and shape are continuous function. Designers join some weights called ducks, to keep the strip fit as a fiddle. In numerical treatment, a spline is a function characterized piecewise by polynomials. Consider a uniform partition $x_0 < x_1 < ... < x_{n-1} < x_n$ of the space $[a, b]$ with $x_0 = a$, $x_n = b$ and abscissas x_i called as knots. A function $S(x)$, a spline of degree k is a k^{th} degree polynomial $p(x)$ in each of the interval $[x_i, x_{i+1}]$, $i = 0, 1, 2..., n-1$ with the property that $p(x)$ and its first $(k-1)$ derivatives are likewise continuous in $[x_0, x_n]$
In this manner a spline $S(x)$ on interval $[a, b]$ can be characterized as:

$$S(x) = \sum_{i=1}^{n} p(x_i)$$

Here $p(x)$ is a k^{th} degree polynomial in every section. Hence the number of coefficients in every portion is $(k+1)$, that results the aggregate number of coefficients to be $n(k+1)$ as there are n sections. In this manner, to characterize a spline, we require $n(k+1)$ number of equations.

Many attempts have been made while applying spline collocation method to compute the numerical solution of differential equations. These incorporate the particular choice of spline basis functions, the order of the spline function, the regularity of the spline curve to employ, and the locations of the collocation points. The application of spline collocation method to get the numerical solutions of the differential equation requires a great deal of work that has been accounted for utilizing spline functions of different degrees. For example, singular two-point boundary value problems have been comprehended by Iyengar and Jain [63] utilizing spline difference method. Khuri and Sayfy [64] introduced a numerical method, in view of spline collocation and finite differences for the numerical approximation of a generalized Fisher's integro-differential equation. Kanth and Reddy [65] solved two-point boundary value problems using the collocation strategy utilizing cubic spline function. Not only low order splines but also the higher order splines are additionally used to take care of higher order boundary value problems. For example, the solution of eighth order boundary value problem has been obtained by Siddiqi and Twizell [66] using an octic spline. Akram and Siddiqi [67] produced collocation strategy to approximate solution of eighth order linear special case boundary value problem utilizing nonic spline. The concept of spline being useful is extended to many other splines and one of them is B-spline.

1.2.2 B-spline basis function

The initial reference to the word B-spline (B stands for basis) in the field of mathematics was given by Schoenberg [68] in 1946, who portrayed it as a smooth piecewise polynomial estimate and is short for basis spline. A B-spline [69] is characterized as a spline function that has minimal support for a given degree, smoothness, and space partition.

The basic core of the B-spline is its basis function. The characterizing highlight of the basis function is the knot sequence x_i. Let X be an arrangement of $N+1$

non-diminishing real numbers $x_0 \leq x_1 \leq x_2 \leq ... \leq x_N$. Here x_i are the knots, and the half-open interval $[x_i, x_{i-1})$ represents the i^{th} knot span. If the knots are equally spaced, the knot partition is said to be uniform else called as non uniform partition. Each B-spline function of degree k covers $k+1$ knots or k intervals. B-spline basis function occurs in various degree and form. Some of the important forms of B-spline includes its standard form, trigonometric form, exponential form and cardinal form. In this research work, we have used trigonometric B-spline basis functions which can be calculated by following formula [69]:

$$T_m^k(x) = \frac{sin\frac{x-x_m}{2}}{sin\frac{x_{m+k}-x_m}{2}} T_m^{k-1}(x) + \frac{sin\frac{x_{m+k}-x}{2}}{sin\frac{x_{m+k}-x_m}{2}} T_{m+1}^{k-1}$$

for $k = 2, 3, 4, ...$. This recurrence relation demonstrates that the trigonometric B-spline basis elements of a higher degree can be steadily assessed as a straight mix of basis elements of lower degree. The recurrence formula begins with the principal degree trigonometric B-splines, and fabricates the premise elements of progressively higher degree. For degree $k \geq 1$ basis function $T_{m,k}(x)$ is a direct blend of two $(k-1)^{th}$ degree basis function. $T_m(x)$ is a piecewise trigonometric function with geometric properties like C^∞ continuity, non-negativity and partition of unity. Trigonometric B-spline functions can be further classified based on the degree as follows:

Trigonometric B-spline of degree zero:

The zero degree trigonometric B-spline basis function $TB_m(x)$, for $m = 1, 2, ..., N$ is defined as:

$$TB_m^0(x) = \begin{cases} 1, & x \in [x_m, x_{m+1}) \\ 0, & otherwise \end{cases} \tag{1.1}$$

Trigonometric B-spline of degree one:

The first degree trigonometric B-spline basis function $TB_m(x)$, for $m = 0, 1, ..., N$ is defined as:

$$TB_m^1(x) = \frac{1}{sin\frac{h}{2}} \begin{cases} sin(\frac{x-x_m}{2}), & x \in [x_m, x_{m+1}) \\ sin(\frac{x_{m+1}-x}{2}), & x \in [x_{m+1}, x_{m+2}) \\ 0, & otherwise \end{cases} \tag{1.2}$$

Trigonometric B-spline of degree two:

The second degree trigonometric B-spline basis function $TB_m(x)$, for $m = 0, 1, ..., N+$

1 is defined as:

$$TB_m^2(x) = \frac{1}{sin\frac{h}{2}sinh}\begin{cases} sin^2(\frac{x-x_m}{2}), & x \in [x_m, x_{m+1}) \\ sin(\frac{x-x_m}{2})sin(\frac{x_{m+2}-x}{2}) + sin(\frac{x_{m+3}-x}{2})sin(\frac{x-x_{m+1}}{2}), & x \in [x_{m+1}, x_{m+2}) \\ sin^2(\frac{x_{m+3}-x}{2}), & x \in [x_{m+2}, x_{m+3}) \\ 0, & otherwise \end{cases} \tag{1.3}$$

Trigonometric B-spline of degree three:

The cubic trigonometric B-spline basis function $TB_m(x)$, for $m = -1, 0, ..., N+1$ is characterized as:

$TB_m^3(x) =$

$$\frac{1}{w}\begin{cases} p^3(x_{m-2}), & x \in [x_{m-2}, x_{m-1}) \\ p(x_{m-2})(p(x_{m-2})q(x_m) + q(x_{m+1})p(x_{m-1})) + q(x_{m+2})p^2(x_{m-1}), & x \in [x_{m-1}, x_m) \\ q(x_{m+2})(p(x_{m-1})q(x_{m+1}) + q(x_{m+2})p(x_m)) + p(x_{m-2})q^2(x_{m+1}), & x \in [x_m, x_{m+1}) \\ q^3(x_{m+2}), & x \in [x_{m+1}, x_{m+2}) \\ 0, & otherwise \end{cases} \tag{1.4}$$

where $p(x_m) = \sin(\frac{x-x_m}{2})$, $q(x_m) = \sin(\frac{x_m-x}{2})$, $w = \sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})$ and $h = \frac{b-a}{n}$.

Trigonometric B-spline of degree four:

The quartic trigonometric B-spline basis function $TB_m(x)$, for $m = -1, 0, ..., N+2$ is characterized as:

$$TB_m^4(x) = \frac{1}{w}\begin{cases} p^4(x_{m-3}), & x \in [x_{m-3}, x_{m-2}) \\ p(x_{m-3})(p^2(x_{m-3})q(x_{m-1}) + p(x_{m-3})q(x_m)p(x_{m-2}) & \\ +q(x_{m+1})p^2(x_{m-2})) + q(x_{m+2})p^2(x_{m-2}), & x \in [x_{m-2}, x_{m-1}) \\ p^2(x_{m-3})q^2(x_{m-3}) + p(x_{m-3})p(x_{m-2})q(x_m)q(x_{m+1}) & \\ +p(x_{m-3})q^2(x_{m+1})p(x_{m-1}) + q(x_{m+2})(p^2(x_{m-2}) & \\ q(x_m + p(x_{m-2})q(x_{m+1})p(x_{m-1}) + q(x_{m+2})p^2(x_{m-1})), & x \in [x_{m-1}, x_m) \\ p(x_{m-3})q^3(x_{m+1}) + q(x_{m+2})(p(x_{m-2})q^2(x_{m+1})) & \\ +q^2(x_{m+2})p(x_m) + p(x_{m-1})q(x_{m+1})q(x_{m+2}), & x \in [x_m, x_{m+1}) \\ q^4(x_{m+2}), & x \in [x_{m+1}, x_{m+2}) \\ 0, & otherwise \end{cases} \tag{1.5}$$

where $p(x_m) = \sin(\frac{x-x_m}{2})$, $q(x_m) = \sin(\frac{x_m-x}{2})$, $w = \sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})\sin(2h)$ and $h = \frac{b-a}{n}$.

Trigonometric B-spline of degree five:

The quintic trigonometric B-spline basis function $TB_m(x)$, for $m = -2, -1, 0, ..., N+2$ is characterized as:

$$TB_m^5(x) = \frac{1}{w}\begin{cases} p^5(x_{m-3}), & x \in [x_{m-3}, x_{m-2}) \\ \\ p^4(x_{m-3})q(x_{m-1}) + p^3(x_{m-3})q(x_m)p(x_{m-2}) & \\ +p^2(x_{m-3})q(x_{m+1})p^2(x_{m-2}) + p(x_{m-3})q(x_{m+2}) & \\ p^3(x_{m-2}) + q(x_{m+3})p^4(x_{m-2}), & x \in [x_{m-2}, x_{m-1}) \\ \\ p^3(x_{m-3})q^2(x_m) + p^2(x_{m-3})q(x_{m+1})p(x_{m-2})q(x_m) & \\ +p^2(x_{m-3})q^2(x_{m+1})p(x_{m-1}) + p(x_{m-3})q(x_{m+2})p^2(x_{m-2}) & \\ q(x_m) + p(x_{m-3})q(x_{m+2})p(x_{m-2})q(x_{m+1})p(x_{m-1}) & \\ +p(x_{m-3})q^2(x_{m+2})p^2(x_{m-1}) + q(x_{m+3})p^3(x_{m-2})q(x_m) & \\ +q(x_{m+3}) & \\ p^2(x_{m-2})q(x_{m+1})p(x_{m-1}) + q(x_{m+3})p(x_{m-2}) & \\ q(x_{m+2})p^2(x_{m-1}) + q^2(x_{m+3})p^3(x_{m-1}), & x \in [x_{m-1}, x_m) \\ \\ p^2(x_{m-3})q^3(x_{m+1}) + p(x_{m-3})q(x_{m+2})p(x_{m-2})q^2(x_{m+1}) + p(x_{m-3}) & \\ q^2(x_{m+2})p(x_{m-1})q(x_{m-2}) + p(x_{m-3})q^3(x_{m+2})p(x_m) + q(x_{m+3}) & \\ p^2(x_{m-2})q^2(x_{m+1}) + q(x_{m+3})p(x_{m-2})q(x_{m+2})p(x_{m-1}) & \\ q(x_{m+1}) + q(x_{m+3})p(x_{m-2})q^2(x_{m+2})p(x_m) + q^2(x_{m+3}) & \\ p^2(x_{m-1})q(x_{m+1}) + q^2(x_{m+3})p(x_{m-1})q(x_{m+2})p(x_m) & \\ q^3(x_{m+3})p^2(x_m), & x \in [x_m, x_{m+1}) \\ \\ p(x_{m-3})q^4(x_{m+2}) + q(x_{m+3})p(x_{m-2})q^3(x_{m+2}) & \\ +q^2(x_{m+3})p(x_{m-1})q^2(x_{m+2}) + q^3(x_{m+3})p(x_m) & \\ q(x_{m+2}) + q^4(x_{m+3})p(x_{m+1}), & x \in [x_{m+1}, x_{m+2}) \\ \\ q^5(x_{m+3}), & x \in [x_{m+2}, x_{m+3}) \\ 0, & otherwise \end{cases} \tag{1.6}$$

where $p(x_m) = \sin\left(\frac{x-x_m}{2}\right)$, $q(x_m) = \sin\left(\frac{x_m-x}{2}\right)$, $w = \sin\left(\frac{h}{2}\right)\sin(h)\sin\left(\frac{3h}{2}\right)\sin(2h)\sin\left(\frac{5h}{2}\right)$ and $h = \frac{b-a}{n}$.

TABLE 1.1: The values of cubic trigonometric B-spline basis functions and their derivatives at different node points

x	x_{m-2}	x_{m-1}	x_m	x_{m+1}	x_{m+2}
TB_m	0	a_1	a_2	a_1	0
TB'_m	0	a_3	0	a_4	0
TB''_m	0	a_5	a_6	a_5	0

TABLE 1.2: The values of quintic trigonometric B-spline basis functions and their derivative at different node points

x	x_{m-3}	x_{m-2}	x_{m-1}	x_m	x_{m+1}	x_{m+2}	x_{m+3}
TB_m	0	b_1	b_2	b_3	b_2	b_1	0
TB'_m	0	b_4	b_5	b_6	$-b_5$	$-b_4$	0

The values of $TB_m(x)$ and its derivatives which are used in our research work are presented in Table 1.1 and 1.2 only for cubic and quintic trigonometric B-spline basis functions. The values of a_i and b_i used in the tables are given as follows:

$$a_1 = \frac{sin^2(\frac{h}{2})}{sin(h)sin(\frac{3h}{2})},\ a_2 = \frac{2}{1+2cos(h)},$$

$$a_3 = \frac{-3}{4sin(\frac{3h}{2})},\ a_4 = \frac{3}{4sin(\frac{3h}{2})},$$

$$a_5 = \frac{3(1+3cos(h))}{16sin^2(\frac{h}{2})(2cos(\frac{h}{2})+cos(\frac{3h}{2}))}, a_6 = \frac{-3cos^2(\frac{h}{2})}{sin^2(\frac{h}{2})(2+4cos(h))}.$$

$$b_1 = \frac{sin^4(\frac{h}{2})}{2sin(\frac{3h}{2})sin(\frac{5h}{2})cos(h)sin^2(h)},\ b_2 = \frac{sin(\frac{h}{2})(\frac{3sin(2h)}{4} - sin(3h) + \frac{3sin(h)}{2})}{sin(2h)sin(\frac{3h}{2})sin(\frac{5h}{2})sin(h)},$$

$$b_3 = \frac{\frac{3sin(\frac{h}{2})}{4} + sin(\frac{3h}{2}) - \frac{3sin(5h)}{4}}{sin(\frac{5h}{2})cos(h)sin^2(h)},\ b_4 = -\frac{5(sin(2h)-2sin(h))}{16sin(2h)sin(\frac{3h}{2})sin(\frac{5h}{2})sin(h)},$$

$$b_5 = \frac{-\frac{-5sin^2(2h)}{8} - \frac{15sin^2(\frac{h}{2})}{16} + \frac{15sin^2(\frac{3h}{2})}{16} + \frac{5sin^2(h)}{8}}{sin(2h)sin(\frac{h}{2})sin(\frac{3h}{2})sin(\frac{5h}{2})sin(h)}, \qquad b_6 = 0.$$

1.3 Collocation method

Considerable work has been accounted in the literature utilizing the B-spline collocation strategy for understanding and solving the partial differential equation, particularly on account of issues where constantly continuity of higher order derivatives is required. For the smooth approximations, B-spline functions are exceptionally well known and have been utilized by various scientists to obtain approximate solutions of linear and nonlinear differential equations. In collocation method, for an initial or boundary value problem the numerical solution can be presented in form of a linear combination of coordinate functions. In this method, we approximate a function by a polynomial through values of the function at selected points. The selected points are known as collocation points.

While applying the method, one can easily do the partition of the domain $[a, b]$ into the consistently estimated limited components of span h by the knots x_m such that $a = x_0 < x_1 < x_2 < ... < x_{n-1} < x_n = b$. Let $TB_m(x)$ be trigonometric cubic B-splines with knots at the points $x_m, m = -1, 0, 1, 2, ..., N + 1$. The arrangement of splines $\{TB_{-1}, TB_0, TB_1, ..., TB_N, TB_{N+1}\}$ frames a foundation for functions defined over $[a, b]$. An approximation $U_N(x, t)$ to the exact solution $u(x, t)$ can be expressed in terms of the cubic trigonometric B-spline trial functions as:

$$U(x,t) = \sum_{j=m-k+2}^{m+k-2} C_j(t)TB_j(x). \tag{1.7}$$

where m represents number of node points, k is degree of basis function.

When $k = 3$ the above expression converted to:

$$U(x,t) = \sum_{j=m-1}^{m+1} C_j(t)TB_j(x). \tag{1.8}$$

where $C(t)$ are the time subordinate quantities which can be computed by solving the above system with initial and boundary conditions used to write the numerical solution of collocation method. Using the values of cubic trigonometric B-spline

basis function as tabulated in Table 1.1, the approximations for the value of $U(x)$, $U^{'}(x)$ and $U^{''}(x)$ at the knots can be written in the terms of C_m as:

$$\begin{aligned} U(x_m,t) &= a_1C_{m-1} + a_2C_m + a_1C_{m+1} \\ U'(x_m,t) &= a_3C_{m-1} + a_4C_{m+1} \\ U''(x_m,t) &= a_5C_{m-1} + a_6C_m + a_5C_{m+1} \end{aligned} \tag{1.9}$$

where the first and second derivatives with respect to x of $U(x)$ are denoted by $U'(x_m,t)$ and $U''(x_m,t)$.

1.4 Differential quadrature method

Differential quadrature method is a numerical technique for finding the solution of initial and boundary value problems. In recent years, differential quadrature method [34] has been successfully implemented to solve a variety of problems in engineering and physical sciences mainly in the field of fluid mechanics, vibration analysis, and structural analysis. In this method, the derivative of a function at any location is approximated by a linear summation of all the functional values along a grid line. Many numerical examples have also shown the super accuracy, efficiency, convenience and the great potential of this method (see [40, 54, 70]).

In differential quadrature method, the weighting coefficients are evaluated using various test functions such as spline functions, Lagrange interpolation polynomials, cubic B-splines, modified cubic B-splines, trigonometric B-spline and sinc function etc. The trigonometric B-spline differential quadrature method is the differential quadrature method in which the weighting coefficients are obtained by using trigonometric B-spline functions as a set of basis functions. Let $TB_m(x)$ be trigonometric B-splines with knots at the points $x_n, n = 0, 1, 2, ..., N$. To fit a spline on the defined grid points, one needs to add additional knot points on both sides of the domain partition. Hence the arrangement of splines $\{TB_{-1}, TB_0, TB_1, ..., TB_N, TB_{N+1}\}$ frames a foundation for functions defined over domain $[a, b]$.

Since weighting coefficients in DQM are dependent on the spatial grid spacing, one can assume N grid points on the real axis distributed uniformly, that is, $a = x_1 < x_2, ..., x_{N-1} < x_N = b$ with $x_{j+1} - x_j = h$. The solution $u(x,t)$ at any time on the knot x_j is $u(x_j,t)$ for $j = 1, 2, ..., N$. The approximate values of derivatives can be computed as follows:

$$\begin{aligned} U_x &= \sum_{j=1}^{N} a_{ij} u(x_j,t),\ U_{xx} = \sum_{j=1}^{N} b_{ij} u(x_j,t), \\ U_{xxx} &= \sum_{j=1}^{N} c_{ij} u(x_j,t),\ U_{xxxx} = \sum_{j=1}^{N} d_{ij} u(x_j,t), \end{aligned} \tag{1.10}$$

In the similar way the approximate values of derivatives in two dimensions can be computed as follows:

$$\begin{aligned} U_x &= \sum_{j=1}^{N} a_{ij} u(x_j,y,t),\ U_{xx} = \sum_{j=1}^{N} b_{ij} u(x_j,y,t), \\ U_{xxx} &= \sum_{j=1}^{N} c_{ij} u(x_j,y,t),\ U_{xxxx} = \sum_{j=1}^{N} d_{ij} u(x_j,y,t) \\ U_y &= \sum_{j=1}^{N} a_{ij}^{'} u(x,y_j,t),\ U_{yy} = \sum_{j=1}^{N} b_{ij}^{'} u(x,y_j,t), \\ U_{yyy} &= \sum_{j=1}^{N} c_{ij}^{'} u(x,y_j,t),\ U_{yyyy} = \sum_{j=1}^{N} d_{ij}^{'} u(x,y_j,t). \end{aligned} \tag{1.11}$$

for $i = 1, 2, ..., N$. The first and higher space derivatives can thus be computed approximately using a_{ij}, b_{ij}, c_{ij}, d_{ij}, $a_{ij}^{'}$, $b_{ij}^{'}$, $c_{ij}^{'}$, and $d_{ij}^{'}$, which are the weighting coefficients. To calculate the first order derivative in x direction, the computation of a_{ij} is required which is explained in the next two sub sections for cubic and quintic trigonometric B-spline basis functions respectively. After calculating a_{ij}, rest of the weighting coefficient can be calculated either independently or depending upon values of a_{ij} as discussed in coming section.

1.4.1 Computation of the weighting coefficients(a_{ij}) for cubic trigonometric B-spline function

The first order approximation at the grid point x_i, $i = 1, 2, ..., N$ can be written as:

$$TB_m^{'}(x_i) = \sum_{j=1}^{N} a_{ij} TB_m(x_j) \ for \ m = 1, 2, ..., N. \tag{1.12}$$

The solution can be obtained by using trigonometric cubic B-spline basis functions. In this research work, we have used modified trigonometric B-spline basis functions which are obtained from the trigonometric B-spline basis. These basis functions are modified in such a way that the resulting matrix system of equations become diagonally dominant. The modified basis functions at the knots are defined as follows [70]:

$$MTB_1(x) = TB_1(x) + 2TB_0(x),$$

$$MTB_2(x) = TB_2(x) - TB_0(x),$$

$$MTB_j(x) = TB_j(x) \ for \ j = 3, 4, ..., N-2.$$

$$MTB_{N-1}(x) = TB_{N-1}(x) - TB_{N+1}(x).$$

$$MTB_N(x) = TB_N(x) + 2TB_{N+1}(x),$$

where $\{TB_1, TB_2, ..., TB_N\}$ form a basis over the region $[a, b]$.

Using modified basis functions in the equation (1.12), it can be written as:

$$MTB_m^{'}(x_i) = \sum_{j=1}^{N} a_{ij} MTB_m(x_j) \ for \ m = 1, 2, ..., N. \tag{1.13}$$

that results in a matrix system

$$A\overrightarrow{a}[i] = \overrightarrow{R}[i] \tag{1.14}$$

where A is a coefficient matrix given by:

$$\begin{bmatrix} a_2+2a_1 & a_1 & 0 & 0 & . & 0 \\ 0 & a_2 & a_1 & . & . & 0 \\ 0 & a_1 & a_2 & a_1 & . & 0 \\ 0 & 0 & a_1 & a_2 & a_1 & 0 \\ 0 & . & . & . & . & 0 \\ 0 & 0 & . & . & a_1 & a_2+2a_1 \end{bmatrix} \tag{1.15}$$

and $\overrightarrow{a}[i]$ denotes the weighting coefficient vector corresponding to grid point x_i, that is $\overrightarrow{a}[i] = [a_{i1}, a_{i2}, ..., a_{iN}]^T$ and the coefficient matrix $\overrightarrow{R}[i] = [r_{i1}, r_{i2}, ..., r_{iN}]^T$ corresponding to $x_i, i = 1, 2, ..., N.$ are evaluated as:

$$\begin{gathered} \overrightarrow{R}[1] = [-2a_4, a_3 - a_4, 0, ..., 0]^T, \\ \overrightarrow{R}[2] = [a_4, 0, a_3, 0, ..., 0]^T, \\ \overrightarrow{R}[3] = [0, a_4, 0, a_3, 0, ..., 0]^T, \\ . \\ . \\ . \\ \overrightarrow{R}[N-1] = [0, ..., 0, a_4, 0, a_3]^T, \\ \overrightarrow{R}[N] = [0, ..., 0, a_4 - a_3, 2a_3]^T. \end{gathered}$$

For solving the above tridiagonal matrix system we used programming in MATLAB that results in the vector $\overrightarrow{a}[i]$, i.e. the weighting coefficients $a_{i1}, a_{i2}, ..., a_{iN}$ for $i = 1, 2, ..., N$.

In similar way a'_{ij} can be computed, which is further used to calculate the first order derivative with respect to y.

1.4.2 Computation of the weighting coefficients(a_{ij}) for quintic trigonometric B-spline function

The first order approximation at the grid point x_i, $i = 1, 2, ..., N$ can be written as:

$$TB_m^{'}(x_i) = \sum_{j=1}^{N} a_{ij} TB_m(x_j) \; for \; m = 1, 2, ..., N. \tag{1.16}$$

that results in a matrix system

$$A\overrightarrow{a}[i] = \overrightarrow{R}[i] \tag{1.17}$$

where A is the coefficient matrix given by:

$$\begin{bmatrix} b_3 & b_2 & b_1 & 0 & 0 & 0 & . & 0 \\ b_2 & b_3 & b_2 & b_1 & 0 & 0 & . & 0 \\ b_1 & b_2 & b_3 & b_2 & b_1 & 0 & . & 0 \\ . & . & . & . & . & . & . & . \\ 0 & . & 0 & b_1 & b_2 & b_3 & b_2 & b_1 \\ 0 & . & 0 & 0 & b_1 & b_2 & b_3 & b_2 \\ 0 & . & 0 & 0 & 0 & b_1 & b_2 & b_3 \end{bmatrix} \tag{1.18}$$

and $\overrightarrow{a}[i]$ denotes the weighting coefficient vector corresponding to grid point x_i, that is $\overrightarrow{a}[i] = [a_{i1}, a_{i2}, ..., a_{iN}]^T$ and
the coefficient matrix $\overrightarrow{R}[i] = [r_{i1}, r_{i2}, ..., r_{iN}]^T$ corresponding to $x_i, i = 1, 2, ..., N$. are evaluated as

$$\overrightarrow{R}[1] = [b_6, b_5, b_4, 0, ..., 0]^T,$$
$$\overrightarrow{R}[2] = [-b_5, b_6, b_5, b_4, 0, ..., 0]^T,$$
$$\overrightarrow{R}[3] = [-b_4, -b_5, b_6, b_5, b_4, ..., 0]^T,$$

$$
\begin{gathered}
. \\
. \\
. \\
\overrightarrow{R}[N-2] = [0, ..., -b_4, -b_5, b_6, b_5, b_4]^T, \\
\overrightarrow{R}[N-1] = [0, ..., 0, -b_4, -b_5, b_6, b_5]^T, \\
\overrightarrow{R}[N] = [0, ..., 0, -b_4, -b_5, b_6]^T.
\end{gathered}
$$

For solving the above five band matrix system we used programming in MATLAB, that results in the vector $\overrightarrow{a}[i]$, i.e. the weighting coefficients $a_{i1}, a_{i2}, ..., a_{iN}$ for $i = 1, 2, ..., N$.

In a similar way, a'_{ij} can be computed, which is further used to calculate the first order derivative with respect to y.

1.4.3 Computation of the weighting coefficients for high order derivatives

There are different approaches to calculate the weighting coefficients for high order derivatives. One of the approaches is given by X. Wang [71] which is as follows:

1. Second order derivative can be obtained as follows:

$$
\begin{aligned}
\frac{\partial^2 u_i}{\partial x^2} &= \frac{\partial}{\partial x}\Big(\frac{\partial u}{\partial x}\Big) = \sum_{k=1}^{N} a_{ik}\Big(\frac{\partial u}{\partial x}\Big)_{x=x_k} = \sum_{k=1}^{N} a_{ik}\Big(\sum_{j=1}^{N} a_{kj} u(x_j, t)\Big) \\
&= \sum_{k=1}^{N}\sum_{j=1}^{N} a_{ik} a_{kj} u(x_j, t) = \sum_{j=1}^{N} b_{ij} u(x_j, t),\ i = 1, 2, 3, ..., N. \\
\frac{\partial^2 u_i}{\partial y^2} &= \frac{\partial}{\partial y}\Big(\frac{\partial u}{\partial y}\Big) = \sum_{k=1}^{N} a'_{ik}\Big(\frac{\partial u}{\partial y}\Big)_{y=y_k} = \sum_{k=1}^{N} a'_{ik}\Big(\sum_{j=1}^{N} a'_{kj} u(y_j, t)\Big) \\
&= \sum_{k=1}^{N}\sum_{j=1}^{N} a'_{ik} a'_{kj} u(y_j, t) = \sum_{j=1}^{N} b'_{ij} u(y_j, t),\ i = 1, 2, 3, ..., N.
\end{aligned}
\tag{1.19}
$$

2. Third order derivative can be obtained as follows:

$$\begin{aligned}
\frac{\partial^3 u_i}{\partial x^3} &= \frac{\partial}{\partial x}\Big(\frac{\partial^2 u}{\partial x^2}\Big) = \sum_{k=1}^{N} a_{ik}\Big(\frac{\partial^2 u}{\partial x^2}\Big)_{x=x_k} = \sum_{k=1}^{N} a_{ik}\Big(\sum_{j=1}^{N} b_{kj}u(x_j,t)\Big) \\
&= \sum_{k=1}^{N}\sum_{j=1}^{N} a_{ik}b_{kj}u(x_j,t) = \sum_{j=1}^{N} c_{ij}u(x_j,t),\ i=1,2,3,...,N. \\
\frac{\partial^3 u_i}{\partial y^3} &= \frac{\partial}{\partial y}\Big(\frac{\partial^2 u}{\partial y^2}\Big) = \sum_{k=1}^{N} a_{ik}^{'}\Big(\frac{\partial^2 u}{\partial y^2}\Big)_{y=y_k} = \sum_{k=1}^{N} a_{ik}^{'}\Big(\sum_{j=1}^{N} b_{kj}^{'}u(y_j,t)\Big) \\
&= \sum_{k=1}^{N}\sum_{j=1}^{N} a_{ik}^{'}b_{kj}u(y_j,t) = \sum_{j=1}^{N} c_{ij}^{'}u(y_j,t),\ i=1,2,3,...,N.
\end{aligned} \tag{1.20}$$

3. Fourth order derivative can be obtained as follows:

$$\begin{aligned}
\frac{\partial^4 u_i}{\partial x^4} &= \frac{\partial}{\partial x}\Big(\frac{\partial^3 u}{\partial x^3}\Big) = \sum_{k=1}^{N} a_{ik}\Big(\frac{\partial^3 u}{\partial x^3}\Big)_{x=x_k} = \sum_{k=1}^{N} a_{ik}\Big(\sum_{j=1}^{N} c_{kj}u(x_j,t)\Big) \\
&= \sum_{k=1}^{N}\sum_{j=1}^{N} a_{ik}c_{kj}u(x_j,t) = \sum_{j=1}^{N} d_{ij}u(x_j,t),\ i=1,2,3,...,N. \\
\frac{\partial^4 u_i}{\partial y^4} &= \frac{\partial}{\partial y}\Big(\frac{\partial^3 u}{\partial y^3}\Big) = \sum_{k=1}^{N} a_{ik}^{'}\Big(\frac{\partial^3 u}{\partial y^3}\Big)_{y=y_k} = \sum_{k=1}^{N} a_{ik}^{'}\Big(\sum_{j=1}^{N} c_{kj}^{'}u(y_j,t)\Big) \\
&= \sum_{k=1}^{N}\sum_{j=1}^{N} a_{ik}^{'}c_{kj}u(y_j,t) = \sum_{j=1}^{N} d_{ij}^{'}u(y_j,t),\ i=1,2,3,...,N.
\end{aligned} \tag{1.21}$$

Another approach to find out these weighting coefficients is given by Shu [39] for polynomial basis function and for non polynomial basis functions differently as follows:

For polynomial basis function the general formula is given as:

$$a_{ij}^{m} = m(a_{ij}^{1}a_{ij}^{m-1} - \frac{a_{ij}^{m-1}}{x_i - x_j}),\ for\ i \neq j,\ and\ a_{ii}^{m} = -\sum_{i=1, i\neq j}^{N} a_{ij}^{m}.$$

Here a_{ij}^{m} represents b_{ij}, c_{ij}, and d_{ij} for $m = 2, 3, 4$ respectively. Similar formula can be used to calculate $b_{ij}^{'}$, $c_{ij}^{'}$, and $d_{ij}^{'}$ if we consider $a_{ij}^{'}$ in place of a_{ij}.

If the basis function is non polynomial the following expression can be used to calculate the weighting coefficients:

for $i \neq j$

$$b_{ij} = a_{ij}(\frac{2sin(x_i)}{(cos(x_i - x_j))}) + 2a_{ii} + cot(x_i)),$$

and for $i = j$

$$b_{ij} = c_i^{(2)} + 2cot(x_i)c_i^{(1)}(x_i) - 1$$

where $c_i^{(2)}$ and $c_i^{(1)}$ are as follows:

$$c_j^{(1)}(x_i) = \frac{a_{ij}}{\frac{sin(x_i)}{sin(x_j)}}$$

and

$$c_j^{(2)}(x_i) = \frac{b_{ij} - 2cot(x_i)a_{ij}}{\frac{sin(x_i)}{sin(x_j)}}.$$

In the present research work, while computing the numerical results for different test problems for concerned differential equations we have used the both above mentioned approaches. For quintic trigonometric basis functions, we have obtained better results using the first technique while for cubic trigonometric basis functions last two techniques are found to give better results.

1.5 Strong stability-preserving time-stepping Runge-Kutta (SSP-RK43) scheme

After substituting the approximations for all space derivatives in the concerned time dependent partial differential equation, a system of ordinary differential equations is obtained(say):

$$u_t = R(u).$$

This system can be solved by any appropriate numerical scheme. In the present work to solve this system we use strong stability-preserving time-stepping Runge-Kutta (SSP-RK43) scheme [72] which is as follows:

$$u^{(1)} = u^{(m)} + \frac{\Delta t}{2}R(u^{(m)})$$

$$u^{(2)} = u^{(1)} + \frac{\Delta t}{2}R(u^{(1)})$$

$$u^{(3)} = \frac{2}{3}u^{(m)} + \frac{u^{(2)}}{3} + \frac{\Delta t}{6}R(u^{(2)})$$

$$u^{(m+1)} = u^{(3)} + \frac{\Delta t}{2} R(u^{(3)})$$

1.6 Stability of the numerical schemes

To check the stability of the numerical schemes we have used the eigenvalues approach for both the proposed methods which is as follows:

1. In the case of collocation method after discretizing the considered differential equation and substituting the approximation values from (1.9), we get the system which can be written in the following form:

$$AC^{n+1} = BC^n + F(u) \tag{1.22}$$

or

$$C^{n+1} = MC^n + b_m \tag{1.23}$$

where b_m is a vector containing known boundary values and $M = A^{-1}B$ is the iterative matrix whose eigenvalues are λ_m for all $m = N$, where N is the dimension of the matrix M. The condition for stability of the obtained system depends on the eigenvalues and given as:

$$max\ |\lambda| < 1$$

.

2. On the other hand, applying differential quadrature method to any partial differential equation, we obtain the system of ordinary differential equations which further can be solved by any numerical method. In this research work, we have used Runge-Kutta method to solve the obtained system of ordinary differential equations. In the book by M. K. Jain [73] conditions are given for eigenvalues to ensure whether any system solved by Runge-Kutta method is stable or not. On applying the differential quadrature method to any partial differential equation we can write the equation in the form:

$$u_t(x,t) = B + f(u(x,t)). \tag{1.24}$$

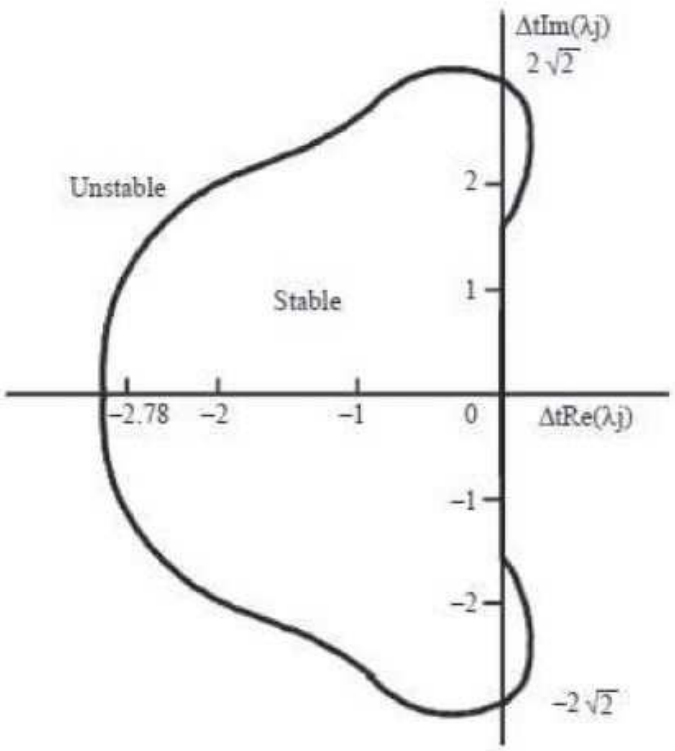

FIGURE 1.1: Stability region when the obtained system has complex eigenvalues

where B is a matrix obtained from the concerned partial differential equation and f(u(x,t)) is nonlinear terms in that equation. The stability of the above system of equations depends upon the eigenvalues of the matrix B. The obtained system is stable if and only if the eigenvalues satisfy the following conditions:

(a) Real $\lambda_j : -2.78 < \Delta t\lambda_j < 0$.

(b) Pure imaginary $\lambda_j : -2\sqrt{2} < \Delta t\lambda_j < 2\sqrt{2}$.

(c) Complex $\lambda_j : \Delta t\lambda_j$ lie inside the region R (as shown in the Figure 1.1).

The above mentioned conditions are depending on the values of Δt. Hence the stability of the obtained system could vary based on the time steps. So one can choose the time steps accordingly to get the stable numerical solution for the considered problem.

1.7 Error norms and order of convergence

To validate the obtained numerical results it is very important to compare the obtained results with the exact solutions if present or with numerical results computed by the other numerical approaches in the literature. For this purpose different type of error norms are used in this research work. Some important formulas to find out numerical errors and order of convergence are as follows:

- L_2 error $=||u - U_N||_2 = \sqrt{h \sum_{j=0}^{N} |u_j - (U_N)_j|^2}$
- L_∞ error or Maximum error $= ||u - U_N||_\infty = \max_j |u_j - (U_N)_j|$
- RMS error $= \frac{1}{N+1} \sqrt{\sum_{j=0}^{N} |u_j - (U_N)_j|^2}$
- Average error $= \frac{1}{N} \sum_{j=0}^{N} |u_j - (U_N)_j|$
- GRE or GR error $= \frac{\sum_{j=0}^{N} |u_j - (U_N)_j|}{\sum_{j=0}^{N} |u_j|}$
- Order of convergence $= \frac{log(\frac{E(N)}{E(2N)})}{log(\frac{N}{2N})}$

where u and U_N represents the exact and numerical solutions respectively and N represents the number of partitions of the domain. $E(N)$ and $E(2N)$ are used to represent the errors calculated with N and $2N$ partitions of the domain.

Chapter 2

Numerical solution of Fisher's equation using cubic trigonometric collocation method

2.1 Introduction

The nonlinear partial differential equations exist in modelling of various phenomena existing in engineering and sciences. One of the most important nonlinear partial differential equation is Fisher's equation having well known applications in various biological and chemical processes, and in engineering such as gene propagation [74, 75], combustion [76], autocatalytic chemical reaction [77], and tissue engineering [78, 79]. This one dimensional nonlinear parabolic partial differential equation proposed by Fisher [74], is a reaction diffusion type of equation that examine the wave proliferation of a beneficial quality gene in a population. It also describes the kinetic advancing rate of an advantageous gene to portray the propagation of viral mutant in an infinitely long habitat. So the study of this type of partial differential equation becomes a relevant area of research.

Genetic Engineering is a new area of research in which different techniques have been applied to change the genetic makeup of cells with the purpose of making an organism better in some way. By changing the genome engineers enables to give desirable properties to different organisms. The genetic engineering of animals has increased significantly in recent years. The most exciting potential applications

of genetic modelling are concerned with animal breeding and in the treatment of genetic disorders. Fisher's equation plays an important role in describing the various perspective of genetic engineering. Most commonly it is used to describe the cell behaviour. It is one of the simplest reaction-diffusion equations, originally used to study the spread of a favored gene through a population.

Fisher's equation also serves as a good model to describe the extension of a vitro monolayer cell province in tissue engineering. It depicts the conduct of a cell population as a union of irregular cell movement and logistic multiplication, i.e. expansion up to a greatest cell thickness. It also predicts that, under specific conditions, the cell movement front takes the type of a travelling wave of fixed shape that proceed at a steady speed.

The Fisher's equation has the following form:

$$\frac{\partial u(x,t)}{\partial t} = \nu \frac{\partial^2 u(x,t)}{\partial x^2} + \rho f(u(x,t)),\ x \in (-\infty, \infty),\ t > 0 \tag{2.1}$$

where t and x are the time and spatial coordinate respectively. $\nu > 0$ represents the diffusion coefficient, $\rho > 0$ is the reaction factor and $f(u(x,t))$ is the nonlinear reaction term. $f(u(x,t))$ represents a nonlinear development rate and diminishes as u expands, and vanishes when $u = 1$. It compares the development of a populace u when there is a point of confinement $u = 1$ on the population's span the habitat can bolster. In the event, if $u > 1$ then $f(u) < 0$, i.e. the population diminishes when u is more noteworthy than limiting value. This interpretation suggests that the habitat can support a certain maximum population so that

$$0 \leq u(x,0) \leq 1 \ for\ x \in \mathbb{R} \tag{2.2}$$

Due to the existence of Fisher's equation in many contexts, the nonlinear term $f(u)$ is referred to different terminology based on its role. The term $f(u)$ alludes to a source or reaction term that articulate to the confinement-fatality process in a biological background. And if a chemically responding substance is diffusing through a medium, then its focus $u(x,t)$ fulfills mathematical equation (2.2), where $f(u)$ represent the rate of expansion of the substance because of the synthetic response. Different issues depicted by equation (2.1) incorporate the spread of creature or plant populace and the advancement of neutron populace in an atomic reactor, where $f(u)$ articulates to the net development rate.

Fisher's equation is extensively studied by many researchers in the past few years because of its vital use in many other fields. First of all pseudospectral approach has been applied to get the numerical solution of the Fisher's equation by Gazdag and Canosa [80] and by Olmos and Shizgal [81]. The moving mesh method is applied to it by Qiu and sloan [82]. In their paper, Wazwaz and Gorguis [83] used the Adomian decomposition scheme to find the exact solution to this equation. Parekh and Puri [84] and Twizell *et al.* [85] have offered the implicit and explicit finite difference method to thrash out the numerical study of Fisher's equation. Mittal and Jiwari [86] used the differential quadrature method to solve this equation. Tang and weber [87] anticipated a Galerkin finite element scheme and Al-Khaled [88] projected the scheme known as sinc collocation to solve this equation. Abdusalam and Fahmy [89] found the exact solution of generalized telegraph Fisher's equation. Mickens [90] offered a best finite difference scheme for Fisher's equation. The wavelet Galerkin method has been used by Mittal [91] and El-Azab [92]. The B-spline Galerkin approach to find the approximate solution of this equation have been used by Sahin *et al.* [93]. Mittal and Arora [94] used a B-spline collocation method to solve Fisher's equation. Haar wavelet method for this equation has been used by Hariharan *et al.* [95]. Aghamohamadi *et al.* [96] used tension spline to solve nonlinear Fisher's equation. Recently fractional derivatives and dynamical systems approach has been used to solve Fisher's equation by Alquran *et al.* [97] and Faye [98] respectively.

In this chapter, we used collocation method with a trigonometric B-spline basis to solve the various type of Fisher's equations. The altered primary and boundary conditions for the equation (2.1) are prearranged as:

$$u(x,0) = u_0(x) \in [0,1],\ x \in (-\infty, \infty), \tag{2.3}$$

$$\lim_{x\to-\infty} u(x,t) = 1,\ \lim_{x\to\infty} u(x,t) = 0. \tag{2.4}$$

$$\lim_{x\to\pm\infty} u(x,t) = 0. \tag{2.5}$$

To implement the numerical scheme to equation (2.1), the unbounded substantial domain can be replaced with a limited working out domain as $[a, b]$. The target of our employment is to explore the practicality of trigonometric B-spline collocation scheme for the numerical solution of Fisher's equation, with nonlocal boundary conditions. Additionally, some numerical illustrations with neighborhood limit

conditions and another with distinctive boundary conditions are comprehended. The outcomes were observed to be empowering.

2.2 Solution of Fisher's equation

The estimated solution $U_m(x)$ for the equation (2.1), can be attained by considering the limited domain $[a, b]$ with beginning condition $u(x, 0) = u_0(x)$, and limit condition $u(a, t) = g_0,\ u(b, t) = g_1$. On discretizing the time derivative in normal finite difference way, we have:

$$\frac{U_m^{n+1} - U_m^n}{\Delta t} = \nu\Big(\frac{(U_{xx})_m^n + (U_{xx})_m^{n+1}}{2}\Big) + \rho\frac{(f(U_m))^n + (f(U_m))^{n+1}}{2}. \tag{2.6}$$

On simplifying and linearizing the nonlinear terms in the equation by using the quasilinearization formula:

$$X^{n+1}(u) = X^n(u) + (u^{n+1} - u^n)(X_u)^n$$

we get

$$U_m^{n+1}q - \frac{\nu}{2}(U_{xx})_m^{n+1} = R \tag{2.7}$$

where $q = (\frac{1}{\Delta t} - \rho\frac{1}{2} + \rho U_m^n)$ and $R = (\frac{U_m^n}{\Delta t} + \nu\frac{(U_{xx})_m^n}{2} + \rho\frac{U_m^n}{2})$. On substituting the values of U_m and its derivatives from the equation (1.9) in (2.7) and simplifying we obtain the system:

$$C_m^{n+1}(qa_1 - \frac{\nu}{2}a_5) + C_{m+1}^{n+1}(qa_2 - \frac{\nu}{2}a_6) + C_{m+2}^{n+1}(qa_1 - \frac{\nu}{2}a_5) = R. \tag{2.8}$$

This system has $(N+1)$ linear equations in $(N+3)$ unknowns $d^n = \{C_{-1}^n, C_0^n, C_1^n, ..., C_N^n, C_{N+1}^n\}$. To get an inimitable solution for this framework two extra restrictions are required, these are acquired from the boundary conditions $u(a, t) = g_0$ and $u(b, t) = g_1$. The obligation of boundary conditions empowers us to wipe out the parameters C_{-1} and C_{N+1} from framework (2.8) and now it diminishes to a $(N + 1) \times (N + 1)$ matrix system, given by:

$$A\vec{C} = \vec{B} \tag{2.9}$$

where A matrix is given as:

$$\begin{bmatrix} (\frac{a_2 a_5 \nu}{2a_1}) - \frac{a_6 \nu}{2} & 0 & 0 & 0 & 0 & 0 \\ (qa_1 - \frac{a_5 \nu}{2}) & (qa_2 - \frac{a_6 \nu}{2}) & (qa_1 - \frac{a_5 \nu}{2}) & . & . & 0 \\ 0 & (qa_1 - \frac{a_5 \nu}{2}) & (qa_2 - \frac{a_6 \nu}{2}) & (qa_1 - \frac{a_5 \nu}{2}) & 0 & 0 \\ 0 & 0 & (qa_1 - \frac{a_5 \nu}{2}) & (qa_2 - \frac{a_6 \nu}{2}) & (qa_1 - \frac{a_5 \nu}{2}) & 0 \\ 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & (\frac{a_2 a_5 \nu}{2a_1}) - \frac{a_6 \nu}{2} \end{bmatrix} \tag{2.10}$$

$\vec{C} = [C_0, C_1, ..., C_{m-1}, C_m]^T$ and the value of $\vec{B} = [R - \frac{g_0}{a_1}(qa_1 - \frac{\nu a_5}{2}), R, ..., R, R - \frac{g_1}{a_1}(qa_1 - \frac{\nu a_5}{2})]^T$ which can be solved by using programming in MATLAB. Once we have the values of unknowns, we can obtain the numerical solution using the following formulas:

$$\begin{aligned} U_m(x) &= a_1 C_{m-1} + a_2 C_m + a_1 C_{m+1} \\ U'_m(x) &= a_3 C_{m-1} + a_4 C_{m+1} \\ U''_m(x) &= a_5 C_{m-1} + a_6 C_m + a_5 C_{m+1} \end{aligned} \tag{2.11}$$

where the first and second derivatives with respect to x of $U(x)$ are denoted by $U'(x)$ and $U''(x)$. The time advancement of the numerical solution $U_N(x,t)$ is computed by the vector $\vec{C}$, which is discovered over and over by the recurrence relation. To start the process of recurrence we need to use the initial values named as C^0, which can be obtained using the process discussed in the next section.

2.3 The initial vector C^0

The introductory vector C^0 can be acquired from the beginning conditions and boundary values of the derivatives subsidiaries conditions as the accompanying expressions:

$$U_x(a,0) = u_x(x_0,0) = g_0$$

$$U(x_m, 0) = u_0(x_m, 0)\ m = 1, 2, ..., N-1.$$

$$U_x(b, 0) = u_x(x_N, 0) = g_1$$

This will generate the $(N+1) \times (N+1)$ system of the equations in the following form:

$$\begin{bmatrix} a_2 & a_1 - \frac{a_1 a_4}{a_3} & 0 & 0 & 0 & 0 \\ a_1 & a_2 & a_1 & . & . & 0 \\ 0 & a_1 & a_2 & a_1 & 0 & 0 \\ 0 & 0 & a_1 & a_2 & a_1 & 0 \\ 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & a_1 - \frac{a_1 a_3}{a_4} & a_2 \end{bmatrix} \begin{bmatrix} C_0^0 \\ C_1^0 \\ . \\ . \\ C_{m-1}^0 \\ C_m^0 \end{bmatrix} = \begin{bmatrix} g_0 - \frac{a_1}{a_3} u_x(x_0, 0) \\ u(x_1, 0) \\ . \\ . \\ u(x_N, 0) \\ g_1 - \frac{a_1}{a_4} u_x(x_N, 0) \end{bmatrix} \tag{2.12}$$

this system can be solved using programming on MATLAB, to find the initial values C_i^0 for $i = 0, 1, 2, ..., m$ which are further used to calculate the numerical value of $u(x, t)$.

2.4 Stability of the numerical scheme

The stability of the given scheme is discussed using matrix method. To implement the method, on discretizing differential equation (2.1), we get:

$$\frac{U_m^{n+1} - U_m^n}{k} = \nu \frac{(U_{xx})_m^{n+1} + (U_{xx})_m^n}{2} + F(u). \tag{2.13}$$

where $F(u)$ is a nonlinear term, and the above equation further can be written as:

$$\frac{U_m^{n+1}}{k} - \nu \frac{(U_{xx})_m^{n+1}}{2} = \frac{U_m^n}{k} + \nu \frac{(U_{xx})_m^n}{2} + F(u). \tag{2.14}$$

On substituting the approximation values from equation (2.11) we get

$$\begin{aligned} C_{m-1}^{n+1}(\frac{a_1}{k}-\frac{a_5\nu}{2})+C_m^{n+1}(\frac{a_2}{k}-\frac{a_6\nu}{2})+C_{m+1}^{n+1}(\frac{a_1}{k}-\frac{a_5\nu}{2})= \\ C_{m-1}^{n}(\frac{a_1}{k}+\frac{a_5\nu}{2})+C_m^{n}(\frac{a_2}{k}+\frac{a_6\nu}{2})+C_{m+1}^{n}(\frac{a_1}{k}+\frac{a_5\nu}{2})+F(u). \end{aligned} \tag{2.15}$$

taking $m = 0, 1, 2, ..., N+1$. The above system can be written in the following matrix form:

$$AC^{n+1} = BC^n + F(u) \tag{2.16}$$

or

$$C^{n+1} = MC^n + b_m \tag{2.17}$$

where b_m is a vector containing known boundary values and the iterative matrix $M = A^{-1}B$ whose eigenvalues are λ_m for all $m = N$, where N is the dimension of the matrix M. The condition for stability on the eigenvalues is $max\ |\lambda| < 1$ which is demonstrated with help of Figures 2.1, 2.2 and 2.3 for different grid points. Hence the given scheme is unconditionally stable.

2.5 Convergence analysis

Let $u(x)$ be the exact solution of given equation (2.1) with the initial condition (2.3) and the boundary condition (2.4) and

$$U(x) = \sum_{k=-1}^{N+1} C_k(t) TB_k(x)$$

be the trigonometric B-spline collocation approximation to $u(x)$. Due to round off error in computations we assume that $\hat{U}(x)$ be the computed spline for $U(x)$ so that

$$\hat{U}(x) = \sum_{k=-1}^{N+1} \hat{C}_k(t) TB_k(x)$$

where $\hat{C} = (\hat{C}_{-1}, \hat{C}_0, \hat{C}_1, ..., \hat{C}_{N+1})$. To calculate approximately the error $\|U(x) - u(x)\|_\infty$, we must assess the error $\|U(x) - \hat{u}(x)\|_\infty$ and $\|\hat{U}(x) - U(x)\|_\infty$ separately. Following we have

$$AC = r \tag{2.18}$$

$$A\hat{C} = \hat{r} \tag{2.19}$$

subtracting (2.18) and (2.19) we get

$$A(C - \hat{C}) = (r - \hat{r}) \tag{2.20}$$

Now to prove the final result we need some theorems and lemmas as follows:

Theorem 2.1. *Suppose that* $f(x) \in C^4[0,1]$ *and* $\|f^4(x)\| \leq L \ \forall x \in [0,1]$ *and* $\Delta = 0 = x_0 < x_1 ... < x_N = 1$ *be equality spaced partition of [0,1] with step size* h. *If* $S_\Delta(x)$ *be the unique spline function interpolate* $f(x)$ *at nodes* $x_0, x_1, ..., x_N \in \Delta$, *then there exist a constant* $\lambda_j \leq 2$ *such that* $\forall x \in [0,1]$

$$\|f^j(x) - S^j_\Delta(x)\| \leq \lambda_j L h^{4-j}$$

where $j = 0, 1, 2, 3$ *and* $\|.\|$ *represent the* ∞*-norm. [99]*

Now we want to find a bound for $\|r - \hat{r}\|_\infty$

$$|r(x_j) - \hat{r}(x_j)| = |r(x_j, U(x_j), U'_j(x_j), U''_j(x_j)) - \hat{r}(x_j, \hat{U}(x_j), \hat{U}'_j(x_j), \hat{U}''_j(x_j))|$$

$$\|r - \hat{r}\|_\infty \leq M(|U(x) - \hat{U}(x)| + |U\prime(x) - \hat{U}'(x)| + |U''(x) - \hat{U}''(x)|)$$

$$\|r - \hat{r}\|_\infty \leq ML\lambda_0 h^4 + ML\lambda_1 h^3 + ML\lambda_2 h^2.$$

Thus we can write

$$\|r - \hat{r}\|_\infty \leq M_1 h^2$$

where $M_1 = ML\lambda_0 h^2 + ML\lambda_1 h^1 + ML\lambda_2$
now from the equation (2.20) we have

$$(C - \hat{C}) = A^{-1}(r - \hat{r})$$

taking norm on both sides we have

$$\|C - \hat{C}\| = A^{-1}(r - \hat{r}) \leq \|A^{-1}\|\|r - \hat{r}\| \leq M_2 h^2$$

where $M_2 = M_1\|A^{-1}\|$

Lemma 2.2. *The trigonometric B-spline $TB_{-1}, TB_0, ...TB_{N+1}$ satisfies the following inequality*

$$|\sum_{j=-1}^{N+1} TB_j(x)| \leq \frac{8}{3}$$

Proof:

$$\sum TB_j(x) = \phi_{j-1}(x) + \phi_j(x) + \phi_{j+1}(x)$$

$$= a_1 + a_2 + a_1$$

$$|\sum TB_j(x)| \leq |a_1| + |a_2| + |a_1|$$

$$\leq 1 + \frac{2}{3} + 1 = 8/3.$$

Now from $U(x) - \hat{U}(x) = \sum_{j=-1}^{N+1}(c_j - \hat{c}_j)TB_j(x)$ we have

$$\|U(x)-\hat{U}(x)\|_\infty = \|\sum_{j=-1}^{N+1}(c_j-\hat{c}_j)TB_j(x)\| \leq \|c_j-\hat{c}_j\|_\infty|\sum_{j=-1}^{N+1}(c_j-\hat{c}_j)TB_j(x)| \leq \frac{8}{3}M_2h^2$$

$$\|u(x)-U(x)\| \leq \|u(x)-\hat{U}(x)\|+\|\hat{U}(x)-U(x)\| \leq \lambda_0 Lh^4+\frac{8}{3}M_2h^2 = (\lambda_0 Lh^2+\frac{8}{3}M_2)h^2$$

$$\|u(x) - U(x)\| \leq wh^2$$

where $w = \lambda_0 Lh^2 + \frac{8}{3}M_2$

$$\|u(x,t^n) - U(x,t^n)\| = \rho(k + h^2)$$

where ρ is some constant. Thus the order of the convergence of collocation method is $o(k + h^2)$

2.6 Numerical experiment and discussion

In this section we have simulated three problems of Fisher's equation with distinctive initial conditions and examined the numerical results.

Example 2.1. *Consider the modified form of Fisher's equation given by:*

$$\frac{\partial u}{\partial t} = \nu\frac{\partial^2 u}{\partial x^2}(x,t) + \rho u(1-u)(u-d) \text{ where } 0 < d < 1 \tag{2.21}$$

which is also known as FitzHugh-Nagumo equation. A particular solution of the equation (2.21) was found

$$u(x,t) = \frac{1}{2}(1+d) + \frac{1}{2}(1-d)\ tanh\Big(\frac{(1-d)}{2\sqrt{2}}x + \frac{(1-d^2)}{2\sqrt{2}}t\Big) \tag{2.22}$$

The numerical solution of the equation (2.21) with boundary conditions:

$$\lim_{x\to-\infty} u(x,t) = 0.2,\ \lim_{x\to\infty} u(x,t) = 1.$$

taking $[a, b] = [-22, 22]$ *has been obtained at* $\rho = 1$ *with* $\triangle t = 0.0001$*, for different number of partitions* $N = 41$*,* 410 *and* 1000*. The obtained results are found in good agreement with the exact solution of the problem. The results are characterized graphically so that we can compare the results with the exact solution. Figure 2.4 shows exact and numerical solution at different time levels. To compare the numerical solution with exact solution for* $\rho = 1$ *and* $d = 0.2$ *results are portrayed in Table 2.1 and* L_2 *and* L_∞ *errors are calculated and presented in Table 2.5.*

Example 2.2. *Consider the equation (2.1) with* $\nu = 1$*,* $\rho = 1$ *and* $f(u(x,t)) = u^2(1-u)$*, which is Fisher-type equation, subject to initial condition:*

$$u(x,0) = \frac{1}{1 + exp(\frac{x}{\sqrt{2}})} \tag{2.23}$$

and the boundary conditions given by

$$\lim_{x \to -\infty} u(x,t) = 1, \ \lim_{x \to \infty} u(x,t) = 0.$$

The exact solution of this equation is given by [100]:

$$u(x,t) = \frac{1}{1 + exp(v(x - vt))}, \ where \ v = \frac{1}{\sqrt{2}} \tag{2.24}$$

The numerical result is obtained for the domain $[-20, 20]$ *at different time levels with* $\triangle t = 0.01$ *and* $N = 360$*. The obtained results are presented in Table 2.2. In Figure 2.5 the exact and the numerical solution are presented graphically at different times. The obtained* L_2 *and* L_∞ *errors are also presented in Table 2.6.*

Example 2.3. *Consider the equation (2.1) with* $f(u) = \rho u(1-u)$ *and* $\nu = 1$ *in the domain* $[0, 1]$*. For this equation initial condition has been considered as:*

$$u(x,0) = \frac{1}{(1 + exp(\rho x))^2} \tag{2.25}$$

with boundary conditions given by (2.4) and the exact solution in closed form [95] given as:

$$u(x,t) = \frac{1}{[1 + exp((\sqrt{\frac{\rho}{6}})x - (\frac{5\rho \ t}{6}))]^2} \tag{2.26}$$

For $\rho = 1$*, the numerical solution computed for* $\triangle t = 0.00001$ *and* $N = 410$ *at different time levels are presented in Table 2.3. The obtained* L_2 *and* L_∞ *errors*

are depicted in Table 2.7. The comparison of numerical and exact solution is presented graphically by Figures 2.6 and 2.7.
For $\rho = 6$*, the numerical solution computed for* $\triangle t = 0.000001$ *and* $N = 410$ *at different time levels are presented in Table 2.4. The obtained* L_2 *and* L_∞ *errors are depicted in Table 2.8. The comparison of the numerical and the exact solution is presented graphically by the Figures 2.8 and 2.9.*

2.7 Summary

Due to numerous application in chemical and biological processes Fisher's equation turn into a focal point of fascination for researchers in recent years and solved by distinctive methods. In this chapter, the trigonometric cubic B-spline basis function has been used to solve Fisher's equation with collocation method. The stability and convergence has been proved using theoretical results. The approximated results are compared with the exact solutions by obtaining the absolute, L_2 and L_∞ errors. It is evident from the obtained errors that the numerical results are in good conformity with the exact solutions. It can be concluded that the method is an economical and efficient technique for finding numerical solutions for a variety of linear and nonlinear physical models.

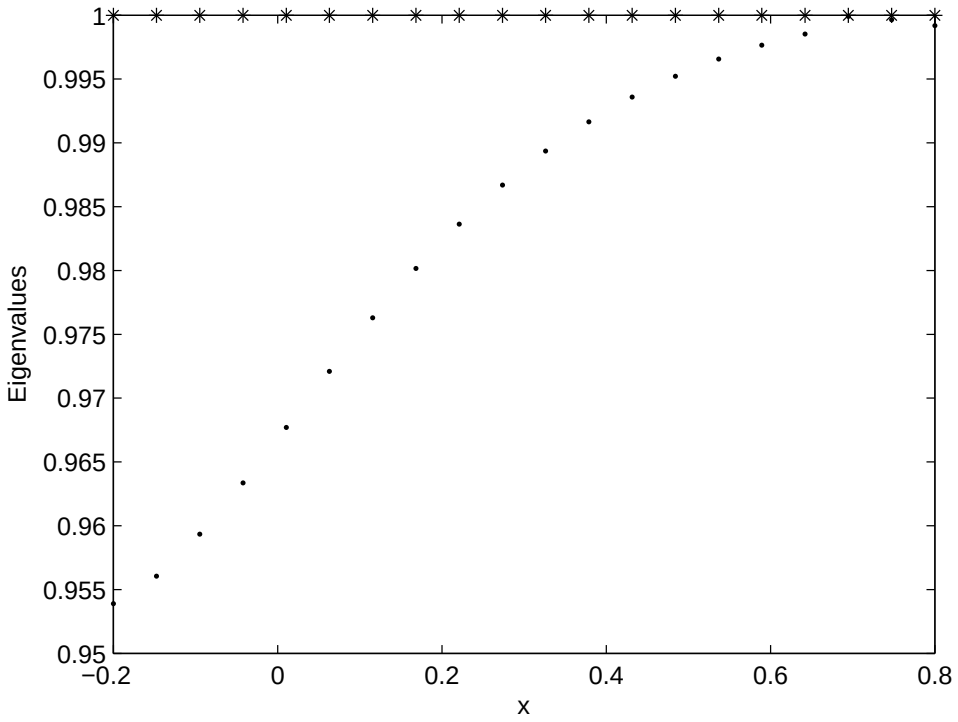

FIGURE 2.1: Eigenvalues of matrix M with 20 grid points

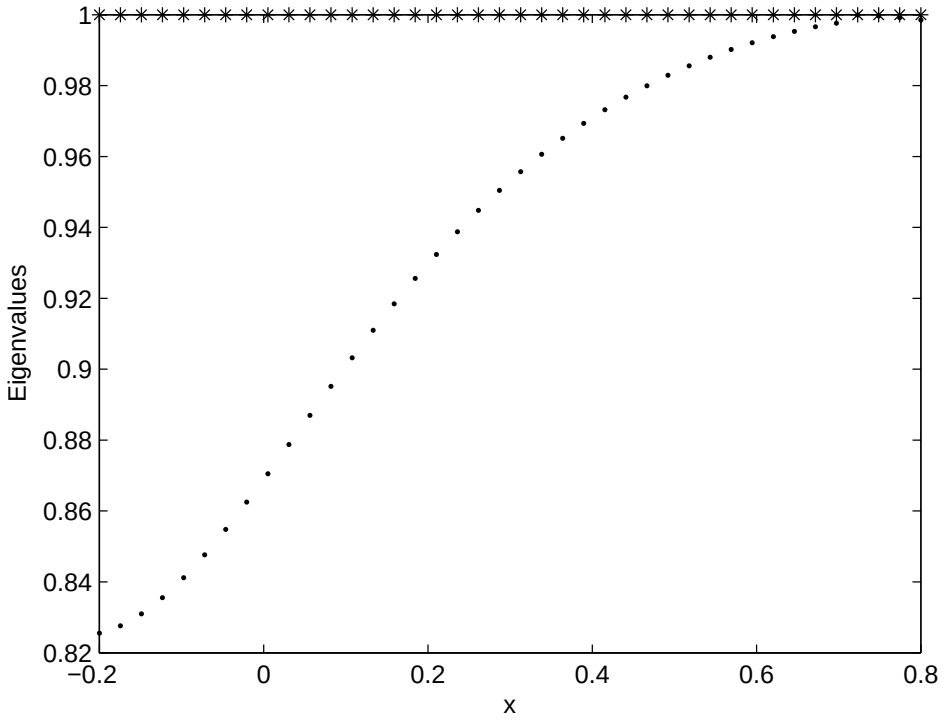

FIGURE 2.2: Eigenvalues of matrix M with 40 grid points

TABLE 2.1: Absolute errors in the present numerical solution of example 2.1 with $\rho = 1$, $a = 0.2$ and $\triangle t = 0.0001$ at different time levels

	Time=0.001			Time=0.002			Time=0.003		
x	Num	Exact	Abs. Error	Num	Exact	Abs. Error	Num	Exact	Abs. Error
-22	0.2000031	0.2000031	2.13875E-9	0.2000031	0.2000031	4.27896E-9	0.2000031	0.2000031	6.42062E-9
-14	0.2003024	0.2003022	1.68083E-7	0.2003026	0.2003022	3.36253E-7	0.2003028	0.2003023	5.0451E-7
-6	0.2264218	0.2264166	5.18408E-6	0.2264391	0.2264287	1.03737E-5	0.2264565	0.2264409	1.55688E-5
2	0.3960443	0.3960148	2.95696E-5	0.3961448	0.3960857	5.91566E-5	0.3962453	0.3961566	8.87612E-5
10	0.9973073	0.9973063	1.07392E-6	0.9973092	0.9973070	2.14679E-6	0.9973110	0.9973078	3.21863E-6
18	0.9999700	0.9999695	5.45852E-7	0.9999701	0.9999690	1.09127E-6	0.9999701	0.9999684	1.63624E-6
22	0.9999968	0.9999968	2.1373E-9	0.9999968	0.9999968	4.27316E-9	0.9999968	0.9999968	6.40756E-9

TABLE 2.2: Absolute errors in the solution of example 2.2 for $N = 360$ and $\triangle t = 0.01$ at different time levels

	Time= 2			Time=3			Time=4		
x	Num	Exact	Abs. Error	Num	Exact	Abs. Error	Num	Exact	Abs. Error
-4	0.9775821	0.9780525	4.70E-4	0.9860122	0.9865722	5.60E-4	0.9912214	0.9918123	5.90E-4
0	0.7235071	0.7232432	2.63E-4	0.8114322	0.8116254	1.93E-4	0.8761190	0.8765986	4.79E-4
4	0.1428995	0.1422273	6.72E-4	0.2155247	0.2146850	8.39E-4	0.3115492	0.3106863	8.62E-4
8	0.0096899	0.0096297	6.02E-5	0.0158838	0.0157782	1.05E-4	0.0259265	0.0257503	1.76E-4
12	0.0005734	0.0005698	3.61E-6	0.0009457	0.0009392	6.57E-6	0.0015593	0.0015475	1.18E-5
13	0.0002822	0.0002805	1.78E-6	0.0004656	0.0004624	3.24E-6	0.0007679	0.0007621	5.82E-6
14	0.0001389	0.0001380	8.75E-7	0.0002291	0.0002276	1.59E-6	0.0003780	0.0003752	2.86E-6
15	6.84E-5	6.79E-5	4.29E-7	0.0001127	0.00011201	7.70E-7	0.0001860	0.0001846	1.36E-6
16	3.36E-5	3.34E-5	2.02E-7	5.55E-5	5.51E-5	3.36E-7	9.14E-5	9.09E-5	5.54E-7
17	1.65E-5	1.65E-5	6.18E-8	2.72E-5	2.71E-5	3.76E-8	4.47E-5	4.47E-5	2.73E-8
18	8.00E-6	8.10E-6	9.95E-8	1.30E-5	1.33E-5	3.27E-7	2.13E-5	2.20E-5	7.34E-7
19	3.55E-6	3.98E-6	4.37E-7	5.55E-6	6.57E-6	1.02E-6	8.82E-6	1.08E-5	2.01E-6

TABLE 2.3: Absolute errors in the solution of example 2.2 with $\rho = 1$, $N = 410$ and $\triangle t = 0.00001$ at different time levels

	Time= 0.001			Time=0.002			Time=0.003		
x	Num	Exact	Abs. Error	Num	Exact	Abs. Error	Num	Exact	Abs. Error
0.2	0.2301669	0.2301668	1.09E-7	0.2303666	0.2303664	1.29E-7	0.2305653	0.2305661	8.04E-7
0.3	0.2204797	0.2204796	1.13E-7	0.2206748	0.2206745	2.26E-7	0.2208699	0.2208696	3.32E-7
0.4	0.2110240	0.2110239	1.16E-7	0.2112143	0.2112141	2.31E-7	0.2114047	0.2114044	3.47E-7
0.5	0.2018060	0.2018059	1.18E-7	0.2019914	0.2019912	2.37E-7	0.2021769	0.2021766	3.55E-7
0.6	0.1928307	0.1928306	1.20E-7	0.1930111	0.1930109	2.41E-7	0.1931917	0.1931913	3.61E-7
0.7	0.1843126	0.1843125	1.22E-7	0.1844881	0.1844878	2.44E-7	0.1846637	0.1846633	3.65E-7
0.8	0.1762373	0.1762371	1.24E-7	0.1764078	0.1764076	2.37E-7	0.1765781	0.1765782	5.14E-8

TABLE 2.4: Absolute errors in the solution of example 2.2 with $\rho = 6$, $N = 410$ and $\triangle t = 0.000001$ at different time levels

	Time= 0.0001			Time=0.0002			Time=0.0003		
x	Num	Exact	Abs. Error	Num	Exact	Abs. Error	Num	Exact	Abs. Error
0.1	0.22570537	0.22570531	6.67E-8	0.2258239	0.2258238	1.33E-7	0.2259425	0.2259423	1.99E-7
0.2	0.20265195	0.20265188	7.09E-8	0.2027634	0.2027633	1.42E-7	0.2028750	0.2028748	2.13E-7
0.3	0.18105066	0.18105059	7.37E-8	0.1811547	0.1811546	1.47E-7	0.1812589	0.1812586	2.21E-7
0.4	0.16095951	0.16095943	7.52E-8	0.1610559	0.1610558	1.50E-7	0.1611525	0.1611522	2.26E-7
0.5	0.14240886	0.14240879	7.54E-8	0.1424976	0.1424974	1.51E-7	0.1425864	0.1425862	2.26E-7
0.6	0.12540281	0.12540274	7.44E-8	0.1254839	0.1254837	1.49E-7	0.1255650	0.1255648	2.23E-7
0.7	0.10992128	0.10992121	7.23E-8	0.1099948	0.1099947	1.45E-7	0.1100684	0.1100682	2.17E-7
0.8	0.09592292	0.09592285	6.94E-8	0.0959892	0.0959890	1.39E-7	0.0960555	0.0960553	2.08E-7
0.9	0.08334843	0.08334836	6.58E-8	0.0834078	0.0834076	1.32E-7	0.0834672	0.0834670	1.95E-7

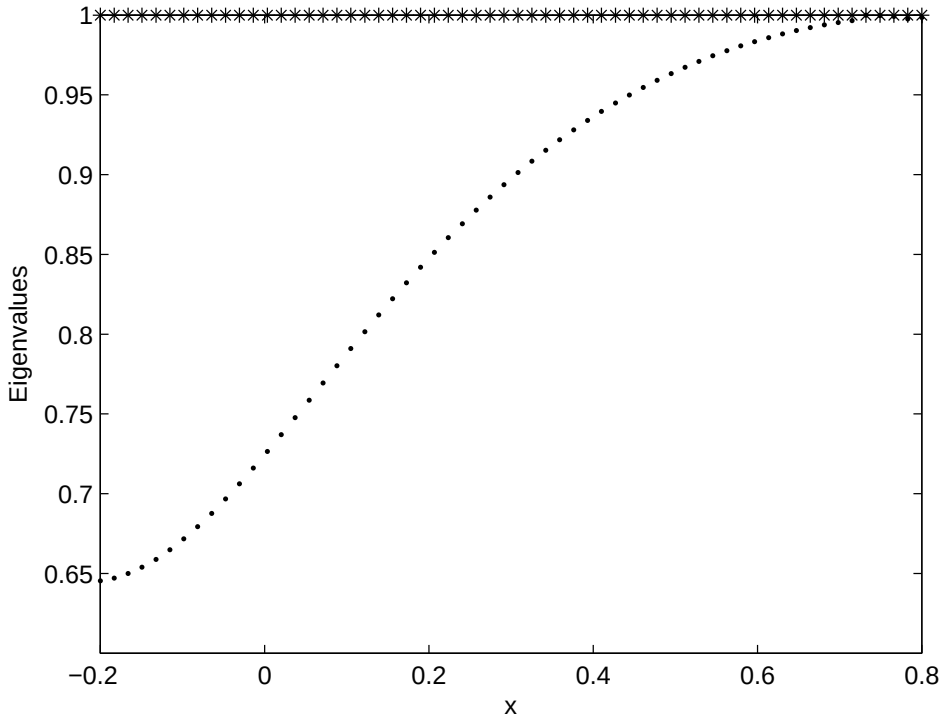

FIGURE 2.3: Eigenvalues of matrix M with 60 grid points

TABLE 2.5: L_2 and L_∞ errors in the solution of example 2.1 with $\rho = 2000$ and $\triangle t = 0.00001$ at different time levels

Time	0.001	0.0015	0.002	0.0025	0.003	0.0035	0.004
L_2	1.8721E-5	2.8082E-5	3.7442E-5	4.6801E-5	5.6161E-5	6.5520E-5	7.4879E-5
L_∞	4.0085E-5	6.0128E-5	8.0170E-5	1.0021E-4	1.2025E-4	1.4030E-4	1.6034E-4

TABLE 2.6: L_2 and L_∞ errors in the solution of example 2.2 with $N = 360$ and $\triangle t = 0.01$ at different time levels

Time	2	3	4	5	6
L_2	5.9148E-4	6.1803E-4	6.2957E-4	6.3419E-4	6.3575E-4
L_∞	9.7531E-4	9.1280E-4	8.6543E-4	8.2610E-4	7.9274E-4

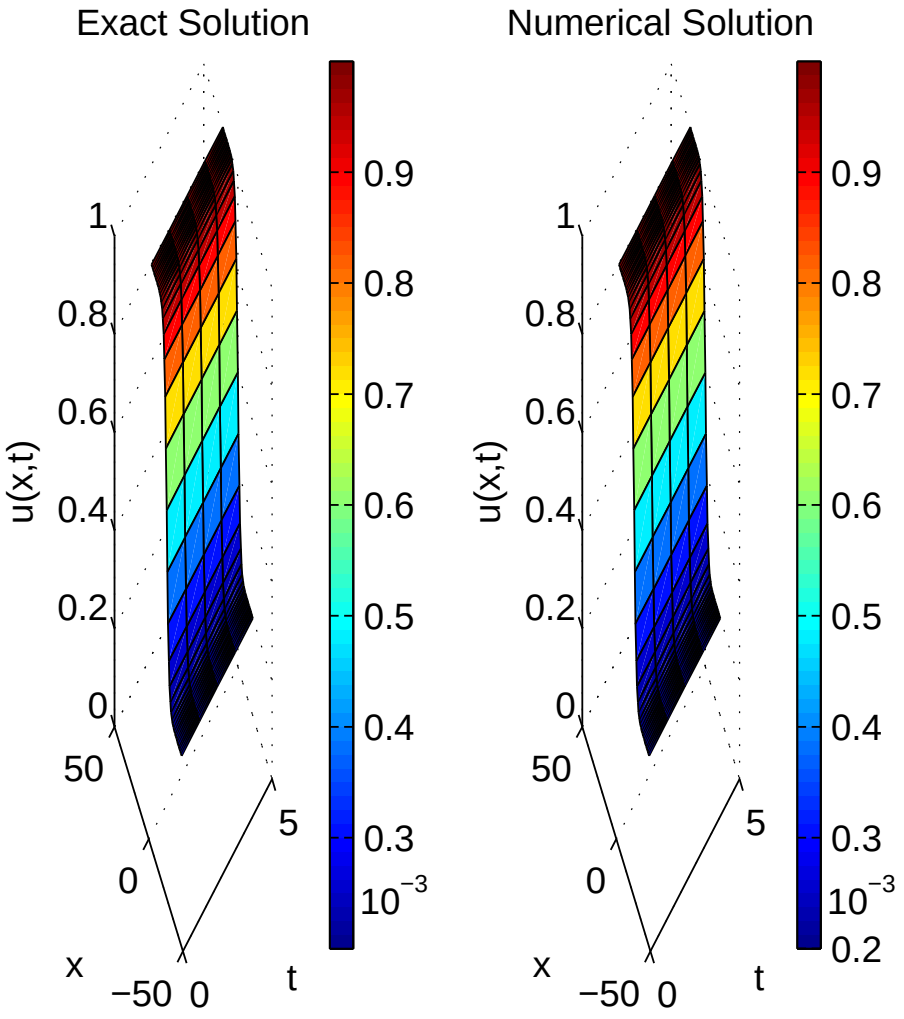

FIGURE 2.4: Traveling wave solutions of example 2.1 for $\rho = 1$ and $N = 41$ at $t = 0.001, 0.002, 0.003, 0.004, 0.005$

TABLE 2.7: L_2 and L_∞ errors in the solution of example 2.3 for $\rho = 1$ with $h = 0.0001$ and $\triangle t = 0.00001$ at different time levels

Time	0.001	0.002	0.003	0.004
L_2	1.1237E-5	1.6863E-5	2.6881E-5	3.9278E-5
L_∞	2.0834E-5	4.1668E-5	6.2504E-5	8.3340E-5

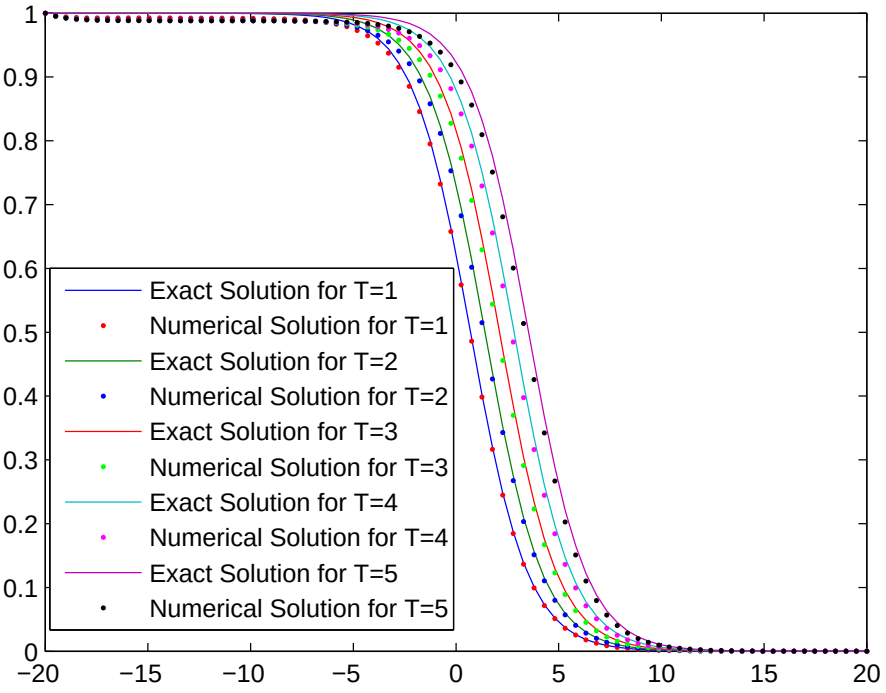

FIGURE 2.5: Time dependent profiles versus x of example 2.2 for $\rho = 1$ and N= 41 at time $t = 1, 2, 3, 4, 5$

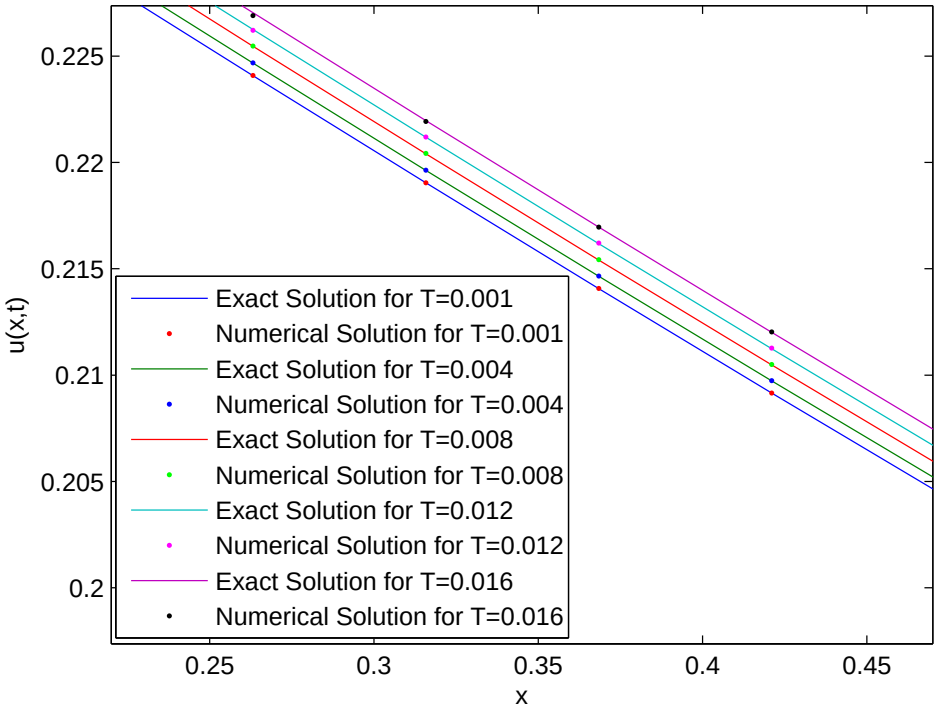

FIGURE 2.6: Time dependent profiles versus x of example 2.3 for $\rho = 1$ and N= 41 at time $t = 0.001, 0.004, 0.008, 0.012, 0.016$

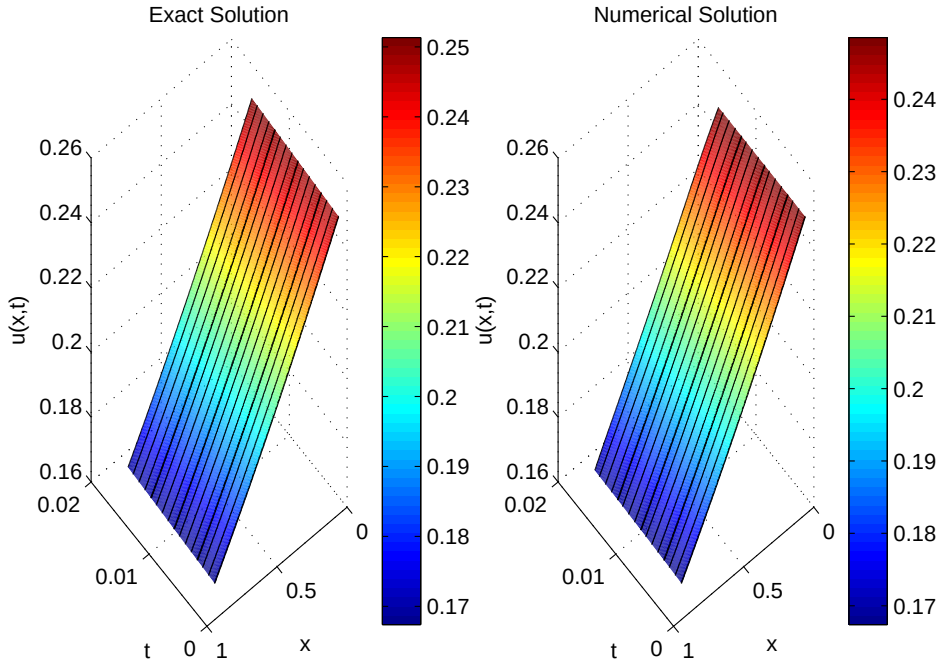

FIGURE 2.7: Traveling wave solutions of example 2.3 for $\rho = 1$ and $N = 410$ at $t = 0.001, 0.004, 0.008, 0.012, 0.016$

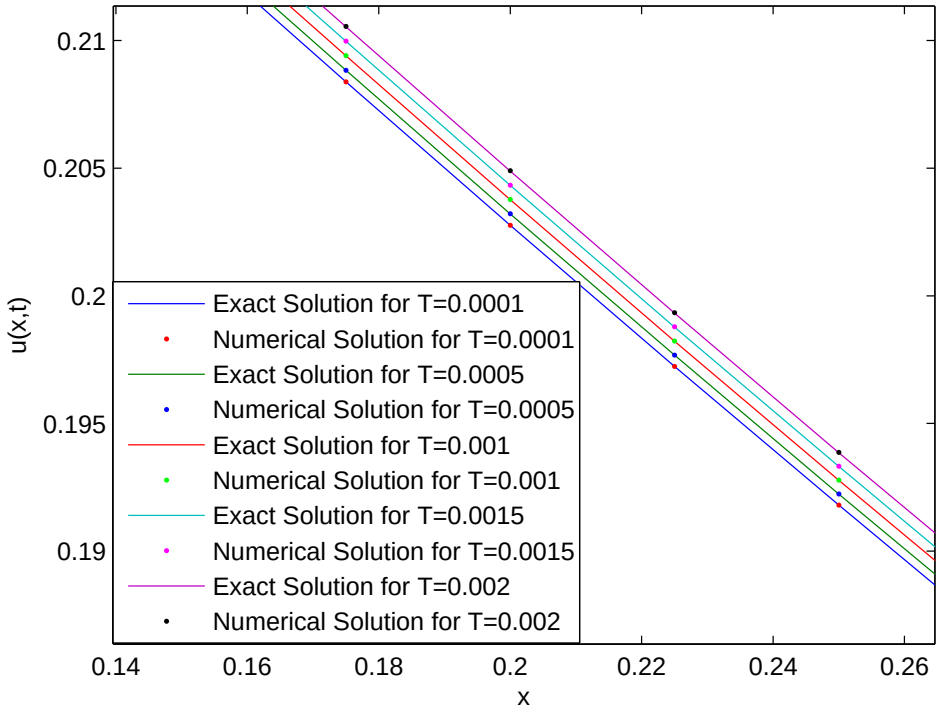

FIGURE 2.8: Time dependent profiles versus x of example 2.3 for $\rho = 6$ and N= 41 at time $t = 0.0001, 0.0005, 0.001, 0.0015, 0.002$

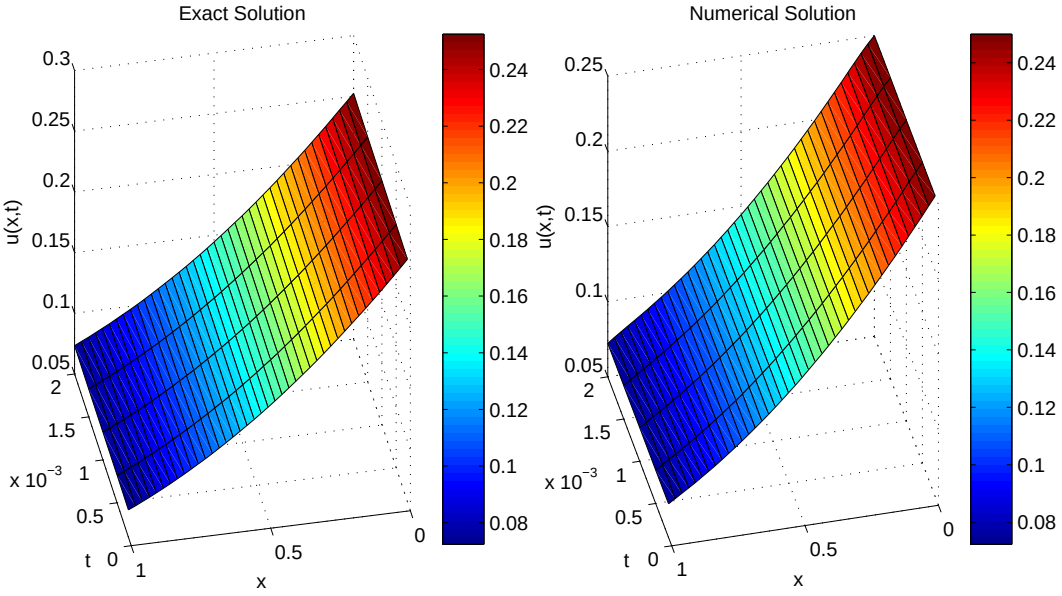

FIGURE 2.9: Traveling wave solutions of example 2.3 for $\rho = 6$ and $N = 41$ at $t = 0.0001, 0.0005, 0.001, 0.0015, 0.002$

TABLE 2.8: L_2 and L_∞ errors in the solution of example 2.3 for $\rho = 6$ with $h = 0.0001$ and $\triangle t = 0.000001$ at different time levels

Time	0.0001	0.0002	0.0003	0.0004
L_2	7.0594E-5	1.5179E-4	2.4280E-4	3.4091E-4
L_∞	1.2502E-4	2.5006E-4	3.7514E-4	5.0025E-4

Chapter 3

Numerical solution of second order partial differential equations using cubic trigonometric differential quadrature method

In the previous chapter, a second order Fisher's equation has been solved using cubic trigonometric collocation method. In collocation method, there are some issues like computational complexity, more computational timing, and moreover method give good results (fewer errors) for a large number of domain partitions. To overcome these issues in this chapter differential quadrature method is hybrid with trigonometric B-spline basis functions. To show the efficiency of the developed method, four important second order partial differential equations are simulated for the numerical solution in this chapter and they are as follows:

3.1 Introduction to sine-Gordon equation

Partial differential equations arise during the mathematical modelling of the various phenomenon of science and engineering. One of the well known partial differential equation is the sine-Gordon equation which has solitons solution in both

one and two dimensions. The sine-Gordon equation in one dimension is given as:

$$u_{tt} + \alpha u_t = \beta u_{xx} + \eta(x) sin(u),\ x \in [a, b],\ t \geq 0 \tag{3.1}$$

with initial conditions

$$\begin{aligned} u(x, 0) &= g_1(x) \\ u_t(x, 0) &= g_2(x) \end{aligned} \tag{3.2}$$

and boundary condition

$$u(a, t) = l_1,\ u(b, t) = l_2 \tag{3.3}$$

where α and β are real constants and $\eta(x)$ is known as Josephson current density; $g_1(x)$ is known as wave modes or kinks and $g_2(x)$ is the velocity of the wave. In this equation, α is a dissipative term, which is assumed to be a real number with $\alpha \geq 0$. When $\alpha = 0$, this equation reduces to the undamped sine-Gordon equation in two space variables, while when $\alpha > 0$, the equation becomes damped.

This is a nonlinear partial differential equation having application in various fields of physics. For instance, it appears in the application of nonlinear optics, the stability of fluid motion, motion of pendulum attached to a stretched string etc. The sine-Gordon equation is a variant form of KDV equation having solitons solution. As discussed by Dodd [101], a soliton can be described as a self-reinforcing solitary wave which propagates forwards with same velocity and shape and will remain unaffected by the interaction of several solitons. This equation has been solved by various researchers because of its existence in the phenomenon involving waves.

3.1.1 Methods proposed to solve sine-Gordon equation type equation

Due to the existence of the soliton solution of the sine-Gordon (SG) equation, it has been investigated analytically and numerically by many authors. Modified decomposition method has been used to find the solution of the sine-Gordon equation in one and two dimensions by Kaya [102] and Ray [103]. The (N+1)-dimensional sine-Gordon equation is solved numerically by Wang [104] using modified Adomian

decomposition method. The numerical simulation of the equation with initial conditions is done by Yucel [105] using Homotopy analysis method. The traveling wave solutions of the sine-Gordon and the coupled sine-Gordon equations are obtained by using the Homotopy-Perturbation approach by Sadighi et al. [106]. Kuo [107] in his paper, analyzed the solutions of the combined sine-cosine-Gordon equation proposed by Wazwaz. The equation has been solved for one and two dimensions by Deghan and his co-authors by various approaches including boundary element and boundary integral approach [108–110]. Differential quadrature method has been used by Jiwari et al. [111] to solve the equation in two-dimension. This equation in one dimension has been solved by Mittal and Rachna [112] using modified cubic B-spline collocation method. Recently, the equation has been solved by F. Yin et al. [113] using spectral method considering Legendre wavelets as basis functions. In the present chapter, this equation is solved using trigonometric B-spline basis functions in differential quadrature method. In authors' knowledge, the equation has not been solved in either one or two dimensions using trigonometric B-splines.

3.1.2 Implementation of numerical scheme on sine-Gordon equation

In order to solve the equation (3.1) the transformation $u_t = v$ is used to reduce the equation in the following system of differential equations:

$$\begin{aligned} u_t &= v, \\ v_t + \alpha\ v &= \beta\ u_{xx} + \eta(x)\ sin(u). \end{aligned} \tag{3.4}$$

On substituting the second order approximation of the spatial derivative in equation (3.4) results in the following system:

$$\begin{aligned} u_t &= v, \\ v_t + \alpha\ v &= \beta\ \sum_{j=1}^{N} b_{ij} u(x_j, t) + \eta(x)\ sin(u) \end{aligned} \tag{3.5}$$

to solve the the above system of ODE we use strong stability-preserving time-stepping RungeKutta (SSP-RK43) scheme [72].

3.1.3 Stability of the scheme

To check the stability of the scheme one can convert the given equations (3.1) to the system of ODEs and stability of the obtained system can be verified using the matrix method. On substituting the approximate values of derivatives in the sine-Gordon equation and taking the nonlinear terms as constant. These equations can be written in simplified form as follows:

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ \\ v \end{bmatrix} = B[u \quad v]^T + f(u,v) \tag{3.6}$$

where the value of $B = \begin{bmatrix} 0 & I \\ \\ \beta\, b_{ij} & 0 \end{bmatrix}$ and $f(u,v)$ is nonliear terms of the equation. The stability of the system (3.6) depends on the eigenvalues of B. Since all eigenvalues of the matrix B satisfy the conditions of [73] as given in Figure 3.1. Hence it can be concluded that the scheme is stable.

3.1.4 Numerical Experiment and Discussion

In this section, four test problems of the sine-Gordon equation are solved and examined for the numerical results. The accuracy and the efficiency of the method are measured by evaluating the L_2, L_∞ and RMS errors. Numerical results are compared and found better from the results present in the literature.

Example 3.1. *In this example the numerical solutions of sine-Gordon equation (3.1) are obtained in one-dimension on the computational domain* $x \in [-2, 2]$ *for* $\alpha = 0$, $\beta = 1$ *and* $\eta(x) = -1$ *with initial conditions* $g_1(x) = 0$ *and* $g_2(x) = 4sech(x)$. *The boundary conditions are obtained from the exact solution* $u(x,t) = 4tan^{-1}(tsech(x))$. *For numerical computation time step is taken as* $\Delta t = 0.0001$ *and space step size is* $h = 0.04$. *In Table, 3.1 the obtained* L_2, L_∞ *and RMS errors are enlisted and compared with results of Mittal and Rachna [112] and Li-Min and Zong-Min [114]. The obtained results are in good agreement with results*

obtained by other researchers as well as with the exact solution. Comparison of numerical and the exact solution is also depicted in Figure 3.2 at different time levels.

Example 3.2. *In this example the numerical solutions of one dimension sine-Gordon equation (3.1) are obtained in the computational domain* $x \in [-3, 3]$ *for* $\alpha = 0$, $\beta = 1$ *and* $\eta(x) = -1$ *with initial conditions* $g_1(x) = 4tan^{-1}(exp(\frac{x}{\sqrt{1-c_1^2}}))$ *and* $g_2(x) = \frac{-4c_1\gamma exp(\gamma x)}{1+exp(2\gamma x)}$. *The boundary conditions are obtained from the exact solution* $u(x,t) = 4tan^{-1}(ep(\gamma(x - c_1 t)))$ *where* c_1 *is the velocity of solitary wave and* $\gamma = \frac{1}{\sqrt{1-c_1^2}}$. *For numerical computation,* Δt *is taken as* 0.0001 *and space step size is* $h = 0.04$. *In Table 3.2, the* L_2, L_∞ *and RMS errors are enlisted and compared with results given in the literature [109, 112]. From the results, it can be concluded that obtained results are in good agreement and even superior as compared to results given by other researchers. The comparison of numerical and the exact solution is depicted in Figure 3.3 at different time levels.*

Example 3.3. *Consider the hyperbolic wave equation for* $x \in [0, 1]$ *as:*

$$u_{tt} = u_{xx} + (\pi^2 + 0.25)exp(-0.5\ t)sin(\pi\ x), \tag{3.7}$$

initial and boundary conditions can be obtained from the exact solution given by:

$$u(x,t) = exp(-0.5\ t)sin(\pi x).$$

Numerical solution for this equation is calculated at different time for $k = 0.0001$, $N = 51$ *and* 101. *Obtained error norms and their comparison with [1] are presented in the Tables 3.3 and 3.4. Comparison of the numerical and the exact solution of this test problem is shown in Figure 3.4.*

Example 3.4. *Consider the hyperbolic wave equation for* $x \in [0, 1]$ *as:*

$$u_{tt} = u_{xx} + (x^2 - 2)exp(t), \tag{3.8}$$

initial and boundary conditions can be obtained from the exact solution given by:

$$u(x,t) = x^2 exp(t).$$

Numerical solution for this equation is calculated at different time for $k = 0.0001$ *and* $N = 51$. *Obtained error norms are presented in the Table 3.5. Comparison of the numerical and the exact solution of this test problem shown in Figure 3.5.*

3.1.5 Summary

Due to the existence of soliton solutions, the sine-Gordon equation has been solved by researchers in both one and two dimensions using various methods. In this chapter, the equation is solved using modified trigonometric cubic B-spline basis function with differential quadrature method. Four test problems are taken for verifying the efficiency and applicability of the method by calculating the L_2, L_∞ and RMS errors in one dimension. From the obtained results it is evident that the obtained numerical solutions are in good conformity with the exact/analytical solutions. The developed method proves to be an economical and efficient technique for finding numerical solutions for a variety of physical models.

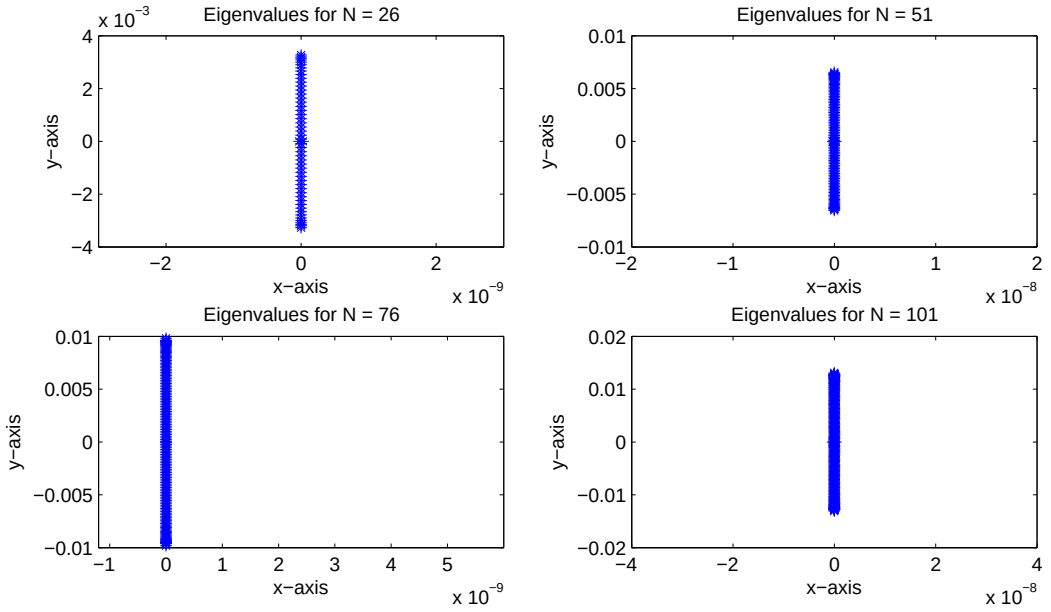

FIGURE 3.1: Eigenvalues of the matrix B for different partitions of the domain in one dimension

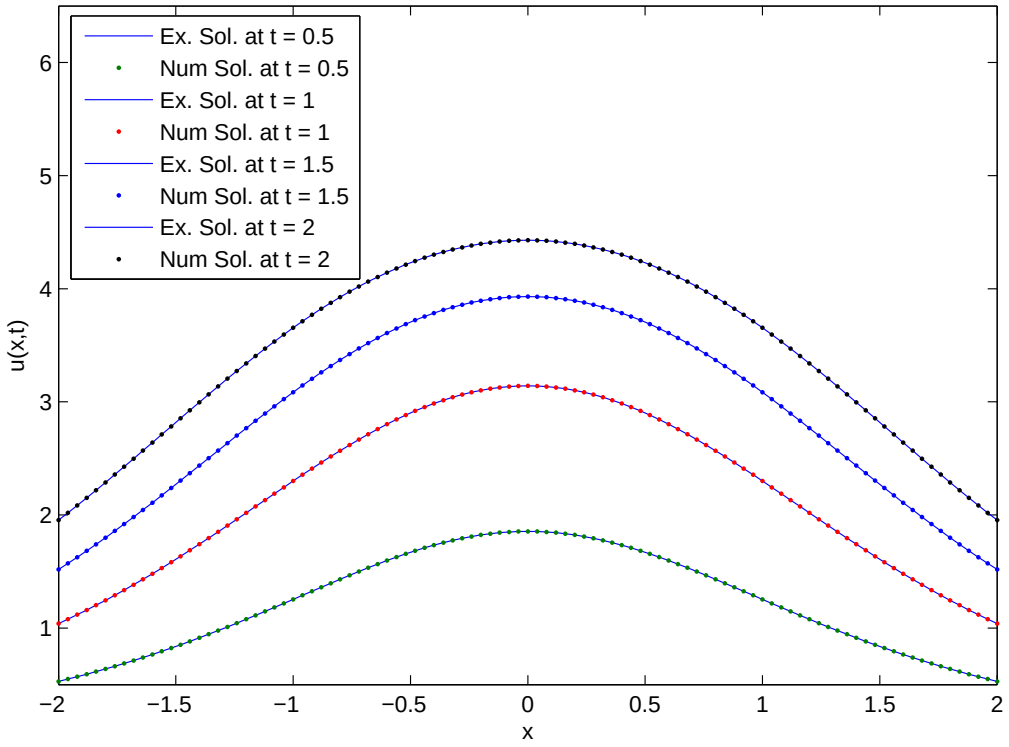

FIGURE 3.2: The traveling wave solution of example 3.1 at different time levels with $\Delta t = 0.0001$ and $h = 0.04$

TABLE 3.1: Comparison of L_2, L_∞, and RMS errors in the MCTB-DQM solution of example 3.1 for $\Delta t = 0.0001$ and $h = 0.04$ at different time levels with the errors obtained by other researchers

	MCTB-DQM			Mittal and Rachna[112]		Li-Min and Zong-Min[114]	
t	L_2	L_∞	RMS	L_∞	RMS	L_∞	RMS
0.3	5.55E-5	1.05E-4	2.72E-6	4.54E-5	6.36E-7	9.02E-5	3.60E-5
0.6	6.61E-5	1.03E-4	3.24E-6	1.55E-4	2.96E-6	3.73E-4	1.65E-4
1	7.08E-5	9.93E-5	3.47E-6	3.84E-4	9.56E-6	8.49E-4	4.37E-4

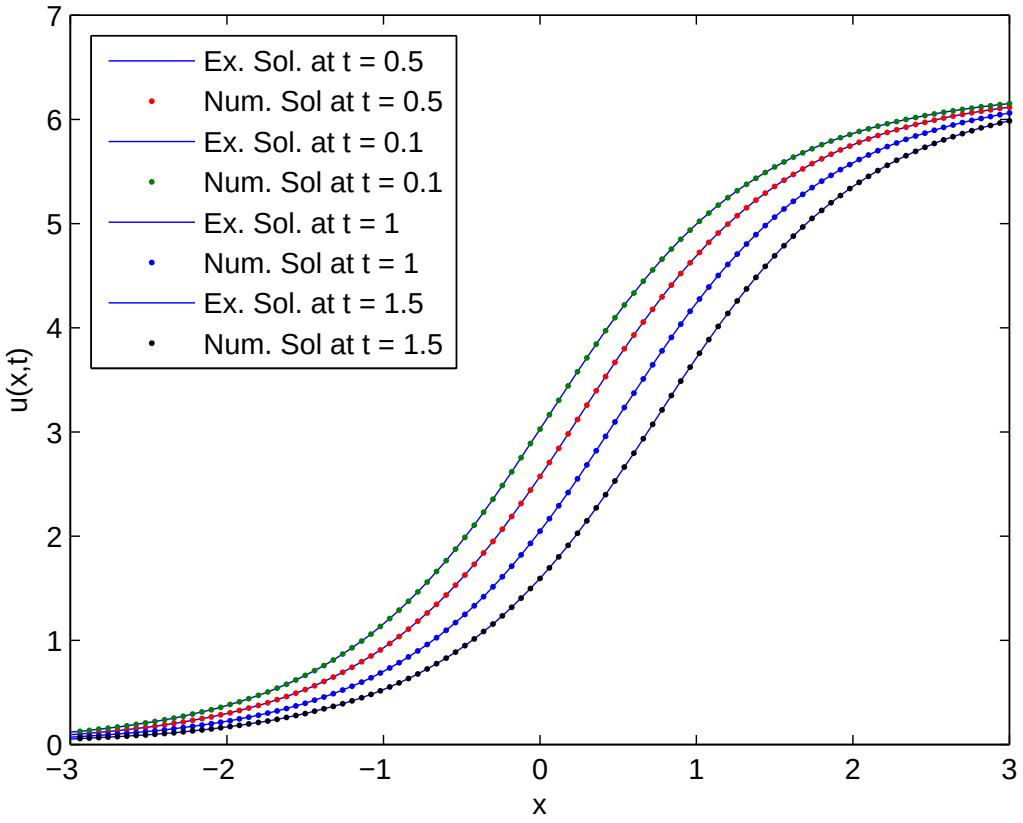

FIGURE 3.3: The traveling wave solution of example 3.2 at different time levels with $\Delta t = 0.0001$ and $h = 0.04$

TABLE 3.2: Comparison of L_2, L_∞, and RMS errors in the MCTB-DQM solution of example 3.2 for $\Delta t = 0.0001$ and $h = 0.04$ at different time levels with the errors obtained by other researchers

Time	MCTB-DQM			Mittal and Rachna [112]		Dehghan and Shokri [109]	
t	L_2	L_∞	RMS	L_2	L_∞	L_2	L_∞
0.25	1.21E-5	2.34E-5	4.86E-7	3.66E-5	4.90E-5	1.76E-5	4.95E-6
0.5	1.75E-5	2.39E-5	7.01E-7	9.00E-5	7.55E-5	4.31E-5	8.42E-6
0.75	2.12E-5	2.41E-5	8.47E-7	1.60E-4	1.43E-4	8.25E-5	1.65E-5
1	2.41E-5	2.40E-5	9.63E-7	2.27E-4	2.10E-4	1.27E-4	2.51E-5

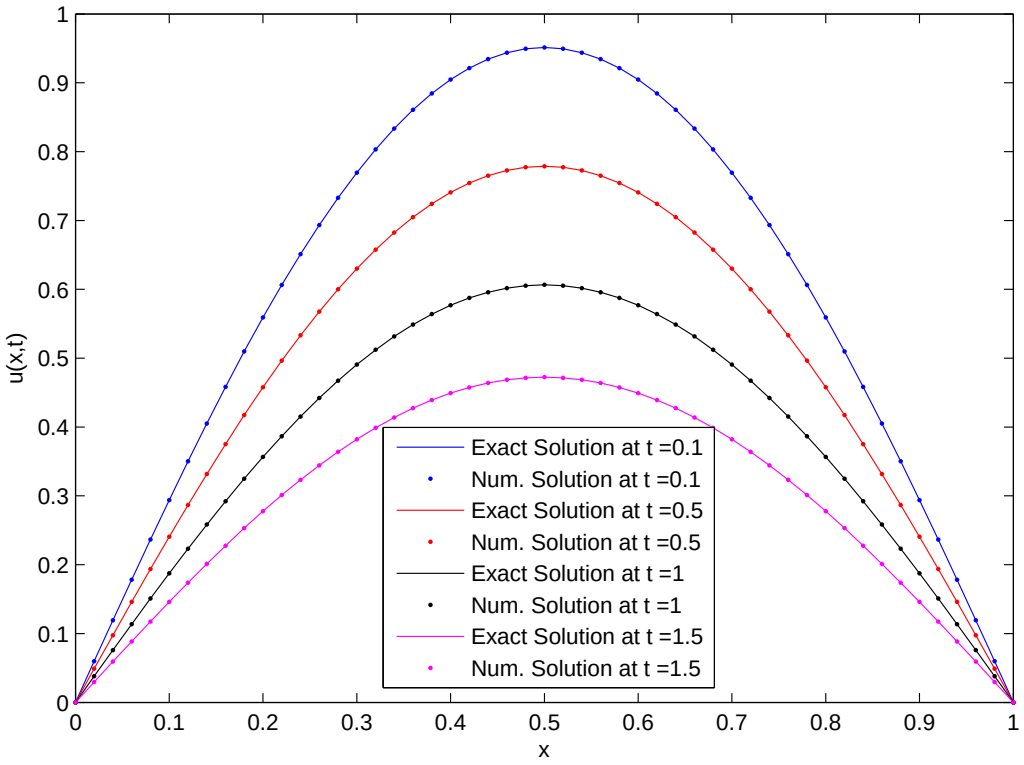

FIGURE 3.4: The traveling wave solution of example 3.3 at different time levels with $\Delta t = 0.0001$ and $N = 51$

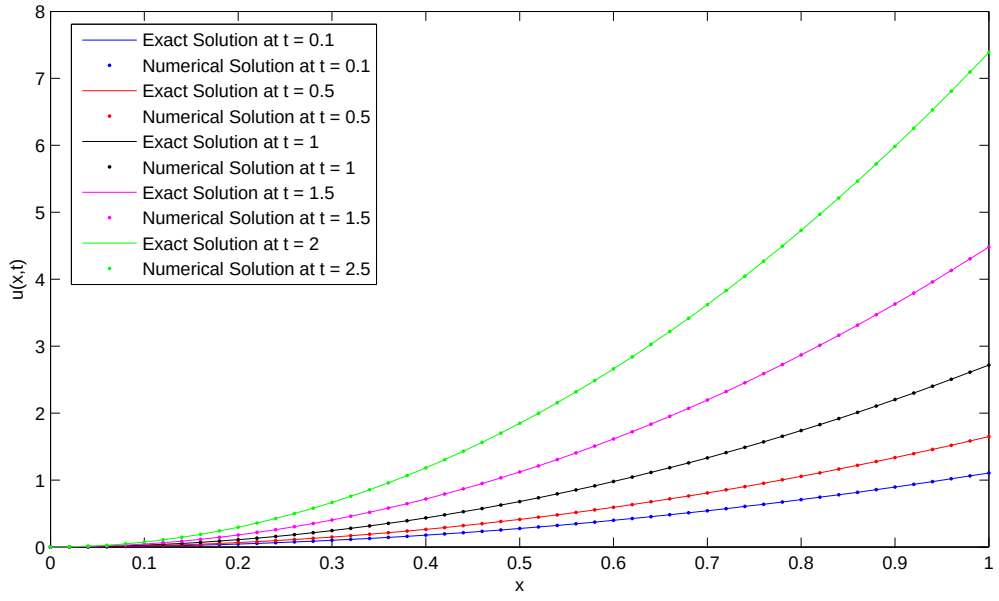

FIGURE 3.5: The traveling wave solution of example 3.4 at different time levels with $\Delta t = 0.0001$ and $N = 51$

TABLE 3.3: Comparison of L_2, L_∞, and CPU time(s) in the MCTB-DQM solution of example 3.3 for $\Delta t = 0.0001$ and $N = 51$ at different time levels with the errors obtained by other researchers

Time	MCTB-DQM			M. Abbas [1]	
t	L_2	L_∞	CPU time(s)	L_∞	CPU time(s)
0.1	3.90E-6	9.84E-6	0.19	-	-
0.5	2.41E-5	3.34E-5	0.83	-	-
1	4.13E-5	5.42E-5	2.14	2.45E-5	2.219
1.5	8.22E-6	1.19E-5	2.81	-	-
2	1.63E-5	2.16E-5	3.32	-	-

TABLE 3.4: Comparison of L_2, L_∞, and CPU time(s) in the MCTB-DQM solution of example 3.3 for m $\Delta t = 0.0001$ and $N = 101$ at different time levels with the errors obtained by other researcher

Time	MCTB-DQM			M. Abbas [1]	
t	L_2	L_∞	CPU time(s)	L_∞	CPU time(s)
0.1	1.03E-6	1.53E-6	0.69	-	-
0.5	1.75E-5	2.47E-5	3.63	3.34E-5	38.937
1	3.00E-5	4.19E-5	4.65	8.06F-4	77.484
1.5	5.85E-6	8.34E-6	7.14	-	-
2	1.18E-5	1.65E-5	9.86	-	-

TABLE 3.5: Error norms in the MCTB-DQM solution of example 3.4 for m $\Delta t = 0.0001$ and $N = 51$ at different time levels

Time	MCTB-DQM	
t	L_2	L_∞
0.1	2.15E-5	1.10E-4
0.5	4.95E-5	1.64E-4
1	5.15E-5	2.71E-4
1.5	1.14E-4	4.48E-4
2	1.59E-4	7.38E-4

3.2 Introduction to Burgers' equation

Relation either between convection and diffusion or between diffusion and reaction portrays numerous physical phenomenon in physical sciences. From a physical perspective, the basic equations to portray a wide assortment of issues in physical, compound, natural, and designing sciences is either the diffusion-reaction process or the convection-diffusion process. These processes are being modeled by many nonlinear partial differential equations given by numerous new experiences concerned with the association of nonlinearity with dispersion. One of the significant nonlinear equation is Burgers' equation which is very important in various fields of research. This well-known equation in one dimension can be expressed as follows:

$$u_t + \alpha u u_x = \nu u_{xx} \tag{3.9}$$

here kinematic consistency is represented by ν. The two dimensional form of Burgers' equation is:

$$u_t = -uu_x - uu_y + \frac{1}{R}(u_{xx} + u_{yy}) \tag{3.10}$$

where R is Reynolds number.

Burgers' equation is also called as viscid Burgers' equation and gives rise to a form

known as inviscid Burgers' equation when the kinematic thickness is not considered. This equation acts as a good example of harmony among time development, nonlinearity, and dispersion. In nonlinear mathematical models, this is one of the easiest model of partial differential equation (PDE) for waves with the diffusive term in liquid elements. Burgers [115] initially builds up this equation fundamentally to reveal insight into the investigation of turbulence portrayed by the connection of the two inverse impacts of convection and dispersion. In any case, turbulence is more perplexing as it is both 3D and measurably irregular in nature. Mathematical equation (3.9) emerges in numerous physical issues including 1D turbulence, waves in elastic tubes with fluid, shock waves, and sound waves in a viscous medium and in traffic flow model. In traffic flow model same equation has been obtained by transformation $u = 2\rho - 1$ and $u \in [-1, 1]$ where ρ represents the density of the cars on the road.

Note that equation (3.9) is parabolic but become hyperbolic when $\nu = 0$. Essentially, the solution properties of the parabolic equation are fundamentally different than a hyperbolic equation. Moreover, Burgers' equation can also be obtained from the generalized Burgers' equation by substituting $\alpha = 1$, $\delta = 1$ and $c = 0$ in the following equation:

$$u_t + (c + \alpha u^{\delta})u_x - \nu u_{xx} = 0 \tag{3.11}$$

where α is real parameter and $\delta > 0$. For particular value of constants, the equation serves as a model equation for the boundary-layer equations and is fundamentally same to the equations governing fluid flow. For $c = 0$, $\alpha = 1$ and $\delta = 2$ the equation (3.11) becomes modified Burgers' equation as:

$$u_t + u^2 u_x - \nu u_{xx} = 0 \tag{3.12}$$

To solve the concerned equation with differential quadrature method the trigonometric B-spline basis functions are used with modification to lead a diagonally dominant system with the initial conditions given as follows:

$$u(x, 0) = u_0(x),\ x \in (-\infty, \infty), \tag{3.13}$$

with the boundary conditions for Burgers' equation in one dimensions (3.9) defined as follows:

$$\lim_{x \to -\infty} u(x, t) = g_0,\ \lim_{x \to \infty} u(x, t) = g_1. \tag{3.14}$$

For equation (3.10) in two dimensions the initial and boundary conditions for equation (3.10) are as follows:

$$u(x, y, 0) = f(x, y),\ a < x < b,\ c < y < d, \tag{3.15}$$

$$u(a, y, t) = g_1(y, t),\ u(b, y, t) = g_2(y, t),\ u(x, c, t) = g_3(x, t),\ u(x, d, t) = g_4(x, t). \tag{3.16}$$

In recent years, a lot of work has been done for the proficient numerical solution of the Burger's equation in estimations of the kinematic thickness. This equation is unraveled for both the interminable space and the limited space [116]. The different strategies for numerical and the exact solution of this equation are Cecchi [117] and Ozis [118] used finite elements method, Dogan [119] implemented Galerkin finite element method, Dag [120] implemented cubic B-splines collocation approach, an automatic differentiation method is used by Asaithambi [121], Mittal and Jain [122] used MCB-spline with collocation method, Khalifa [123] used spectral collocation method, in the paper Jiang [124] used cubic B-spline quasi-interpolation, Korkmaz and dag [43] found the approximate solution using polynomial based differential quadrature method, Korkmaz [44], Korkmaz and Aksoy [45] used cubic and quartic B-spline DQM respectively, Saka [125] implemented the quartic B-splines with collocation method to solve the Burgers' equation, Korkmaz in his paper [42] used sinc DQM to solve this equation, finite difference element approach is implemented on Burgers' equation by Aksan [126] and Ozis [127] using quadratic B-splines as basis, Hassanien [128] used a finite difference method of fourth-order, factorized diagonal Padé approximation is implemented to Burgers' equation by Altparmak [129], Ramadan [130] used a non-polynomial spline approach, a novel numerical scheme is proposed by Min Xu [131], Kutluay [132] proposed an explicit and exact-explicit finite difference methods, Cole [133] introduces the Hopf-Cole transformation to solve the equation. Kutluay and Esen [134] used the least-squares quadratic B-splines finite element method, Kumar[135] used a high order Muscl scheme to solve transport equation, reproducing kernel function method is used by Xie [136], Liao [137] solved the Burgers' equation numerically using implicit fourth order compact finite difference method, Jiwari and Mittal [138] used weighted average differential quadrature method, Aksan and Ozdes [126] used the

variational method, Kadalbajoo and Sharma [139] proposed the numerical solution of this equation using parameter uniform implicit difference scheme, Mittal and Singhal [140] used one dimensional Fourier expansion. Ay et al. [141] used trigonometric quadratic B-spline with Galerkin algorithm, Cubic B-spline with DQM is used by Korkmaz [44] to solve the Burgers' equation and also discusses the stability of the scheme. Time fractional Burgers' equation is solved by Esen in [142, 143]. Kutluay et al. [144] used collocation method with cubic B-spline basis to solve the modified Burgers' equation. The modified bi-quintic B-spline basis function is used to solve two dimensional unsteady Burgers' equation by Kutluay [145].

3.2.1 Implementation of the numerical scheme on Burgers' equation

By substituting the values of derivatives in equation (3.9), we obtain the following system

$$u_t = \nu \sum_{j=1}^{N} b_{ij}\ u_j - \alpha u \sum_{j=1}^{N} a_{ij}\ u_j \tag{3.17}$$

where the values of a_{ij} and b_{ij} are given as follows:

$$a_{ij} = \begin{bmatrix} a_2+2a_1 & a_1 & 0 & 0 & . & 0 \\ 0 & a_2 & a_1 & . & . & 0 \\ 0 & a_1 & a_2 & a_1 & . & 0 \\ 0 & 0 & a_1 & a_2 & a_1 & 0 \\ 0 & . & . & . & . & 0 \\ 0 & 0 & . & . & a_1 & a_2+2a_1 \end{bmatrix} \tag{3.18}$$

and

$$b_{ij} = 2a_{ij}\big(a_{ii} - \frac{1}{x_i - x_j}\big), \ for\ i \neq j, \ and\ b_{ii} = -\sum_{i=1, i\neq j}^{N} b_{ij}. \tag{3.19}$$

After substituting the values of a_{ij} and b_{ij} in equation (3.17), we get a system of ODEs which can be solved by a modified form of Runge-Kutta scheme named as strong stability preserving Runge-Kutta (SSP-RK43) scheme [72].

3.2.2 Stability of the scheme

To check the stability of the scheme one can convert the given equation (3.9) to the system of ODEs and stability of the obtained system can be verified using the matrix method. On substituting the approximate values of derivatives in the Burgers' equations and taking the nonlinear terms as constant. These equations can be written in simplified form as follows:

$$u_t = Bu + f(u) \tag{3.20}$$

where $f(u)$ represent the nonlinear terms in the Burgers' equation. The stability of the system (3.20) depends on the eigenvalues of $B = \nu \sum_{j=1}^{N} b_{ij}$ where $i = 1, 2, 3..., N$ in one dimension. Since all eigenvalues of the matrix, B are real and negative and lie in the region given in chapter 1 and presented in Figure 3.6. Hence it can be concluded that the scheme is stable.

3.2.3 Numerical results and discussion

In this section, we have simulated some problems of Burgers' equation and also examined the numerical results. The accuracy and the efficiency are measured by evaluating the L_2 and L_∞ norm.

Example 3.5. *Consider the equation given by (3.9) with* $\alpha = 1$ *with the domain* $[0, 1.2]$ *and the initial condition given by [70]*

$$u(x, 1) = \frac{x}{1 + exp(\frac{1}{4\nu}(x^2 - \frac{1}{4}))},$$

and the boundary conditions, for $t > 1$ *are* $u(0,t) = 0$, $u(1.2,t) = 0$. *For comparison, the exact solution for this equation is taken as*

$$u(x,t) = \frac{\frac{x}{t}}{1 + (\frac{t}{t_0})^{\frac{1}{2}} exp(\frac{x^2}{4\nu t})}$$

for $t > 0$, *where* $t_0 = exp(\frac{1}{8\nu})$. *To compare the obtained numerical result with the results in the literature, we consider* $\nu = 0.005$ *and* $\Delta t = 0.01$. *The numerical solution of the equation has been obtained at different time levels and at different node points and is presented in Table 3.6 and in Figures 3.8 and 3.9*

Example 3.6. *Let us consider the Burgers' equation (3.9), for* $\alpha = 1$, *over the region* $[0,2]$ *with exact solution [70]:*

$$u(x,t) = 2\pi\nu \frac{sin(\pi x)exp(-\pi^2\nu^2 t) + 4sin(2\pi x)exp(-4\pi^2\nu^2 t)}{4 + cos(\pi x)exp(-\pi^2\nu^2 t) + 2cos(2\pi x)exp(-4\pi^2\nu^2 t)}$$

for $x \in (0,2)$ *and* $t \geq 0$, *here the boundary conditions are taken to be* $u(0,t) = 0$ *and* $u(2,t) = 0$. *Error norms are computed for time* 0.1 *and* 1 *taking* $\Delta t = 0.01$ *and comparison has been done with previously published results [70, 122] in Table 3.7. Figure 3.9 gives the comparison of numerical solution with the exact solution.*

Example 3.7. *Consider equation (3.9), with initial and boundary conditions taken from the exact solution [146]*

$$u(x,t) = \frac{\nu}{1 + \nu t}(x + tan(\frac{x}{2 + 2\nu t})), \ 0.5 \leq x \leq 1.5, \ t \geq 0$$

For this test problem, we have computed the error norms and reported the results in Table 3.8 and 3.9 for viscosity coefcient $\alpha = 1$, $\nu = 0.002$, $\Delta t = 0.002$ *and* $h = 0.02$. *We have also taken* $\nu = 0.0006666$, $\Delta t = 0.002$ *and* $h = 0.02$. *The obtained results by the proposed scheme are compared with [122, 146] and found to be better than previous results. Physical behavior in 3D and contour plot of the numerical solutions are shown in Figures 3.10 and 3.11.*

3.2.4 Summary

Due to numerous applications of Burgers' equation in various fields, this equation turns into a focus of interest of researchers who have attempted to solve this equation through their distinctive scheme. This chapter uses the MTCB-spline function to solve Burgers' equation with differential quadrature method in one dimension. The approximated results are compared with the exact solutions as well as with numerical solutions obtained by other researchers through L_2 and L_∞ errors. The conformity of the obtained numerical results with the exact solutions makes the proposed scheme evidently adequate. So it can be concluded that proposed method is an efficient method for computing the numerical solutions of a variety of linear and nonlinear equations in one dimension.

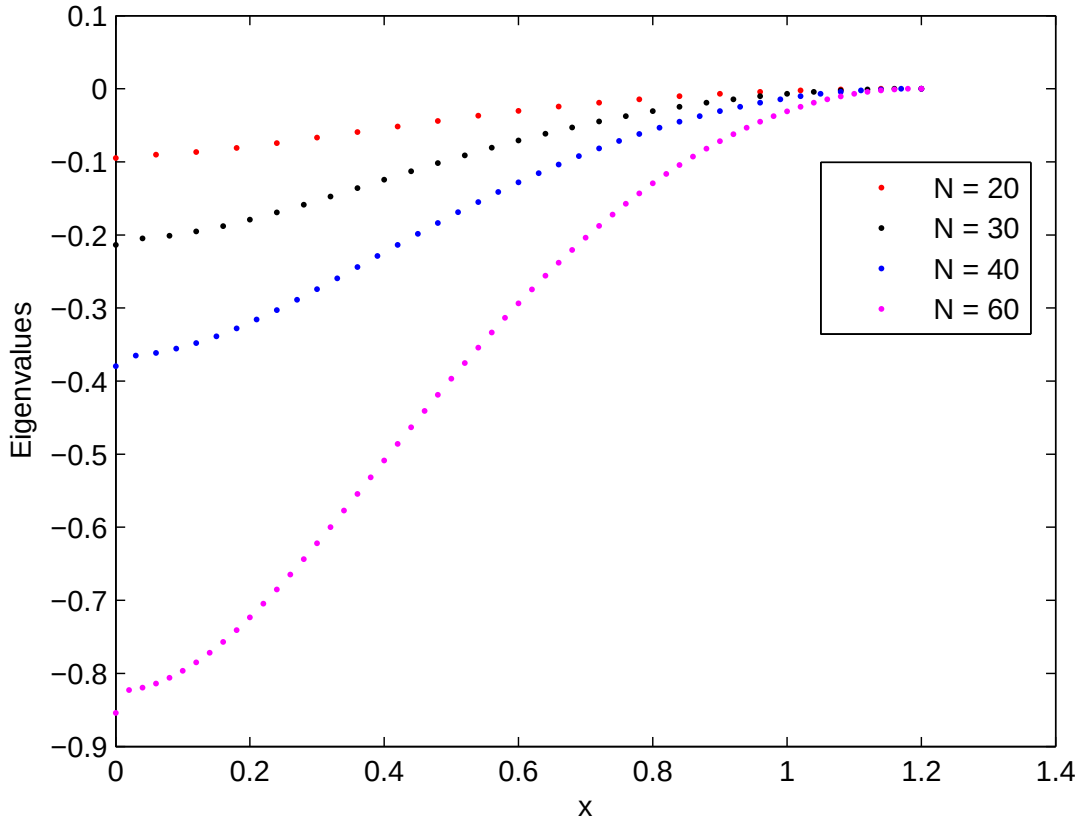

FIGURE 3.6: Eigenvalues of the matrix B for different partitions of the domain in one dimension

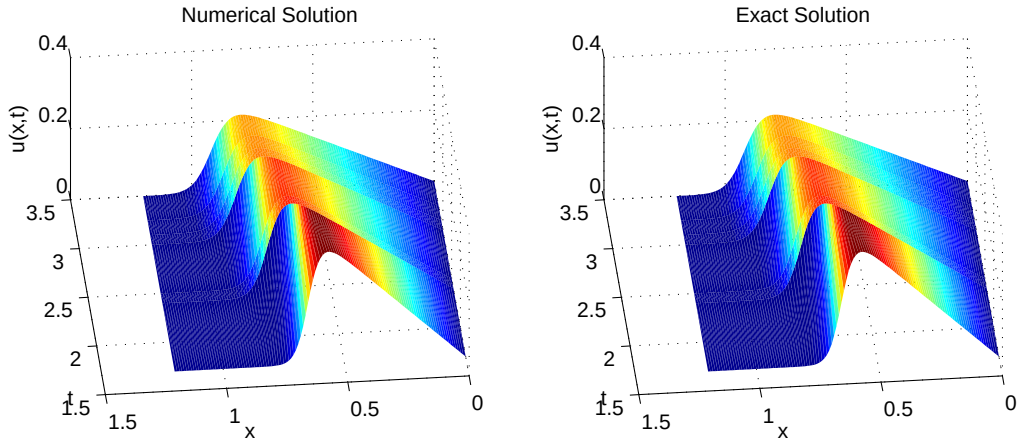

FIGURE 3.7: Traveling wave solutions of example 3.5 for time $t \leq 3.5$ with $\nu = 0.005$ and $\Delta t = 0.01$

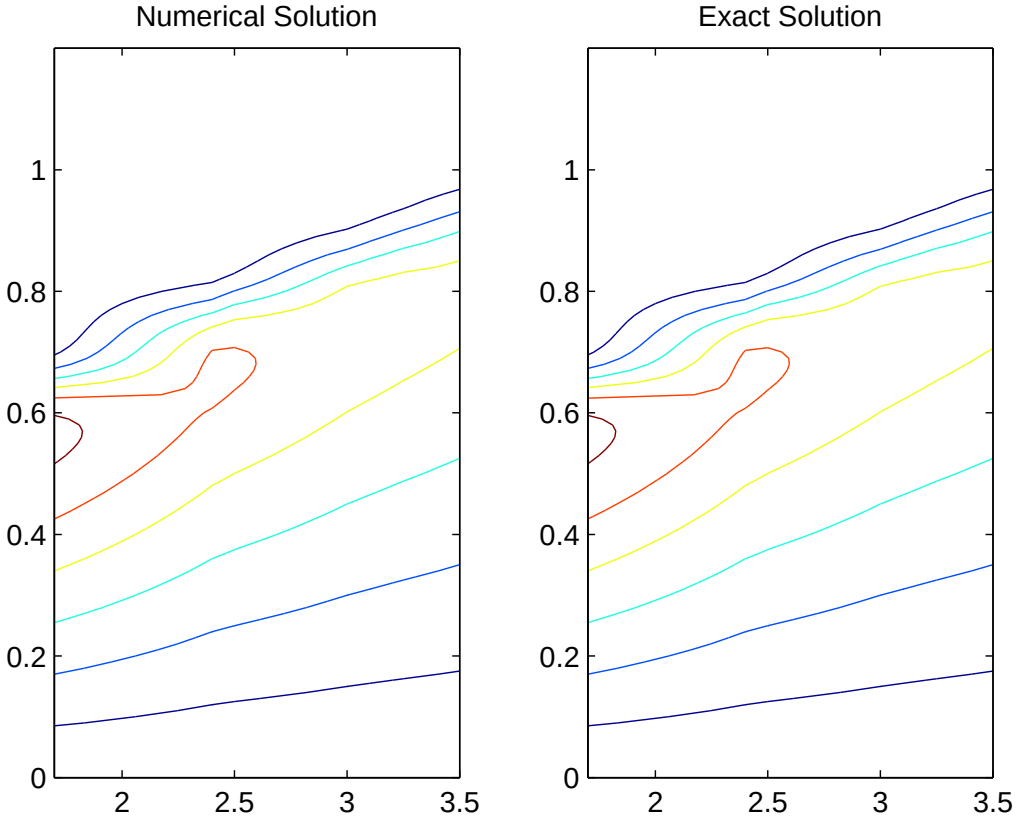

FIGURE 3.8: The contour plot of example 3.5 for time $t \leq 3.5$ with $\nu = 0.005$ and $\Delta t = 0.01$

TABLE 3.6: Comparison of L_2 and L_∞ errors in the TMCB-DQM solution of example 3.5 for $\rho = 0.005$ at different time levels

Methods	N	t=1.7		t=2.4		t=2.5		t=3.1		t=3.25		t=3.5	
		$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
MTCB-DQM	121	0.00190	0.00775	0.000866	0.00308	0.000787	0.00275	0.000475	0.00158	0.000433	0.00140	0.000486	0.00304
MCB-DQM [70]	121	0.00191	0.00777	0.00086	0.00308	0.00778	0.00275	0.00065	0.00331	0.001341	0.00918	0.006177	0.04335
QRTDQ [45]	101	0.109	0.434	0.100	0.339	–	–	0.091	0.266	–	–	–	–
BS.FEM [147]	50	0.857	2.576	0.423	1.242	–	–	0.230	0.680	–	–	–	–
C.S.C.[148]	50	0.857	2.576	0.423	1.242	–	–	0.235	0.688	–	–	–	–
Galerkin [149]	200	0.857	2.576	0.423	1.242	–	–	0.235	0.688	–	–	–	–
QBCM1[150]	200	0.017	0.061	0.012	0.058	–	–	0.601	4.434	–	–	–	–
PDQ [151]	200	0.015	0.056	0.011	0.064	–	–	0.584	4.301	–	–	–	–
CBCDQ [152]	101	–	–	0.210	0.680	–	–	0.190	0.530	–	–	–	–
QBCM [120]	200	0.072	0.311	–	–	0.051	0.189	–	–	1.129	8.983	–	–
CBCM [120]	200	2.466	27.577	–	–	2.111	25.15	–	–	1.925	21.084	–	–
MCB-CM [122]	241	0.0252	0.0994	–	–	–	–	0.0117	0.0486	–	–	0.0117	0.0486
EMCB-DQM (p=1)[153]	121	0.00173	0.00680	0.000799	0.00288	0.000729	0.00256	0.000657	0.00354	–	–	0.006152	0.0431

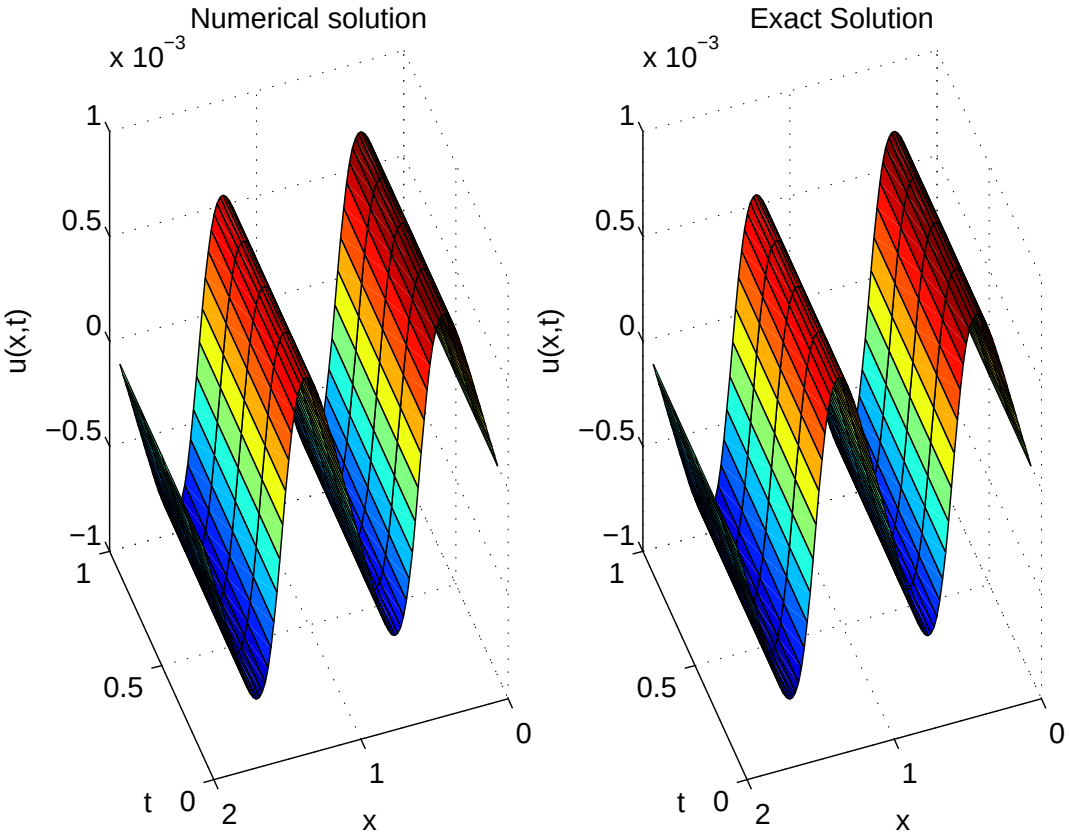

FIGURE 3.9: Traveling wave solutions of example 3.6 for time $t \leq 1$ with $\alpha = 1$ and $\Delta t = 0.01$

3.3 Introduction of Schrödinger equation

Material science has many fundamental equations for portraying the quantum mechanical behaviour [154]. Among them, the most important equation is nonlinear Schrödinger (NLS) equation which is used to elucidate the changes of a quantum system with time. This partial differential equation also known as Schrödinger wave equation, with cubic nonlinearity, delineates the advancement of the wave function of a physical framework after some time. Additionally, this equation has application in various fields of physical sciences such as in configuration of optoelectronic gadgets, in study of electromagnetic wave proliferations, quantum flow computations, underwater acoustics, portraying the mobility of Bose-Einstein condensate at temperature near absolute zero, signal propagation in optical fibers and in many other physical nonlinear systems having instability phenomena (See Refs [155–160]). This equation has many remarkable features such as the existence of a plane wave stationary solution and solution in form of the localized wave with

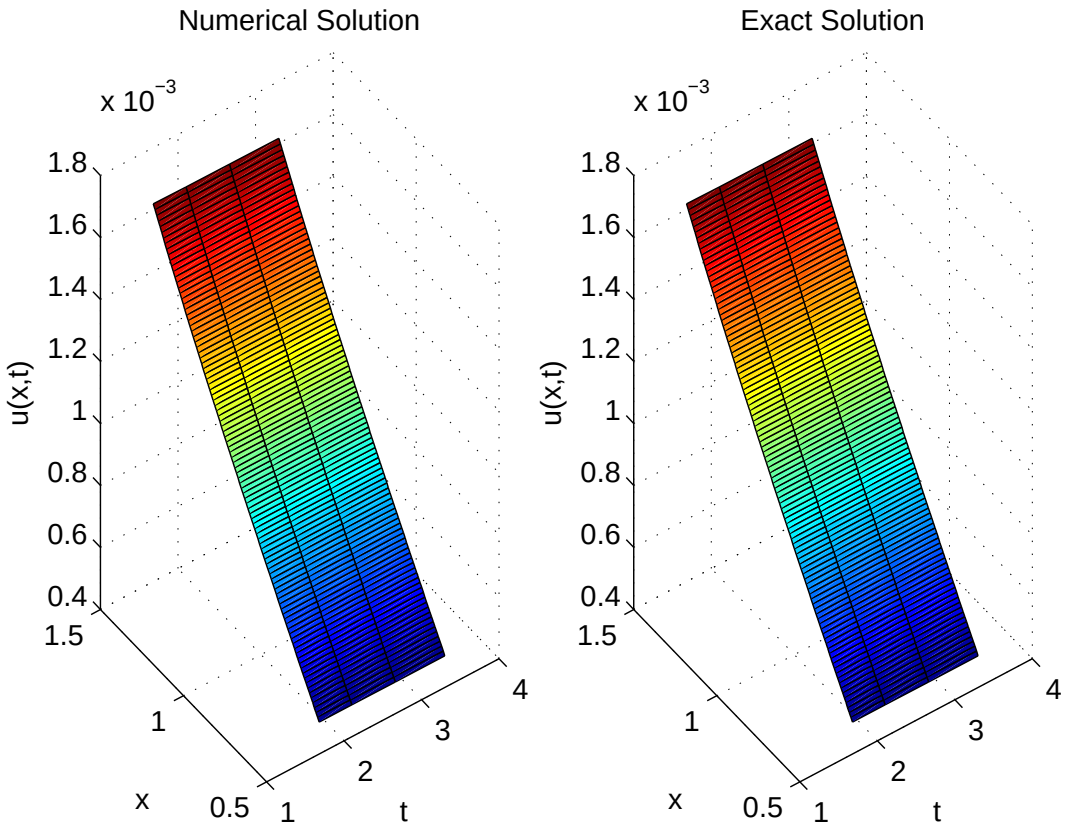

FIGURE 3.10: Traveling wave solutions of example 3.7 for time $t \leq 3.5$ with $\alpha = 1$, $\nu = 0.002$ and $\Delta t = 0.002$

preservation of shape with advancement in time, etc.

The nonlinear Schrödinger equation in one dimension is as follows:

$$i\ U_t = \alpha U_{xx} + \beta\ |U|^2 U + F(x,t)U,\ \ x \in [a,b],\ \ t \geq 0 \tag{3.21}$$

with initial condition

$$U(x,0) = U_0(x) \tag{3.22}$$

and boundary condition

$$\lim_{|x| \to \infty} U(x,t) = 0 \tag{3.23}$$

In two dimension, nonlinear Schrödinger equation can be considered in the following form:

$$i\ U_t = \alpha(U_{xx} + U_{yy}) + \beta|U|^2 U + F(x,y,t)U,\ \ x,y \in [a,b] \times [c,d],\ \ t \geq 0 \tag{3.24}$$

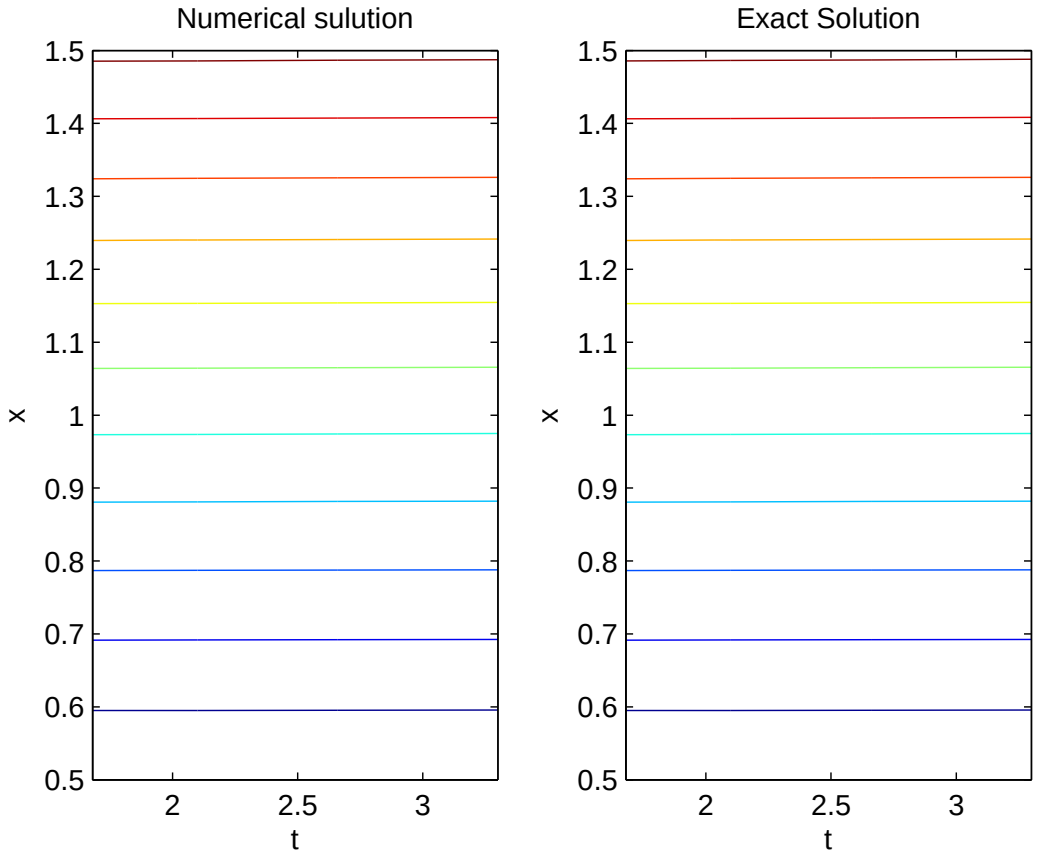

FIGURE 3.11: The contour plot of example 3.7 for time $t \leq 3.5$ with $\alpha = 1$, $\nu = 0.002$ and $\Delta t = 0.002$

with initial condition

$$U(x, y, 0) = U_0(x, y) \tag{3.25}$$

and boundary condition

$$\lim_{|x,y|\to\infty} U(x, y, t) = 0 \tag{3.26}$$

where α and β are arbitrary real values and $F(x, t)$ or $F(x, y, t)$ represents a bounded real valued function or trapping potential and $U = u + iv$ represents the complex valued wave function. The NLS equation results in two types of solutions [158]. When the solution, along with spatial derivatives, vanishes at $|x| = \infty$, it is called a bright soliton solution, and the solution repeats itself after a specific domain L, it is called an L-periodic solution.

Due to the wide applicability of this equation in various fields of science, various schemes have been developed for numerical simulation of this equation and researchers are still improving or developing new numerical schemes to get better results with less computational cost. The solution of the equation was obtained

TABLE 3.7: L_∞ errors in the MTCB-DQM solution of example 3.6 at different time levels

ν	t=0.1					
	Mittal and Jain [122]		MCB-DQM [70]		MTCB-DQM	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
10^{-2}	4.41E-3	3.55E-3	3.89EE-3	3.41E-3	1.76E-3	1.60E-3
10^{-3}	4.60E-5	3.72E-5	4.09E-5	3.55E-5	1.84E-5	1.64E-5
10^{-4}	4.62E-7	3.74E-7	4.11E-7	3.56E-7	1.85E-7	1.65E-7
10^{-5}	4.62E-9	3.74E-9	4.11E-9	3.56E-9	1.85E-9	1.65E-9
10^{-6}	4.62E-11	3.74E-11	4.11E-11	3.56E-11	1.85E-11	1.65E-11
ν	t=1					
	Mittal and Jain [122]		MCB-DQM [70]		MTCB-DQM	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
10^{-2}	3.13E-3	2.66E-2	2.92E-2	2.63E-2	1.24E-2	1.29E-2
10^{-3}	4.45E-4	3.59E-4	3.93E-4	3.45E-4	1.76E-4	1.60E-4
10^{-4}	4.61E-6	3.72E-6	4.09E-6	3.55E-6	1.84E-6	1.64E-6
10^{-5}	4.62E-8	3.74E-8	4.11E-8	3.56E-8	1.85E-8	1.65E-8
10^{-6}	4.62E-10	3.74E-10	4.11E-10	3.56E-10	1.85E-10	1.65E-10

by finite difference schemes [161, 162]. The fully implicit finite difference schemes along with the Barakat and Clark type explicit formula was developed by Deghan and Shokri [163]. The authors have also implemented a scheme based on collocation and radial basis functions to solve this equation numerically. An implicit semi-discrete compact scheme of the higher order was used by Kalita et al. [164]. A split-step Fourier pseudospectral scheme was developed by Weideman et al. [165]. For studying the behavior of the singular solutions of the two-dimensional

TABLE 3.8: Errors norms in the MTCB-DQM solution of example 3.7 at different time levels with $h = 0.02$, $\Delta t = 0.002$ and $\nu = 0.000666666$

	Mittal and Jain [122]		Raslan[146]		MTCB-DQM	
Time	L_∞	L_2	L_∞	L_2	L_∞	L_2
1.68	1.48E-6	2.40E-7	1.63E-3	2.41E-4	1.13E-6	1.82E-7
2.10	1.97E-6	3.27E-7	1.63E-3	2.41E-4	1.18E-6	1.97E-7
2.66	2.65E-6	4.52E-7	1.63E-3	2.41E-4	1.23E-6	2.13E-7
3.30	3.45E-6	6.06E-7	1.63E-3	2.41E-4	1.27E-6	2.28E-7

TABLE 3.9: Errors norms in the MTCB-DQM solution of example 3.7 at different time levels with $h = 0.02$, $\Delta t = 0.002$ and $\nu = 0.002$

	Mittal and Jain [122]		Raslan[146]		MTCB-DQM	
Time	L_∞	L_2	L_∞	L_2	L_∞	L_2
1.68	1.72E-5	3.21E-6	4.88E-3	7.22E-4	3.98E-6	7.63E-7
2.10	2.23E-5	4.31E-6	4.87E-3	7.22E-4	4.07E-6	8.12E-7
2.66	2.92E-5	5.88E-6	4.86E-3	7.22E-4	4.15E-6	8.65E-7
3.30	3.73E-5	7.78E-6	4.85E-3	7.22E-4	4.20E-6	9.13E-7

cubic NLS equation, the spectral method was implemented by Sulem [166]. Several schemes were used by Taha and Ablowitz [167] to study the analytical and numerical solutions of the linear and nonlinear one-dimensional NLS equation that includes finite element approaches and various implicit and explicit finite difference schemes. Chang et al. [168] used an inverse scattering transform scheme to find the solution of the generalized NLS equation. Gardner et al. [169] applied various finite element based schemes for the solution of the one dimensional NLS equation. To find a numerical solution to NLS equation, a Hermite function based pseudospectral approach was developed by Muruganandam et al. [170]. Perez-Garcia et al. [171] solved the NLS equation in one dimension by using several

numerical methods based on finite difference schemes and also done the comparative study of the schemes regarding accuracy. In last few years, the equation has also been solved by a time splitting Fourier spectral approximation [172], an explicit finite difference method [173], Hermite functions based Galerkin spectral scheme [174], spectral collocation method [175], finite difference method [168, 176], radial basis functions [163], compact boundary value method [177] and Chebyshev Spectral collocation method [178].

3.3.1 Implementation of Numerical Scheme

To compute the numerical solution of equation (3.21), on substituting the approximate values of second space derivatives in equation (3.21) results into a system of ordinary differential equations. To deal with the complex terms in the equation, we have used two approaches named as Method 1 and Method 2, and they are as follows:

3.3.1.1 Method 1

In this approach, one can split the equation (3.21) by expanding in real and imaginary parts. This reduces the equation (3.21) into the system of coupled equations, given as:

$$\begin{aligned} -v_t &= \alpha(u_{xx}) + \beta(u^2+v^2)u + F(x,t)u, \\ u_t &= \alpha(v_{xx}) + \beta(u^2+v^2)v + F(x,t)v. \end{aligned} \tag{3.27}$$

Now on substituting the values of first and second derivatives, we get:

$$\begin{aligned} -v_t &= \alpha(\sum_{j=1}^{N} b_{ij}u(x_j,t) + \sum_{j=1}^{N} b'_{ij}u(x_j,t)) + \beta(u^2+v^2)u + F(x,t)u \\ u_t &= \alpha(\sum_{j=1}^{N} b_{ij}v(x_j,t) + \sum_{j=1}^{N} b'_{ij}v(x_j,t)) + \beta(u^2+v^2)v + F(x,t)v \end{aligned} \tag{3.28}$$

where the values of a_{ij} and b_{ij} are same as defined in section 3.2.1. On substituting these values in equation (3.28), we get a system of coupled ordinary

differential equations (ODEs) which can be solved by a modified form of Runge Kutta method called as strong stability-preserving time-stepping Runge-Kutta (SSP-RK43) scheme [72].

3.3.1.2 Method 2

In this approach, substituting the approximation of first and second order of the spatial derivatives directly in the equation (3.21) results in the following system:

$$i\ U_t = \alpha \sum_{j=1}^{N} b_{ij}\ U_j + \beta\ |U_j|U_j + F(x,t)\ U_j = R(U) \tag{3.29}$$

where the values of a_{ij} and b_{ij} are same as defined in section 3.2.1. On substituting these values in equation (3.28), we get a system of ordinary differential equations (ODEs) which is solved by strong stability-preserving time-stepping Runge-Kutta (SSP-RK43) scheme [72].

3.3.2 Stability of the scheme

To check the stability of the scheme one can convert the given equations (3.21) to the system of ODEs and stability of the obtained system can be verified using the matrix method. On substituting the approximate values of derivatives in the Schrödinger equation and taking the nonlinear terms as constant. These equations can be written in the simplified form based on the application of methods which are as follows:
In method 1

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ \\ v \end{bmatrix} = B[u \quad v]^T + f(u,v) \tag{3.30}$$

where the value of $B = \begin{bmatrix} \alpha\ b_{ij} & 0 \\ 0 & \alpha\ b_{ij} \end{bmatrix}$ and $f(u, v)$ represents nonliear terms of the equation.

In method 2

$$U_t = BU + f(U) \tag{3.31}$$

where value of $B = \frac{\alpha \sum_{j=1}^{N} b_{ij}}{i}$ and $f(U)$ represents the nonlinear terms in the equation. The stability of the system (3.30) and (3.31) depends on the eigenvalues of B. Since all eigenvalues of the matrix B are real and negative as presented in Figures 3.12 and 3.13. Hence it can be concluded that the scheme is stable.

3.3.3 Numerical experiment and discussion

In this section, we have solved four test problems of Schrödinger equation, and also examined the numerical results. The accuracy and efficiency of the developed method is demonstrated by evaluating the discrete average error and L_∞ error.

Example 3.8. *For $\alpha = 1$, $\beta = -2$ and $F(x, t) = 0$, consider the equation (3.21) in following form:*

$$i\ U_t(x, t) = U_{xx}(x, t) - 2|U(x, t)|^2 U(x, t)$$

with $x \in [-1, 1]$ and $t > 0$. The initial and the boundary conditions can be obtained from the exact solution given by [175]:

$$U(x, t) = exp(i(2x - 3t))sech(x - 4t)$$

The L_∞ error is computed using both of the proposed methods at different times $t \leq 4.5$ with $h = 0.02$ and $\Delta t = 0.0001$. Numerical results are compared with Javidi [175] and enlisted in Table 3.10. Comparison of numerical and the analytical solutions using both of the methods are presented in form of Figures 3.14 and 3.15.

Example 3.9. *The NLS equation (3.21) is solved with* $\alpha = -\frac{1}{2}$, $\beta = 1$ *and* $F(x,t) = cos^2(x)$, *that results in:*

$$i\ U_t(x,t) = -\frac{1}{2}U_{xx}(x,t) + |U(x,t)|^2 U(x,t) + cos^2(x)U(x,t)$$

on $x \in [0, 2\pi]$, $t > 0$. *The initial and the boundary conditions can be obtained from exact solution given by [175]:*

$$U(x,t) = sin(x)exp(-\frac{3it}{2})$$

L_∞ *error is computed using both of the methods at different times* $t \leq 20$ *with* $N = 100$, $\Delta t = 0.0001$, *and shown in Table 3.11. Comparison of numerical and the analytical solution using both of the methods are presented in form of Figures 3.16 and 3.17.*

Example 3.10. *Consider the one soliton solution of the equation (3.21) with* $\alpha = 1$, $\beta = 2$ *and* $F(x,t) = 0$,

$$i\ U_t(x,t) = U_{xx}(x,t) + 2|U(x,t)|^2 U(x,t)$$

with $x \in [-15, 15]$ *and* $t > 0$. *Initial and boundary conditions are calculated from the exact solution given by [168]:*

$$U(x,t) = exp(-i(2x + 4 - 3t))sech(x + 2 - 4t)$$

To show the accuracy of the method, L_∞ *error is computed at different time* $t \leq 3$ *using both of the methods with* $N = 100$ *and* $\Delta t = 0.0001$, *and shown in Table 3.12. Comparison of numerical and the exact solution using both of the methods are presented in form of Figures 3.18 and 3.19.*

Example 3.11. *Consider the collision of two solitons solution of the equation (3.21) with* $\alpha = 1$, $\beta = 2$ *and* $F(x,t) = 0$,

$$i\ U_t(x,t) = U_{xx}(x,t) + 2|U(x,t)|^2 U(x,t)$$

with $x \in [-15, 15]$ *and* $t > 0$. *The initial and boundary conditions for this equation can be obtained from the exact solution [168]:*

$$U(x,t) = exp(-i(2x-20-3t))sech(x-10-4t)+exp(i(2x+20+3t))sech(x+10+4t)$$

In computation L_∞ *error is computed using both of the methods at time* $t \leq 3$ *with* $N = 100$ *and* $\Delta t = 0.0001$ *and shown in Table 3.13. Comparison of numerical and the exact solution using both of the methods are presented in form of Figures 3.20 and 3.21.*

3.3.4 Summary

In recent years, due to the existence of soliton solution of NLS equation, numerical simulation has become popular among researchers. Several distinctive methods were developed and implemented to solve this equation. As per the author's knowledge, it is for the first time that the modified trigonometric cubic B-spline basis function has been used with differential quadrature method to solve NLS equation in one dimension. To establish the efficiency of the scheme, the approximated results are compared with the exact solutions as well as with numerical solutions obtained by other researchers. From the obtained discrete average error and L_∞ error, it can be inferred that the obtained numerical results are in great congruity with the analytical solution.

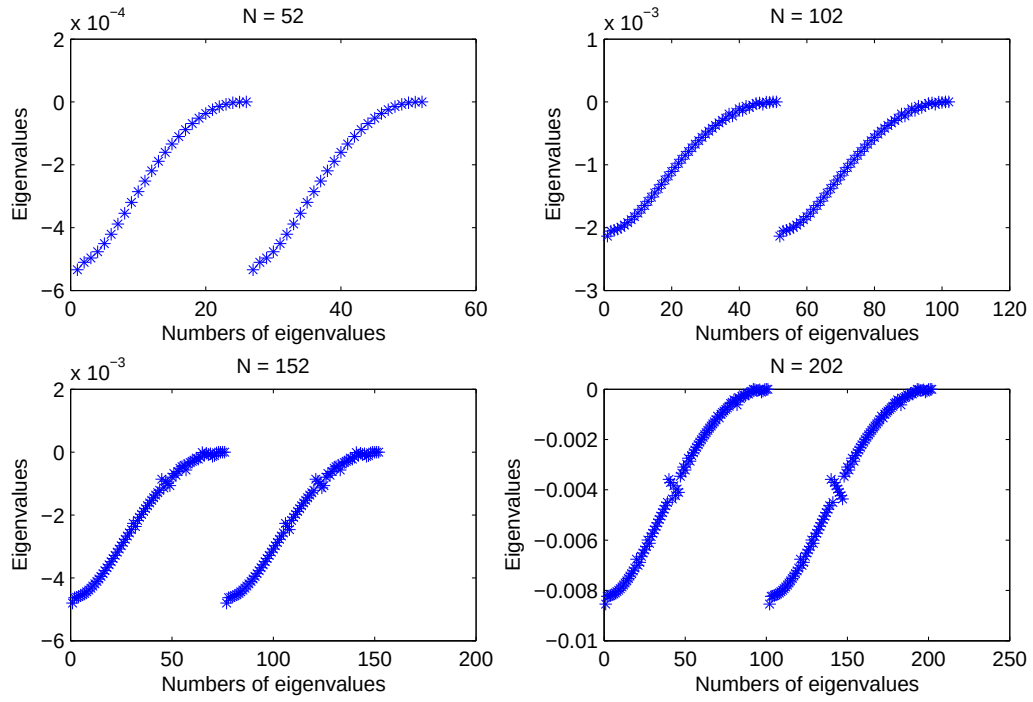

FIGURE 3.12: Eigenvalues of the matrix B in method 1 for different partitions of the domain in one dimension

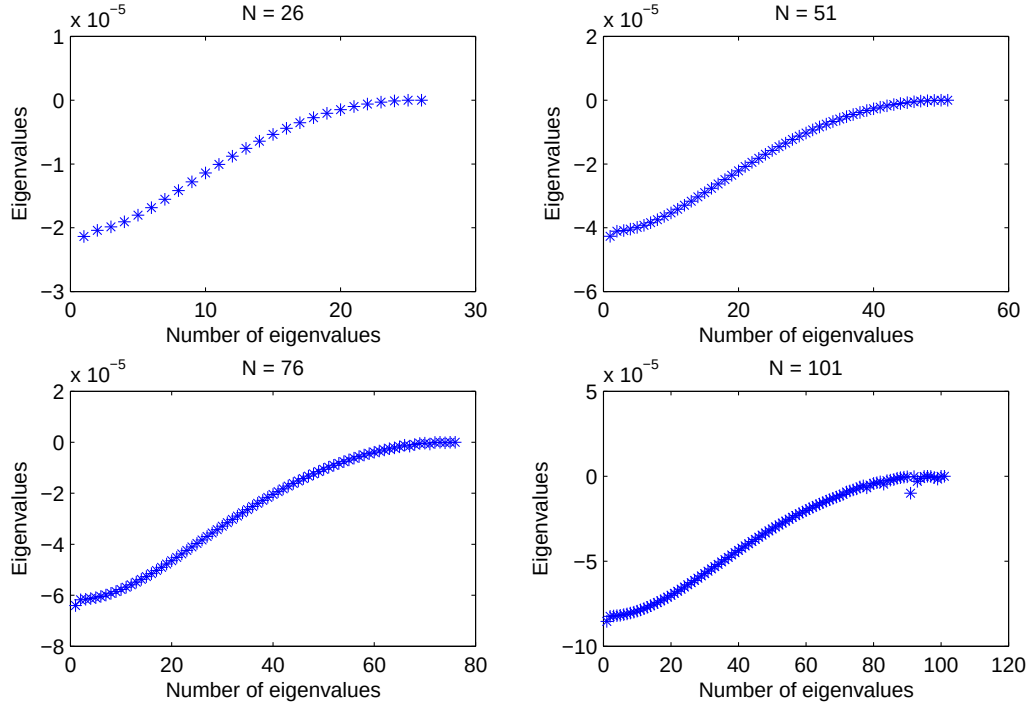

FIGURE 3.13: Eigenvalues of the matrix B in method 2 for different partitions of the domain in one dimension

TABLE 3.10: Comparison of errors obtained by applying both of the methods in example 3.8 for $h = 0.02$, $\Delta t = 0.0001$ at different time levels

t	Method 1			Method 2	Javidi[175]
	L_∞ for u	L_∞ for v	L_2 for U	L_∞ for U	L_2 for U (h=0.125 and Δt = 0.0005)
0.5	6.03E-4	3.93E-4	5.28E-4	1.82E-3	7.08E-6
1	3.48E-4	2.90E-4	3.37E-4	3.09E-4	–
1.5	2.42E-4	4.51E-4	3.54E-4	4.53E-4	7.08E-6
2	3.96E-4	2.96E-4	3.54E-4	4.59E-4	–
2.5	3.55E-4	4.16E-4	3.54E-4	4.21E-4	7.09E-6
3	3.44E-4	2.21E-4	3.54E-4	4.05E-4	–
3.5	2.78E-4	3.01E-4	3.53E-4	3.92E-4	7.09E-6
4	3.70E-4	3.09E-4	3.54E-4	4.60E-4	–
4.5	2.83E-4	3.01E-4	3.54E-4	3.75E-4	7.10E-6

TABLE 3.11: Comparison of errors obtained by applying both of the methods in example 3.9 for $N = 100$, $\Delta t = 0.0001$ at different time levels

t	Method 1		Method 2
	L_∞ for u	L_∞ for v	L_∞ for U
1	1.43E-4	1.39E-5	1.69E-4
5	1.25E-4	4.40E-5	6.57E-4
10	7.51E-5	8.77E-5	2.17E-3
20	8.02E-5	1.35E-5	8.26E-3

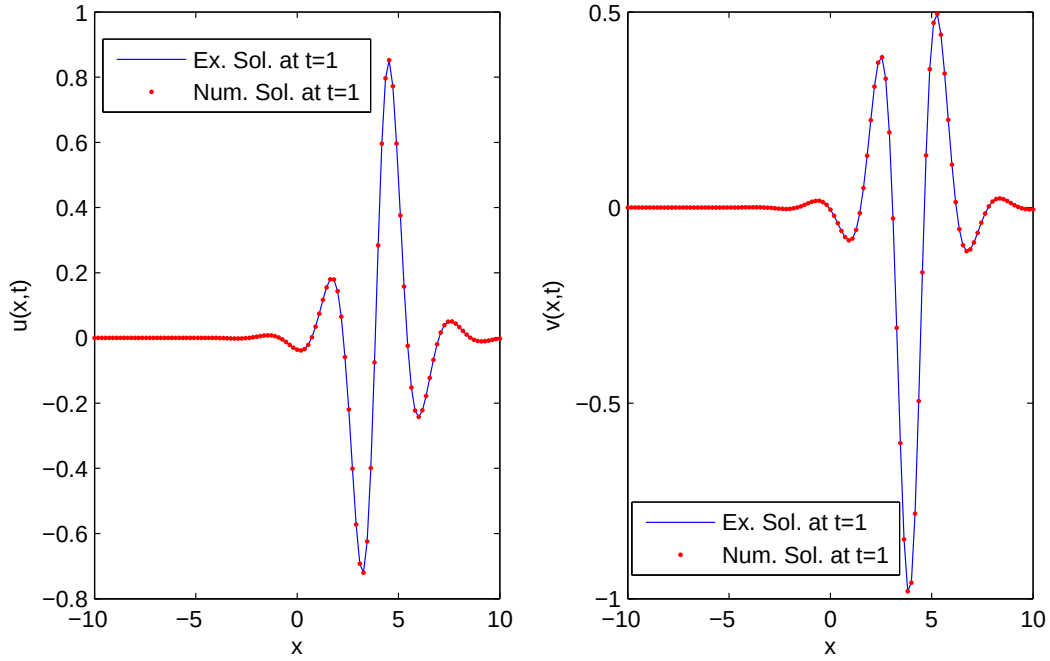

FIGURE 3.14: The comparison of numerical and the exact solutions of example 3.8 for $N = 110$ at $t = 1$ with method 1

TABLE 3.12: Comparison of errors obtained by applying both of the methods in example 3.10 for $N = 200$, $\Delta t = 0.0001$ at different time levels

	Method 1		Method 2
t	L_∞ for u	L_∞ for v	L_∞ for U
0.5	2.17E-4	2.54E-4	2.54E-4
1	1.95E-4	1.84E-4	1.97E-4
1.5	1.51E-4	2.31E-4	2.32E-4
2	2.45E-4	3.32E-4	3.40E-4
2.5	3.97E-4	4.63E-4	4.94E-4
3	6.44E-4	5.64E-4	6.68E-4

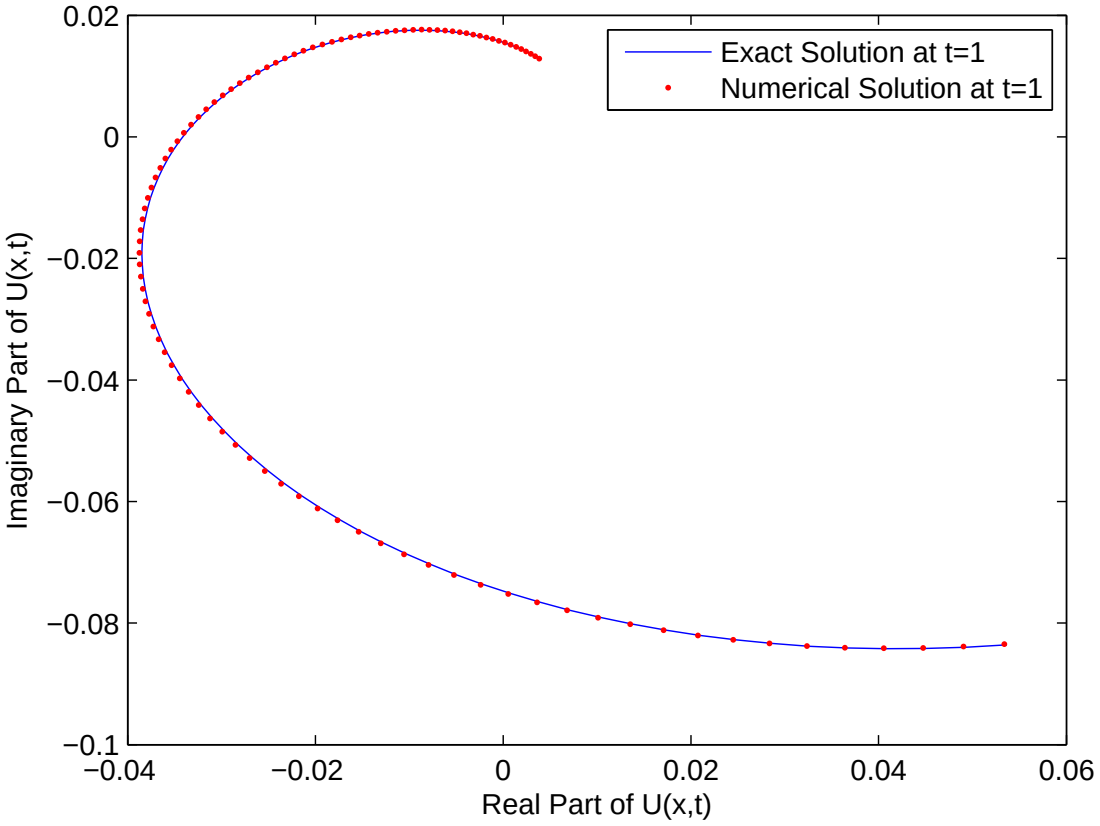

FIGURE 3.15: The comparison of numerical and the exact solutions of example 3.8 for $N = 100$ at $t = 1$ with method 2

TABLE 3.13: Comparison of errors obtained by applying both of the methods in example 3.11 for $N = 350$, $\Delta t = 0.0001$ at different time levels

	Method 1		Method 2
t	L_∞ for u	L_∞ for v	L_∞ for U
0.5	2.72E-4	2.77E-4	2.85E-4
1	2.27E-4	2.47E-4	2.50E-4
1.5	2.06E-4	1.98E-4	2.10E-4
2	4.81E-4	1.83E-4	4.87E-4
2.5	1.92E-3	2.31E-3	2.35E-3
3	2.27E-3	2.40E-3	2.42E-3

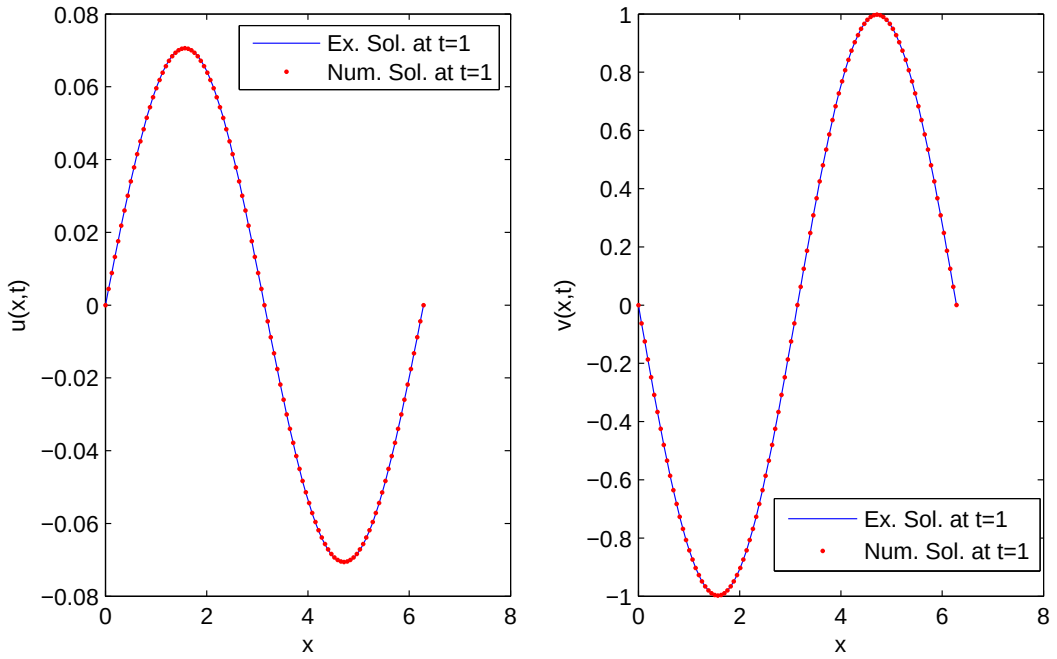

FIGURE 3.16: The comparison of numerical and the exact solutions of example 3.9 for $N = 100$ at $t = 1$ with method 1

3.4 Introduction to telegraph equation

The telegraph equation is used to sculpt reaction-diffusion in many branches of sciences and engineering. This equation is the basis for fundamental equations of atomic physics and used to model the vibrations of structures, e.g. buildings, beams, and machines. It also arises in the study of pulsating blood flow in arteries, in 1D random motion of bugs along a hedge [179], and play an important role in modeling several relevant problems such as signal analysis [180], wave propagation [181], random walk theory [182] etc. This equation is commonly used in the signal analysis for transmission and propagation of electrical signals [183] and also has applications in various other fields [184]. The 1D hyperbolic telegraph equation is given by:

$$u_{tt} + 2\alpha u_t + \beta^2 u = u_{xx} + f(x,t), \quad x \in [a,b], t \geq 0, \tag{3.32}$$

subject to the initial conditions (ICs)

$$u(x,0) = g_1(x), \quad u_t(x,0) = g_2(x), \quad x \in [a,b], \tag{3.33}$$

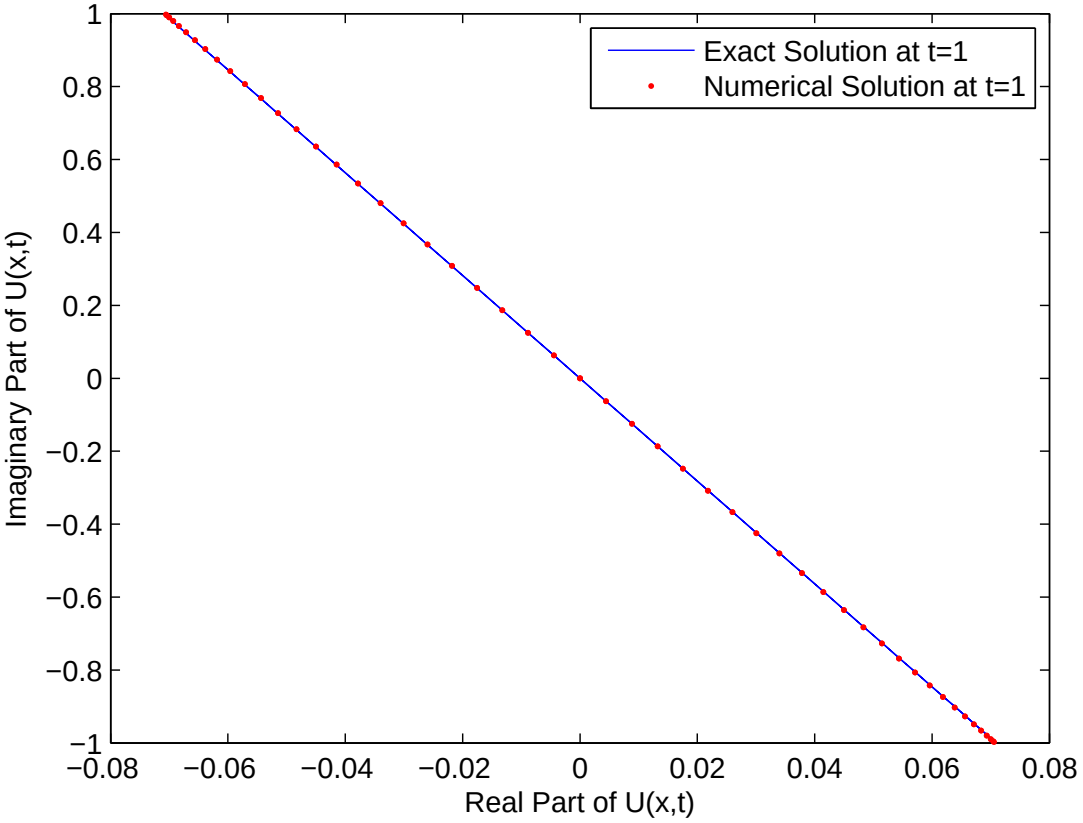

FIGURE 3.17: The comparison of numerical and the exact solutions of example 3.9 for $N = 100$ at $t = 1$ with method 2

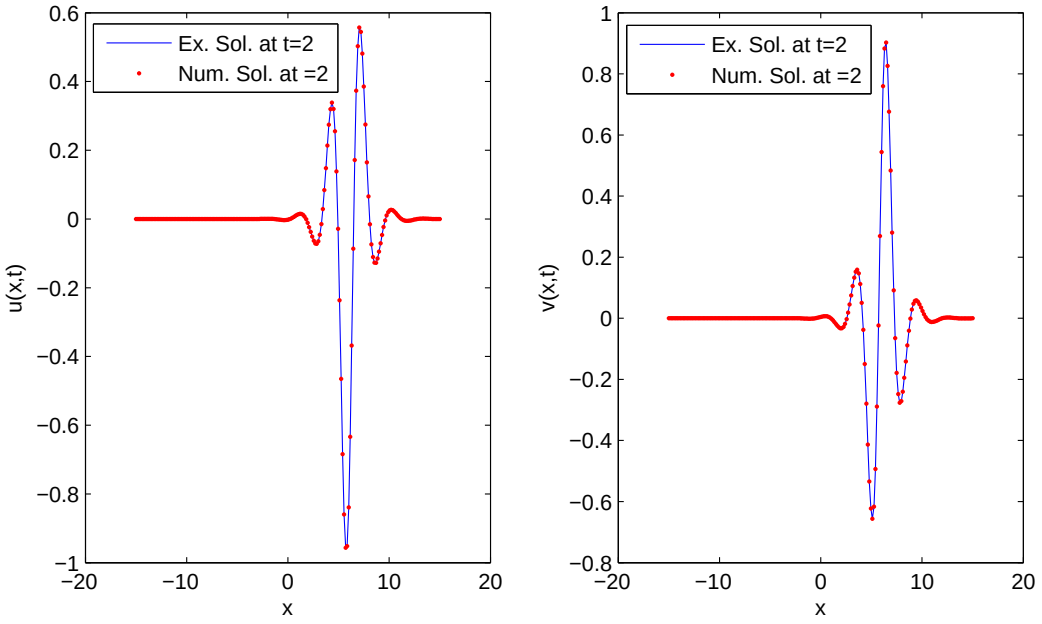

FIGURE 3.18: The comparison of numerical and the exact solutions of example 3.10 for $N = 200$ at $t = 2$ with method 1

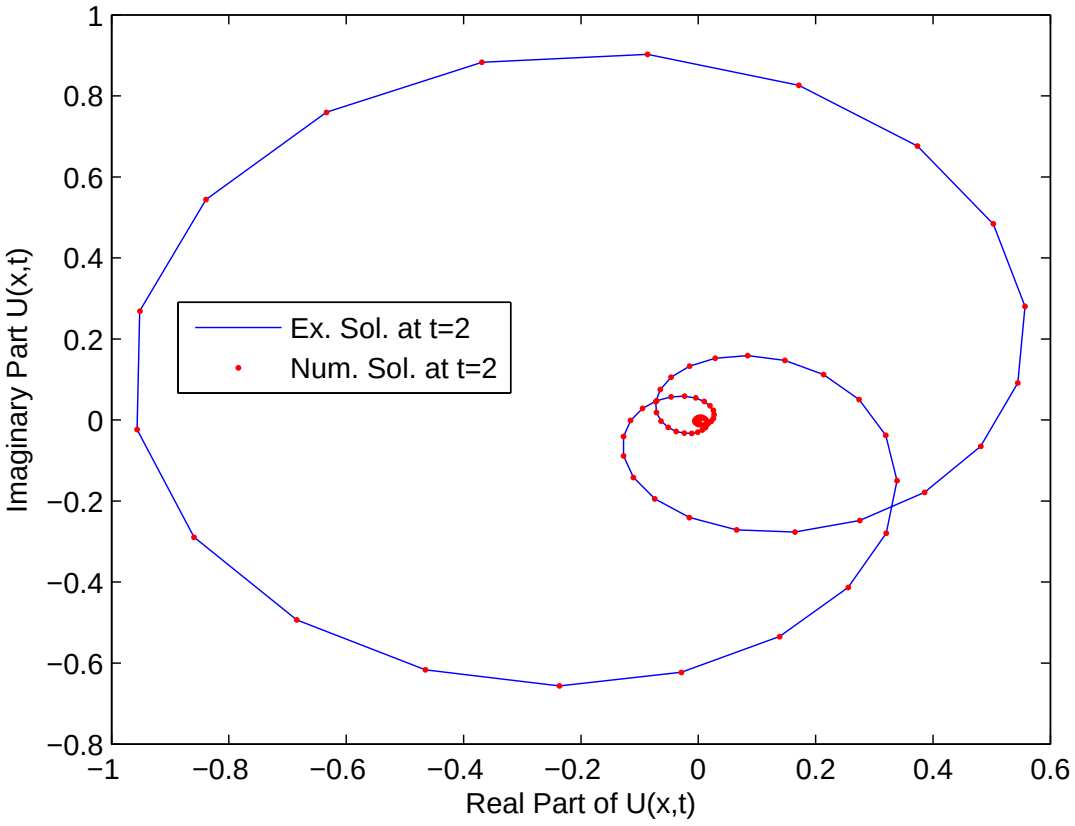

FIGURE 3.19: The comparison of numerical and the exact solutions of example 3.10 for $N = 200$ at $t = 2$ with method 2

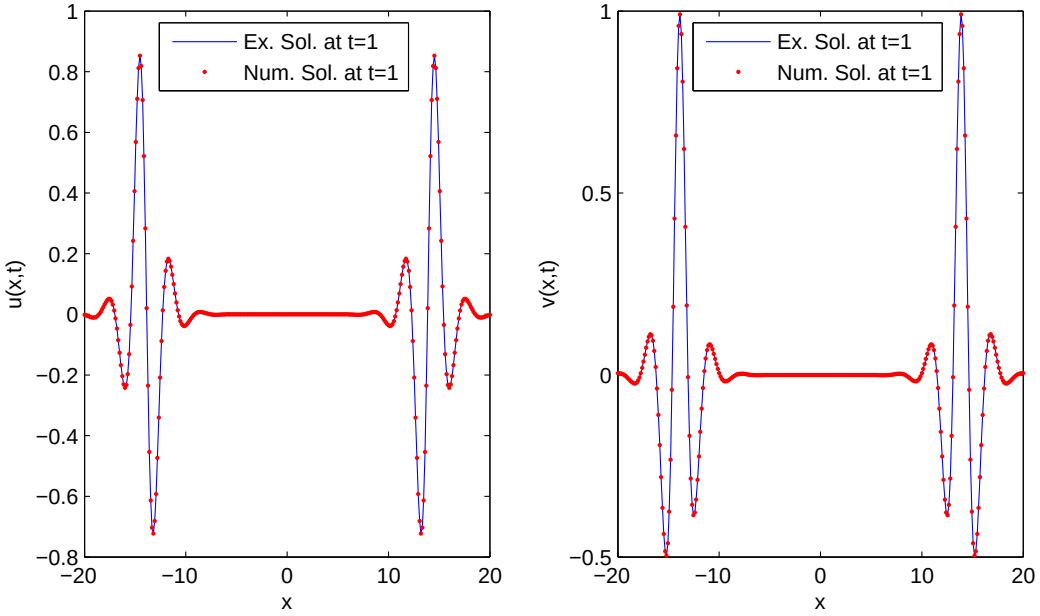

FIGURE 3.20: The comparison of numerical and the exact solutions of example 3.11 for $N = 300$ at $t = 1$ with method 1

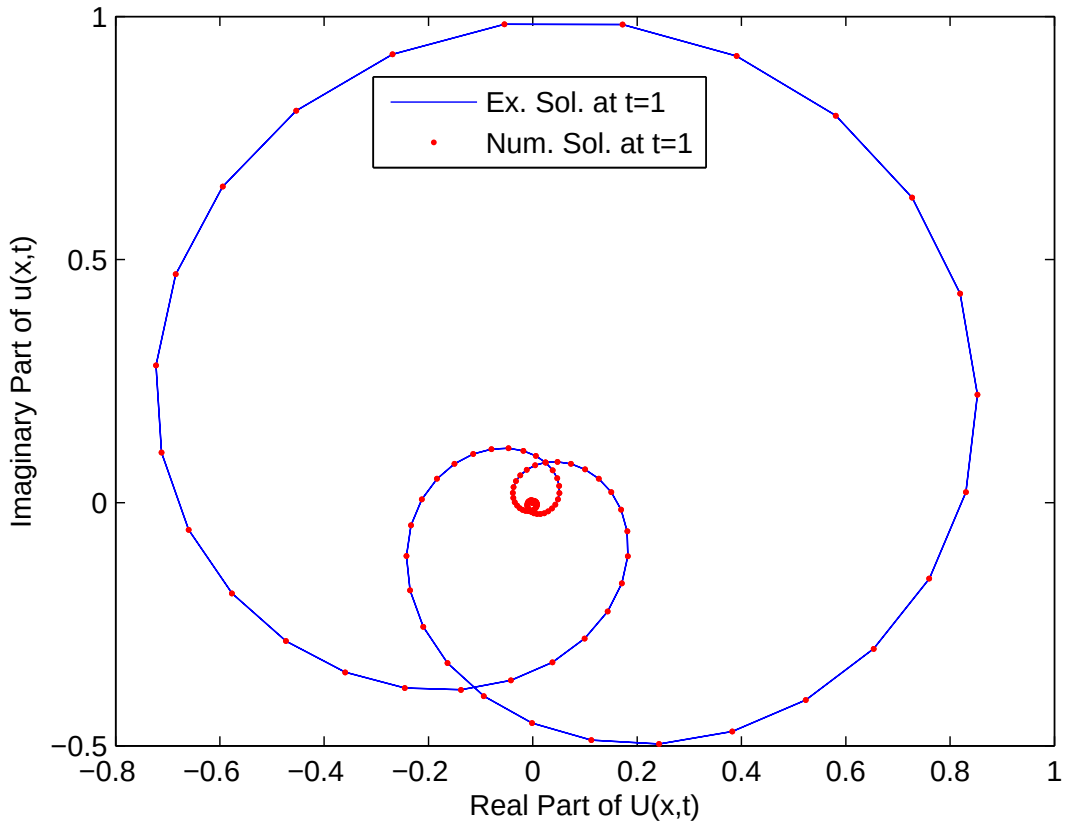

FIGURE 3.21: The comparison of numerical and the exact solutions of example 3.11 for $N = 350$ at $t = 1$ with method 2

and the Dirichlet boundary conditions (BCs)

$$u(a,t) = \psi_1(t), \quad u(b,t) = \psi_2(t), \quad t > 0. \tag{3.34}$$

where $f, g_1, g_2, \psi_1, \psi_2$ are known functions, and $u_{tt} = \frac{\partial^2 u}{\partial t^2}, u_t = \frac{\partial u}{\partial t}, u_{xx} = \frac{\partial^2 u}{\partial x^2}$. Equation (3.32) with the coefficients $\alpha > 0, \beta = 0$ represents a damped wave equation. It represents the telegraph equation if $\alpha > \beta > 0$. The telegraph equation (3.32) is also referred to model the mixture between diffusion and wave propagation by introducing a term that accounts for effects of finite velocity to standard heat or mass transport equation [185].

In recent years, several numerical schemes to solve telegraph equation have been developed by many researchers. The existence of its double periodic solution has been investigated by Kim [186]. The unconditionally stable schemes for telegraph equation are proposed by Mohanty [187, 188] and H. W. Liu and L. B. Liu [189], etc. Further, the schemes proposed in [190–192] are conditionally stable. Mohebbi

and Dehghan [193] combined a high-order compact finite difference scheme to approximate the spatial derivative and the collocation technique for time component to solving the equation. Chebyshev cardinal functions solutions to the equation are obtained in [194]. Some more schemes to solve the telegraph equation available in the literature are: interpolating scaling functions [195], Chebyshev Tau method [196], Rothe-Wavelet method [185], semi-discretion methods [197], explicit difference method [192], dual reciprocity boundary integral equation (DRBIE) method [198], collocation scheme with thin plate splines radial basis function (RBFCM) [199], polynomial differential quadrature method (PDQM)[200], cubic B-spline quasi-interpolation (CBQ) [201], cubic B-spline collocation method (CBCM) [202], quartic B-spline collocation method (QBCM) [203], collocation method based on modified cubic B-spline (CMMCB) [204] etc.

3.4.1 Implementation of the numerical scheme on telegraph equation

Using the transformation: $u_t = v$, the equation (3.32) is first converted into a system of PDEs as follows:

$$\left.\begin{aligned} u_t(x,t) &= v(x,t), \\ v_t(x,t) &= u_{xx}(x,t) - 2\alpha v(x,t) - \beta^2 u(x,t) + f(x,t). \end{aligned}\right\} \tag{3.35}$$

Now, on substituting the values of the second order approximation of the spatial derivatives, obtained by using MCTB-DQM, equation (3.35) can be rewritten as

$$\left.\begin{aligned} u_t(x_i,t) &= v(x_i,t), \\ v_t(x_i,t) &= \sum_{j=1}^{N} b_{ij} u(x_j,t) - 2\alpha v(x_i,t) - \beta^2 u(x_i,t) + f(x_i,t), \end{aligned}\right\} \quad i \in \{1,\ldots,N\}, \tag{3.36}$$

and hence, equation (3.36) is reduced into a coupled system of first-order ODEs in time, that is,

$$\frac{du_i}{dt} = v_i, \text{ and } \frac{dv_i}{dt} = L(u_i), i = \{1, \ldots, N\}. \tag{3.37}$$

subject to the BCs as defined in (3.34), and the ICs

$$u(x_i, 0) = g_1(x_i), \quad v(x_i, 0) = g_2(x_i), \quad i \in \{1, \ldots, N\}, \tag{3.38}$$

where L is a nonlinear differential operator. The resulting system of ODEs is solved by using SSP-RK43 scheme [72].

3.4.2 Stability of the scheme

To check the stability of the scheme one can convert the given equations (3.35) to the system of ODEs and stability of the obtained system can be verified using the matrix method. On substituting the approximate values of derivatives in the telegraph equation and taking the nonlinear terms as constant, equations can be written in simplified form as follows:

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ \\ v \end{bmatrix} = B[u \quad v]^T + f(u, v) \tag{3.39}$$

where the value of $B = \begin{bmatrix} 0 & I \\ \\ b_{ij} & 0 \end{bmatrix}$ and $f(u, v)$ is nonliear terms of the equation. The stability of the system (3.39) depends on the eigenvalues of B. Since all eigenvalues of the matrix B satiesfy the conditions of [73] and presented in Figure 3.22. Hence it can be concluded that the scheme is stable.

3.4.3 Numerical experiments and discussions

In this section, three examples are considered to compute the numerical solutions by the proposed method. The existence of analytical solutions helps to measure the accuracy of numerical methods. The accuracy and the efficiency of the method are measured by evaluating the L_2 and L_∞ error norms. These test problems are as follows:

Example 3.12. *The telegraph equation* (3.32) *over the region* $[0, \pi]$ *is considered with the initial and boundary conditions:*

$$u(x,0) = \sin(x), \qquad u_t(x,0) = -\sin(x),$$

$$u(0,t) = 0, \qquad u(\pi,t) = 0, \quad t \geq 0,$$

and the function

$$f(x,t) = (2 - 2\alpha + \beta^2)e^{-t}\sin(x).$$

The exact solution [197, 198] is given by:

$$u(x,t) = e^{-t}\sin(x). \tag{3.40}$$

The comparison of the L_2 *and* L_∞ *errors at different time levels* $t \leq 2$ *is done for different values of* α *and* $\beta = \sqrt{2}$ *as follows:*

(*a*) *For* $\alpha = 2$, L_2 *and* L_∞ *errors with* $h = 0.02, \triangle t = 0.01$ *are compared with the errors of well known earlier schemes: RBFCM [199] and CMMCB[204], and are reported in Table 3.14. The physical behavior of obtained solution at different time levels* $t \leq 2$ *is shown in Figure 3.23 with* $h = 0.01, \triangle t = 0.01$.

(*b*) *For* $\alpha = 3$, L_2 *and* L_∞ *errors with* $h = 0.02, \triangle t = 0.0001$ *are compared with the errors by RBFCM [199] of Dehghan and Shokri, and are reported in Table 3.15. The physical behavior of the computed solutions with* $\triangle t = 0.0001$ *is depicted in Figure 3.24 at different time levels* $t \leq 2$.

It is evident from the obtained results that the errors are decreasing as time increasing and the numerical results are more accurate than the solutions obtained in RBFCM [199] and CMMCB [204].

Example 3.13. *The telegraph equation (3.32) is considered over the region* $[0,1]$ *for* $\alpha = 0.5, \beta = 1$ *with the initial and boundary conditions*

$$u(x,0) = 0, \qquad u_t(x,0) = 0,$$

$$u(0,t) = 0, \qquad u(1,t) = 0, t \geq 0,$$

and the function

$$f(x,t) = (2 - 2t + t^2)(x - x^2)e^{-t} + 2t^2e^{-t}.$$

The exact solution is given in [196, 199] as:

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

The comparison of the L_2 *and* L_∞ *errors is done at different time levels* $t \leq 2$ *with time step* $\triangle t = 0.0001$ *and* $h = 0.0125$ *with the errors obtained by the earlier schemes: RBFCM[199], QBCM [203] and CMMCB[204], and are reported in Table 3.16. It is evident that obtained results are better than the results obtained in [199, 203, 204]. The physical behavior of the obtained solution at different time levels* $t \leq 5$ *is shown in Figure 3.25.*

Example 3.14. *The telegraph equation (3.32) over the region* $[0,2]$ *is considered with the initial and boundary conditions:*

$$u(x,0) = \tan(x/2), \qquad u_t(x,0) = \frac{1}{2}(1 + \tan^2(x/2)),$$

$$u(0,t) = \tan(t/2), \qquad u(2,t) = \tan(\frac{2+t}{2}), t \geq 0$$

and the function

$$f(x,t) = \alpha(1 + \tan^2(\frac{x+t}{2})) + \beta^2 \tan(\frac{x+t}{2}).$$

The exact solution is given in [193] as:

$$u(x,t) = \tan(\frac{x+t}{2})$$

The numerical solutions are obtained for $\alpha = 10, \beta = 5$ *taking* $\triangle t = 0.001, 0.0001$ *and* $h = 0.025$ *at different time-levels* $t \leq 1$. *The comparison of* L_∞ *error at different time levels with the recent schemes [201, 203, 204] is reported in Table*

3.17. It is evident from Table 3.17 and 3.18 that obtained results are in good agreement with the exact solutions and better than the results obtained in [201, 203, 204]. The physical behavior of the obtained solutions are depicted in Figure 3.26 at different time levels $t \leq 1.0$ *taking* $h = 0.025, \triangle t = 0.0001$.

3.4.4 Summary

The solution of the hyperbolic telegraph equation is obtained numerically in this chapter using MTCB-DQM. This scheme is based on the DQM combined with modified cubic trigonometric B-spline as basis functions. On implementing the scheme and substituting the derivatives, set of ODEs is obtained, which is solved using SSP-RK43. The efficiency and precision of the proposed method are revealed by three test problems. The numerical results, L_2 and L_∞ errors are compared with numerical solutions from literature and are found to be in decent agreement with formerly obtained results. The advantage of the developed method is easy to implement and reduced data complexity as compared to the old schemes present in the literature.

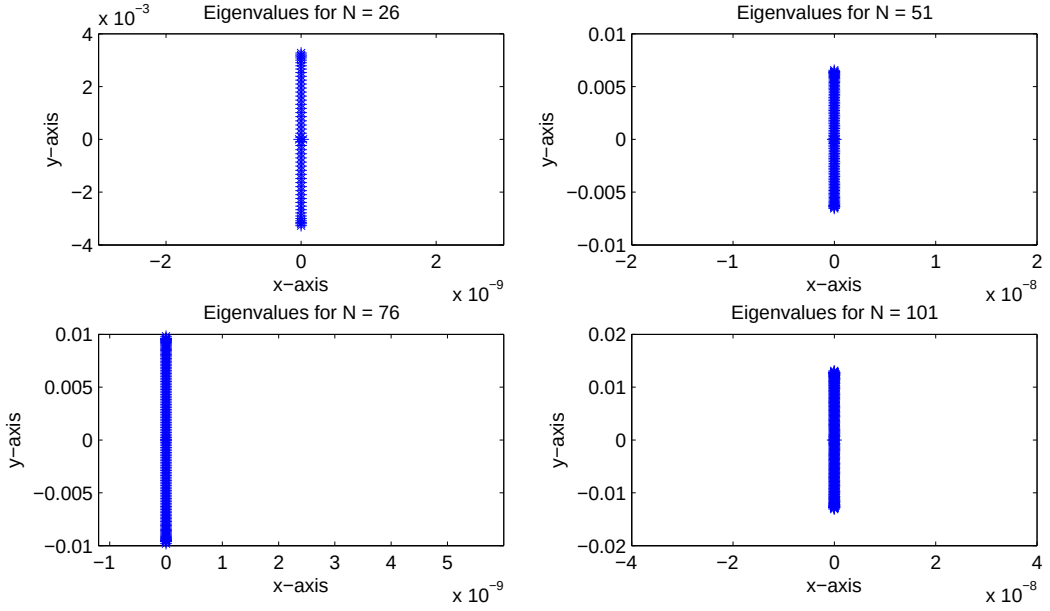

FIGURE 3.22: Eigenvalues of the matrix B for different partitions of the domain in one dimension

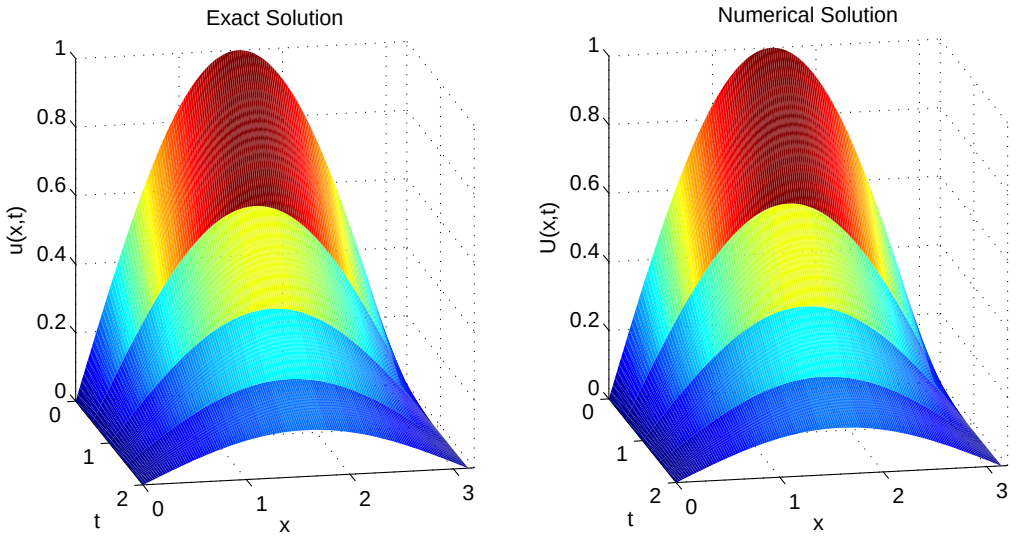

FIGURE 3.23: The physical behavior of exact and numerical solutions of example 3.12 with $\alpha = 2, \beta = \sqrt{2}$ at different time levels using trigonometric B-spline with differential quadrature for $t \leq 2$

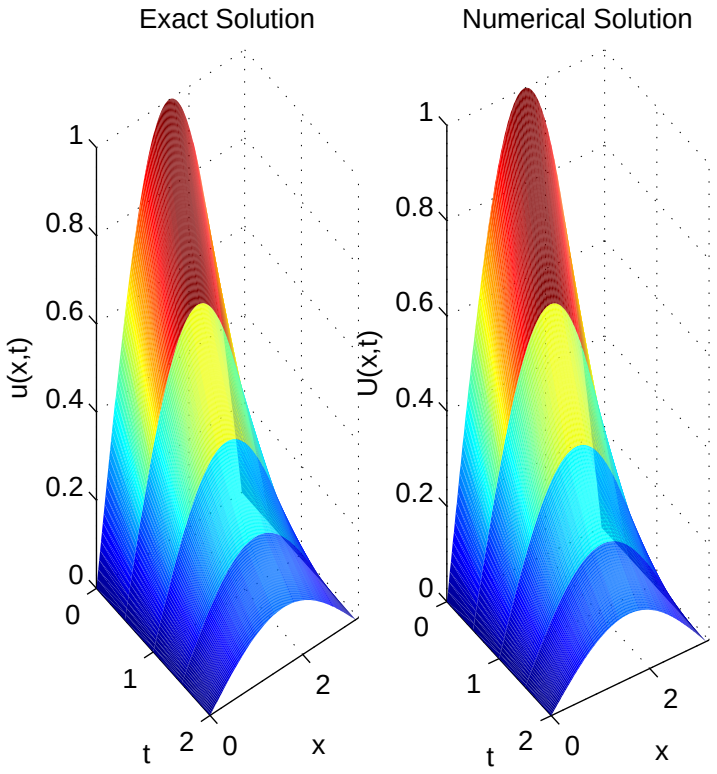

FIGURE 3.24: The physical behavior of exact and numerical solutions of example 3.12 with $\alpha = 3, \beta = \sqrt{2}$ at different time levels using trigonometric B-spline with differential quadrature for $t \leq 2$

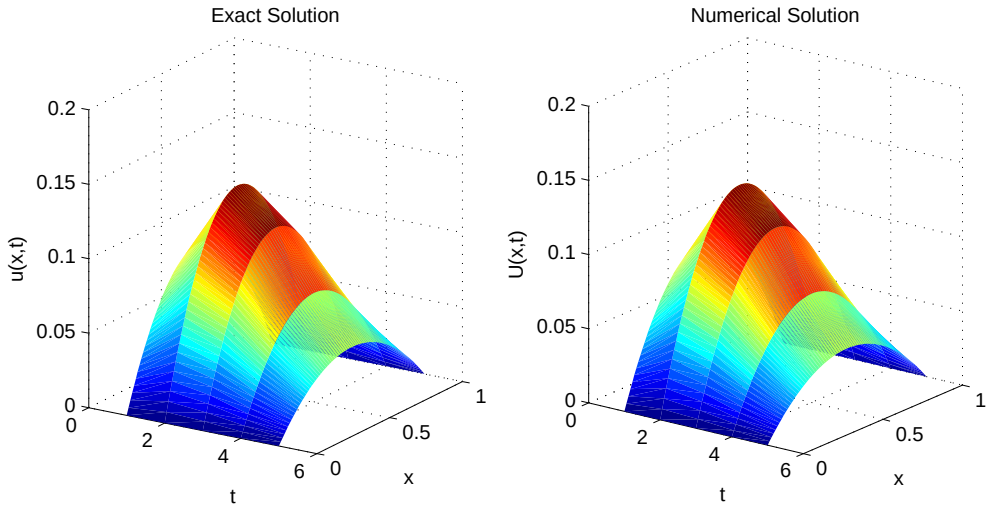

FIGURE 3.25: The physical behavior of exact and numerical solutions of example 3.13 with $\triangle t = 0.0001$ and $h = 0.0125$ at different time levels using trigonometric B-spline with differential quadrature for $t \leq 5$

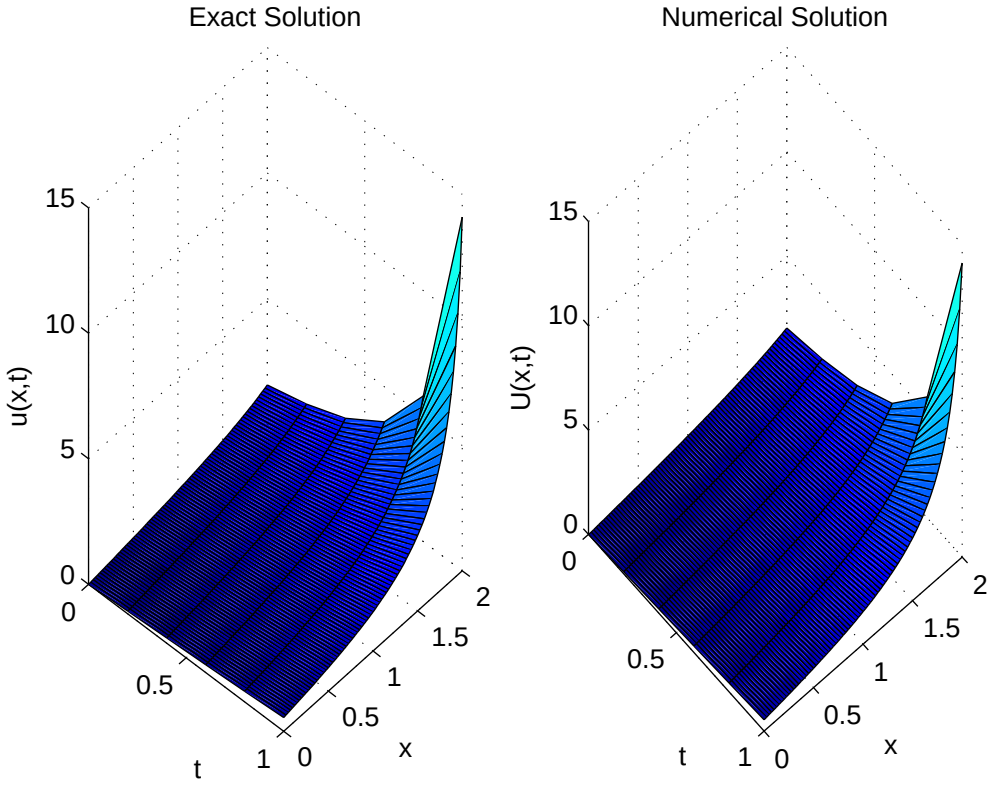

FIGURE 3.26: The physical behavior of exact and numerical solutions of example 3.14 taking $\triangle t = 0.0001$ and $h = 0.025$ at different time levels using trigonometric B-spline with differential quadrature for $t < 1$

TABLE 3.14: Comparison of L_2 and L_∞ errors at $t \leq 2$ with $h = 0.02$ and $\triangle t = 0.01$

	L_2				L_∞			
t	MTCB-DQM	MCB-DQM	CMMCB [204]	RBFCM [199]	MTCB-DQM	MCB-DQM	CMMCB [204]	RBFCM [199]
0.5	1.4022E-7	2.1570E-7	2.3328E-6	7.9491E-5	1.9891E-7	3.5160E-7	1.8612E-6	8.3721E-6
1.0	1.059E-7	1.6491E-7	4.3667E-6	1.4554E-4	1.1694E-7	2.1430E-7	3.4839E-6	1.5680E-5
1.5	7.2312E-8	1.1434E-7	4.7817E-6	1.5895E-4	7.1148E-8	1.3045E-7	3.8251E-6	1.7412E-5
2.0	5.0662E-8	8.0521E-8	4.2706E-6	1.4185E-4	4.3259E-8	7.9350E-8	3.4073E-6	1.5813E-5

TABLE 3.15: Comparison of L_2 and L_∞ errors at $t \leq 2$ with $h = 0.02$ and $\triangle t = 0.0001$

	L_2			L_∞		
t	MTCB-DQM	MCB-DQM	RBFCM [199]	MTCB-DQM	MCB-DQM	RBFCM [199]
0.5	5.5700E-6	6.22E-5	1.55E-5	4.4591E-6	4.46E-6	1.41E-4
1.0	9.7571E-6	1.05E-5	2.71E-5	7.8127E-6	7.81E-6	2.46E-4
1.5	1.0903E-5	1.16E-5	3.03E-5	8.7306E-6	8.74E-6	2.74E-4
2.0	1.0407E-5	1.11E-5	2.92E-5	8.3216E-6	8.53E-6	2.62E-4

TABLE 3.16: Comparison of L_2 and L_∞ errors at different time levels ≤ 5 with the errors due to well known earlier schemes

	MTCB-DQM		MCB-DQM		CMMCB[204]		RBFCM[199]		QBCM [203]
	$h = 0.0125, \triangle t = 0.001$		$h = 0.0125, \triangle t = 0.001$		$h = 0.01, \triangle t = 0.001$		$h = 0.01, \triangle t = 0.001$		$h = 0.005$
t	L_2	L_∞	L_2	L_∞	L_2	L_∞	L_2	L_∞	$L_\infty(\triangle t = 0.01)$
1	4.00E-5	5.40E-5	4.2025E-5	5.5884E-5	4.5526E-5	5.9153E-5	1.4386E-4	1.8479E-5	1.9175E-4
2	2.51E-6	5.32E-6	6.4909E-6	9.7153E-6	1.4307E-5	1.7864E-5	8.0879E-5	1.0713E-5	1.1387E-4
3	5.72E-6	8.48E-6	3.9665E-6	7.0063E-6	6.4273E-6	1.4309E-5	1.2944E-4	1.8161E-5	1.7053E-4
4	1.40E-5	2.00E-5	1.1978E-5	1.7648E-5	8.9203E-6	1.3529E-5	1.1845E-4	1.6489E-5	2.0271E-4
5	5.26E-6	7.49E-6	4.2339E-6	6.2693E-6	3.0161E-6	5.2032E-6	7.5545E-5	1.0455E-5	9.8405E-5

TABLE 3.17: Comparison of L_∞ errors in example 3.14 at different time levels $t \leq 1$ with the errors in the earlier schemes

Error	Schemes	$(h, \triangle t)$	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	$t = 1.0$
L_∞	MCB-DQM	(0.025, 0.0001)	4.8544E-5	4.9179E-5	2.2678E-5	4.3378E-4	1.1202E-2
L_∞	MTCB-DQM	(0.025, 0.0001)	2.42E-4	3.80E-4	6.98E-4	1.73E-3	1.64E-2
L_∞	CMMCB [204]	(0.020, 0.0001)	3.4797E-5	5.3456E-5	9.4725E-5	1.8792E-4	5.8720E-4
L_∞	MCB-DQM	(0.025, 0.0010)	9.6264E-4	1.4778E-3	2.5684E-3	5.6119E-3	1.8461E-2
L_∞	MTCB-DQM	(0.025, 0.0010)	2.42E-3	3.80E-3	6.97E-3	1.72E-2	9.92E-2
L_∞	CMMCB [204]	(0.020, 0.0010)	2.6332E-4	6.9997E-4	1.4860E-3	3.4057E-3	1.1148E-2
L_∞	CBQ [201]	(0.005, 0.0010)	1.8918E-4	3.9943E-4	7.9715E-4	1.8799E-3	8.0113E-3
L_∞	QBCM [203]	(0.001, 0.0010)	2.7740E-4	7.0782E-4	1.3848E-3	3.0930E-3	1.3424E-2

TABLE 3.18: Comparison L_2 errors in example 3.14 with the errors obtained by other scheme at different time levels $t \leq 1$

Errors	Schemes	$(h, \triangle t)$	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	$t = 1.0$
L_2	MCB-DQM	(0.025, 0.0001)	1.3150E-5	1.4880E-5	1.0158E-5	8.5657E-5	1.9393E-3
L_2	MTCB-DQM	(0.025, 0.0001)	4.85E-5	8.18E-5	1.40E-4	3.30E-4	3.33E-3
L_2	CMMCB [204]	(0.020, 0.0001)	5.0305E-5	9.5163E-5	2.2041E-4	7.8267E-4	7.9277E-3
L_2	MTCB-DQM	(0.025, 0.001)	5.82E-4	1.02E-3	1.82E-3	3.95E-3	1.76E-2
L_2	MCB-DQM	(0.025, 0.0010)	2.4192E-4	3.8283E-4	6.4500E-4	1.3183E-3	3.9769E-3
L_2	CMMCB [204]	(0.020, 0.0010)	1.8782E-4	4.8888E-4	9.4864E-4	1.8743E-3	5.0981E-3

Chapter 4

Numerical solution of second order partial differential equations in two dimensions using cubic trigonometric differential quadrature method

This chapter is an extension of the work discussed in the previous chapter in the context of dimensions. The research work in one dimension represented in the previous chapter is extended to two dimensions with cubic trigonometric differential quadrature method. The considered four partial differential equations which are introduced and solved in chapter 3 are presented for the numerical solution in two dimensions in this chapter.

4.1 Partial differential equations in two dimensions

Two-dimensional form of partial differential equations which arise during the mathematical modelling of the various phenomenon of science and engineering are as follows:

4.1.1 Sine-Gordon equation in two dimensions

Sine-Gordon equation is well-known equation having solitons solution in two dimensions, given as:

$$u_{tt} + \alpha u_t = \beta u_{xx} + \delta u_{yy} + \eta(x,y)sin(u),\ \ x,y \in [a,b] \times [c,d],\ \ t \geq 0 \tag{4.1}$$

with initial conditions

$$\begin{aligned} u(x,y,0) &= g_1(x,y) \\ u_t(x,y,0) &= g_2(x,y) \end{aligned} \tag{4.2}$$

and boundary condition

$$\begin{aligned} u(a,y,t) = l_1(y,t),\ \ u(b,y,t) = l_2(y,t) \\ u(x,c,t) = l_3(x,t),\ \ u(x,d,t) = l_4(x,t) \end{aligned} \tag{4.3}$$

where α, β, δ are real constants and $\eta(x,y)$ is known as Josephson current density; $g_1(x,y)$ is known as wave modes or kinks and $g_2(x,y)$ is velocity of the wave. In this equation, α is a dissipative term, which is assumed to be a real number with $\alpha \geq 0$. When $\alpha = 0$ this equation reduces to an undamped sine-Gordon equation in two space variables, while when $\alpha > 0$ the equation become damped. In order to solve the equation (4.1) the transformation $u_t = v$ is used to reduce the equation in the following system of differential equations:

$$\begin{aligned} u_t &= v, \\ v_t + \alpha\ v &= \beta\ u_{xx} + \delta\ u_{yy} + \eta(x,y)\ sin(u). \end{aligned} \tag{4.4}$$

4.1.2 Burgers' equation in two dimensions

The two dimensional form of Burgers' equation is

$$u_t = -uu_x - uu_y + \frac{1}{R}(u_{xx} + u_{yy}),\ \ x,y \in [a,b] \times [c,d],\ \ t \geq 0, \tag{4.5}$$

where R is Reynolds number.
Burgers' equation is also called as viscid Burgers' equation and get reduce to inviscid Burgers' equation when the kinematic thickness is not considered. The two

dimensional Burgers' equation is considered for the solution with initial conditions:

$$\begin{aligned} u(x,y,0) &= g_1(x,y) \\ u_t(x,y,0) &= g_2(x,y) \end{aligned} \tag{4.6}$$

and boundary conditions:

$$\begin{aligned} u(a,y,t) = l_1(y,t),\ u(b,y,t) = l_2(y,t) \\ u(x,c,t) = l_3(x,t),\ u(x,d,t) = l_4(x,t) \end{aligned} \tag{4.7}$$

4.1.3 Nonlinear Schrödinger equation in two dimensions

In two dimension, nonlinear Schrödinger equation can be considered in the following form:

$$i\ U_t = \alpha(U_{xx} + U_{yy}) + \beta|U|^2U + F(x,y,t)U,\ x,y \in [a,b]\times[c,d],\ t \geq 0 \tag{4.8}$$

with initial condition

$$U(x,y,0) = U_0(x,y) \tag{4.9}$$

and boundary condition:

$$\lim_{|x,y|\to\infty} U(x,y,t) = 0 \tag{4.10}$$

where α and β are arbitrary real values and $F(x,y,t)$ represents a bounded real valued function or trapping potential and $U = u+iv$ represents the complex valued wave function. The NLS equation results in two types of solutions [158]. When the solution along with spatial derivatives vanishes at $|x| = \infty$, it is called a bright soliton solution, and the solution which repeats itself after a specific domain L is called an L-periodic solution.

4.1.4 Linear telegraph equation in two dimensions

In two dimension, hyperbolic linear telegraph equation is as follows:

$$u_{tt}(x,y,t)+2\alpha u_t(x,y,t)+\beta^2u(x,y,t) = u_{xx}(x,y,t)+u_{yy}(x,y,t)+f(x,y,t),\ (4.11)$$

where $x, y \in [a, b] \times [c, d]$, and $t \geq 0$. Here α and β are real constants. Above described telegraph equation is solved with initial conditions:

$$\begin{aligned} u(x, y, 0) &= g_1(x, y) \\ u_t(x, y, 0) &= g_2(x, y) \end{aligned} \tag{4.12}$$

and boundary conditions:

$$\begin{aligned} u(a, y, t) = l_1(y, t),\ u(b, y, t) = l_2(y, t) \\ u(x, c, t) = l_3(x, t),\ u(x, d, t) = l_4(x, t) \end{aligned} \tag{4.13}$$

To solve the equation we use the transformation: $u_t = v$ and the equation (4.11) converted into a system of PDEs as follows:

$$\left.\begin{aligned} u_t(x, y, t) &= v(x, y, t), \\ v_t(x, y, t) &= u_{xx}(x, y, t) + u_{yy}(x, y, t) - 2\alpha v(x, y, t) - \beta^2 u(x, y, t) + f(x, y, t). \end{aligned}\right\} \tag{4.14}$$

4.1.5 Implementation of numerical scheme to partial differential equations

In order to solve these equations, substituting the first and second order approximation of the spatial derivatives in equations the following system is obtained:

1. Sine-Gordon equation in two dimensions:

$$u_t = v$$

$$v_t + \alpha\ v = \beta\ \sum_{j=1}^{N} b_{ij} u(x_j, y, t) + \delta\ \sum_{j=1}^{N} b'_{ij} u(x_j, y, t) + \eta(x, y)\ sin(u)$$

2. Burgers' equation in two dimensions:

$$u_t = -u\ \sum_{j=1}^{N} a_{ij} u(x_j, y, t) - u\ \sum_{j=1}^{N} a'_{ij} u(x_j, y, t) + \frac{1}{R}\left(\sum_{j=1}^{N} b_{ij} u(x_j, y, t) + \sum_{j=1}^{N} b'_{ij} u(x_j, y, t)\right)$$

3. Nonlinear Schrödinger equation in two dimensions:

$$U_t = \frac{1}{i}(\alpha(\sum_{j=1}^{N} b_{ij}U(x_j,y,t)) + \sum_{j=1}^{N} b'_{ij}U(x_j,y,t) + \beta|U|^2U + F(x,y,t))U$$

4. Linear telegraph equation in two dimensions:

$$u_t = v$$

$$v_t = -2\alpha v - \beta^2 u \sum_{j=1}^{N} b_{ij}u(x_j,y,t) + \sum_{j=1}^{N} b'_{ij}u(x_j,y,t) + f(x,y,t)$$

where a'_{ij} and b'_{ij} are the coefficient matrices as defined in section 1.4 of chapter 1, to calculate the approximations of first and second space derivatives in y direction. Now to solve the above system of ODE we use strong stability-preserving time-stepping RungeKutta (SSP-RK43) scheme [72].

4.2 Stability of the scheme

To check the stability of the scheme one can convert the considered equations to the system of ODEs and stability of the obtained system can be verified using the matrix method. On substituting the approximate values of derivatives in the equations and taking the nonlinear terms as constant, these equations can be written in simplified form as follows:

$$u_t = Bu + F(u) \tag{4.15}$$

where $F(u)$ represent the nonlinear terms in the equations and B takes different values as follows:

1. Sine-Gordon equation in two dimensions: $B = k\begin{bmatrix} 0 & I \\ \sum_{j=1}^{N} b_{ij} + \sum_{j=1}^{N} b'_{ij} & -\alpha \end{bmatrix}$
2. Burgers' equation in two dimensions: $B = k\frac{1}{R}(\sum_{j=1}^{N} b_{ij} + \sum_{j=1}^{N} b'_{ij})$
3. Nonlinear Schrödinger equation in two dimensions:$B = \frac{k\ \alpha\ (\sum_{j=1}^{N} b_{ij} + \sum_{j=1}^{N} b'_{ij})}{i}$

4. Linear telegraph equation in two dimensions: $B = k \begin{bmatrix} 0 & I \\ \sum_{j=1}^{N} b_{ij} + \sum_{j=1}^{N} b'_{ij} & -2\alpha \end{bmatrix}$

The stability of the system (4.15) depends on the eigenvalues of B. Since all eigenvalues of the matrix B presented in Figures 4.1, 4.2, 4.3, and 4.4 satisfy the conditions for stability, stated in the chapter 1. Hence it can be concluded that the scheme is stable.

4.3 Numerical solution and discussion

In this section, twelve test problems of considered equations are solved and examined for the numerical results. The accuracy and the efficiency of the method are measured by evaluating the L_2, L_∞ and RMS errors. Numerical results are compared and found better from the results present in the literature.

Example 4.1. *Consider the 2D sine-Gordon equation (4.1), in which superposition of two line solitons is obtained by taking* $\alpha = 1$, $\delta = 1$, $\beta = 1$, *and* $\eta(x, y) = -1$ *with initial conditions* $g_1(x, y) = 4tan^{-1}(exp(x)) + 4tan^{-1}(exp(y))$ *and* $g_2(x, y) = 0$ *for* $(x, y) \in [-6, 6] \times [-6, 6]$. *The numerical solutions of this problem are presented by means of graphs in 3D in Figure 4.5 for time 1, 3, 7 and 10 for* $\beta = 0.05$ *with* $\Delta t = 0.001$ *and* $N \times M = 31 \times 31$. *The graphs show the break up of two orthogonal line solitons which move away from each other in an undisturbed form. The presented graphs are in agreement with those given by Li et al. [205], Dehghan and Shokri [110], and Jiwari et al. [111]. Error norms are shown for different values of* $h_x = h_y$ *and presented in Figures 4.6 and 4.7. In Table 4.1* L_∞*-error is presented at different time levels and the* L_2, L_∞, *and* RMS *errors at time t=1 are given in Table 4.2.*

Example 4.2. *The 2D sine-Gordon equation (4.1) is solved having numerical solutions for a line soliton in an inhomogeneous medium, obtained for the Josephson current density* $\eta(x, y) = -(1 + sech^2(\sqrt{x^2 + y^2}))$, $\alpha = 1$, $\delta = 1$, *and* $\beta = 1$ *with initial conditions* $g_1(x, y) = 4tan^{-1}(exp(\frac{x-3.5}{0.954}))$, *and* $g_2(x, y) = 0.629sech(exp(\frac{x-3.5}{0.954}))$ $(x, y) \in [-7, 7] \times [-7, 7]$. *The numerical solutions of this problem are presented as*

graphs at time 1, 3, 6 and 9 for $\beta = 0.05$ *with* $\Delta t = 0.001$ *and* $N \times M = 31 \times 31$ *in Figure 4.8. Figure shows that in all the obtained graphs, the line soliton is moving almost as a straight line during transmission through inhomogeneity. The presented graphs are in agreement with those given by Dehghan and Shokri [110] and Jiwari et al. [111]. Error norms are shown for different values of* $h_x = h_y$ *and presented in Figures 4.9 and 4.10. Tables 4.3 shows* L_∞ *error at different time levels* $t = 1, 1.5, 2$. *The obtained* L_2, L_∞, *and RMS errors at time t=1 are presented in Table 4.4.*

Example 4.3. *This example is considered to solve the 2D sine-Gordon equation (4.1), in which circular ring solitons are found for the case when* $\alpha = 1$, $\delta = 1$, $\beta = 1$, *and* $\eta(x, y) = -1$. *The equation is solved with initial conditions* $g_1(x, y) = 4tan^{-1}(exp(3 - \sqrt{x^2 + y^2}))$ *and* $g_2(x, y) = 0$ *in domain* $(x, y) \in [-7, 7] \times [-7, 7]$. *The numerical solutions are presented as graphs at different time, t= 1, 5.6, 8.4 and 11.2 for* $\beta = 0.05$ *with* $\Delta t = 0.001$ *and* $N \times M = 31 \times 31$ *in Figure 4.11. It is observed that the ring soliton shrinks for initial stage but as time goes on, oscillations and radiations begin to form and continue to form up to* $t = 8.4$ *while at* $t = 11.2$ *the graph shows a nearly formed ring soliton again. The graphs are in agreement with those given by Dehghan and Shokri [110] and Jiwari et al. [111]. Error norms are shown for different values of* $h_x = h_y$ *and presented in Figures 4.12 and 4.13. In Table 4.5 the obtained* L_∞*-errors are presented at different time levels. The* L_2, L_∞, *and RMS errors are shown in Table 4.6 at time t=1.*

Example 4.4. *Consider two dimension Burgers' equation of form:*

$$u_t = -uu_x - uu_y + \frac{1}{R}(u_{xx} + u_{yy})$$

with space domain $(x, y) \in [0, 1] \times [0, 1]$. *The initial and boundary conditions for this test problem are obtained from the exact solution [206]:*

$$u(x, y, t) = \frac{1}{1 + exp(\frac{R(x+y-t)}{2})}$$

The L_2 *and* L_∞ *error for numerical solution are obtained and enlisted in Table 4.7 at different time levels* $t = 0.05, 0.25, 0.5$ *for* $R = 1$ *and* 10 *with* $N = M = 60$ *for* $\Delta t = 0.0001$. *The physical behavior of the numerical solution is obtained for* $t = 0.2, 0.61, 1.4$ *and presented in the Figure 4.14. In Figure 4.15 comparison of the numerical and the exact solution is given in form of surface and contour plots*

with $N = M = 30$, $\Delta t = 0.0001$, *and* $R = 50$.

Example 4.5. *In this example, consider the two dimension Burgers' equation:*

$$u_t = -uu_x - uu_y + \frac{1}{R}(u_{xx} + u_{yy})$$

with domain $(x, y) \in [-0.5, 0.5] \times [-0.5, 0.5]$. *The initial and boundary conditions for this test problem are obtained from the exact solution [206]:*

$$u(x, y, t) = 0.5 - tanh(\frac{R(x + y - t)}{2})$$

The L_2 *and* L_∞ *error for numerical solution are obtained and enlisted in the Table 4.8 at different time levels* $t = 0.05, 0.25, 0.5$ *for* $R = 1$ *and* 10 *with* $N = M = 60$ *for* $\Delta t = 0.0001$. *The physical behavior of the numerical solution is obtained for* $t = 0.25, 0.5, 0.75, 1$ *and presented in the Figure 4.16. In Figure 4.17 the comparison of the numerical and the exact solution is given in form of surface and contour plots at* $N = M = 30$, $\Delta t = 0.0001$, *and* $R = 50$.

Example 4.6. *Consider the 2D NLS equation (4.8) with* $\alpha = 1$ *and* $\beta = 0$

$$i\, U_t(x, y, t) = U_{xx}(x, y, t) + U_{yy}(x, y, t) + F(x, y)U(x, y, t)$$

where $(x, y) \in [0, 1] \times [0, 1]$, $t > 0$ *and* $F(x, y) = 3 - 2tanh^2(x) - 2tanh^2(y)$. *The initial and boundary conditions can be obtained from the exact solution [177]:*

$$U(x, y, t) = \frac{iexp(it)}{cosh(x)cosh(y)}$$

In numerical computations maximum or L_∞ *error and average errors are calculated using both of the methods as discussed in chapter 3, at time* $t \leq 1$ *with* $N = 30$, $\Delta t = 0.0001$ *and enlisted in Table 4.9, with comparison of the errors given by Dehghan [163] and Mohebbi [177]. Comparison of numerical and the exact solutions is presented in Figure 4.18.*

Example 4.7. *By taking* $\alpha = 1$ *and* $\beta = 0$, *2D form of the NLS equation (4.8) become:*

$$i\, U_t(x, y, t) = U_{xx}(x, y, t) + U_{yy}(x, y, t) + F(x, y)U(x, y, t)$$

where $x, y \in [0,1]$, $t > 0$ *and* $F(x,y) = 1 - \frac{2}{x^2} - \frac{2}{y^2}$. *The equation is solved with initial and boundary conditions obtained from exact solution [177]:*

$$U(x,y,t) = x^2 y^2 exp(it).$$

To check the accuracy of the proposed methods L_∞ *and average errors are calculated using both of the methods as discussed in chapter 3, at time* $t \leq 1$ *with* $N \times M = 30 \times 30$, $\Delta t = 0.0001$ *and shown in Table 4.10, comparison of the error is given with Dehghan [163] and Abdur[178]. Comparison of numerical and the exact solutions is presented in Figure 4.19.*

Example 4.8. *Consider the 2D NLS equation (4.8) with* $\alpha = -\frac{1}{2}$ *and* $\beta = 1$

$$i\, U_t(x,y,t) = -\frac{1}{2}(U_{xx}(x,y,t) + U_{yy}(x,y,t)) + F(x,y)U(x,y,t) + |U(x,y,t)|^2 U(x,y,t)$$

where $x, y \in [0, 2\pi]$, $t > 0$ *and* $F(x,y) = 1 - sin^2(x)sin^2(y)$. *Initial and boundary conditions can be obtained from the exact solution [176]:*

$$U(x,y,t) = sin(x)sin(y)exp(-i2t)$$

In numerical computations L_∞ *and average errors are calculated using both of the methods as discussed in chapter 3, at time* $t \leq 16$ *with* $N = M = 40$ *and* $\Delta t = 0.0001$ *and shown in Table 4.11, comparison of the error is given with Wang [176]. Comparison of numerical and the exact solutions is presented in Figure 4.20.*

Example 4.9. *Consider the telegraph equation (4.11) with* $(x,y) \in [0,1] \times [0,1]$, $\alpha = 1$, $\beta = 1$ *and* $f(x,y,t) = 2(cos(t) - sin(t))sin(x)sin(y)$. *The initial and boundary conditions can be obtained from the exact solution:*

$$u(x,y,t) = cos(t)sin(x)sin(y).$$

L_2 *and* L_∞ *errors are calculated at different time levels with* $\Delta t = 0.001$, $N_x = N = N_y = M = 21$ *and enlisted in Table 4.12 and results are found to be empowering. Obtained results are also shown in form of surface and contour plots at different times in Figure 4.21.*

Example 4.10. *Consider the telegraph equation (4.11) with* $(x,y) \in [0,1] \times [0,1]$, $\alpha = 10$, $\beta = 5$ *and* $f(x,y,t) = (-2\alpha + \beta^2 - 1)exp(-t)sinh(x)sinh(y)$. *The initial*

and boundary conditions can be obtained from the exact solution:

$$u(x, y, t) = exp(-t)sinh(x)sinh(y).$$

L_2 and L_∞ errors are calculated at different time levels with $\Delta t = 0.001$, $N_x = N = N_y = M = 21$ and enlisted in Table 4.13. Obtained results are also shown in form of surface and contour plots at different times in Figure 4.22.

Example 4.11. *Consider the telegraph equation (4.11) with $(x, y) \in [0, 1] \times [0, 1]$, $\alpha = 1$, $\beta = 1$ and $f(x, y, t) = -2exp(x+y-t)$. The initial and boundary conditions can be obtained from the exact solution:*

$$u(x, y, t) = exp(x + y - t).$$

L_2 and L_∞ errors are calculated at different time levels with $\Delta t = 0.001$, $N_x = N = N_y = M = 21$ and enlisted in Table 4.14. Obtained results are also shown in form of surface and contour plots at different times in Figure 4.23.

Example 4.12. *Consider the telegraph equation (4.11) with $(x, y) \in [0, 1] \times [0, 1]$, $\alpha = 1$, $\beta = 1$ and $f(x, y, t) = 2\pi^2 exp(-t)sin(\pi x)sin(\pi y)$. The initial and boundary conditions can be obtained from the exact solution:*

$$u(x, y, t) = exp(-t)sin(\pi x)sin(\pi y).$$

L_2 and L_∞ errors are calculated at different time levels with $\Delta t = 0.001$, $N_x = N = N_y = M = 21$ and enlisted in Table 4.15. Obtained results are also shown in form of surface and contour plots at different times in Figure 4.24.

4.4 Summary

In this chapter, the numerical solutions of second order partial differential equations in two dimensions are computed using developed hybrid scheme in which differential quadrature method is implemented using modified cubic trigonometric B-spline basis function. In this chapter four well-known equations namely: sine-Gordon, Burgers', Nonlinear Schrödinger, and linear telegraph equations in two dimensions are considered for the numerical solution. The previous section numerical results and discussion, demonstrate the good accuracy of the proposed method.

Various numerical examples are taken for each considered equation. While implementing the proposed scheme, these equations are converted to a system of ordinary differential equations which is further solved using the SSP-RK43 method. Since errors are the best way to justify the working and ability of a method, the error norms are calculated and enlisted in the form of a table to show the robustness of the method. Stability of the scheme is also evaluated with the help of eigenvalues and the scheme is found to be unconditionally stable. As a summary, this method proved to be a significant tool in solving the linear and nonlinear two dimensional partial differential equations of second order.

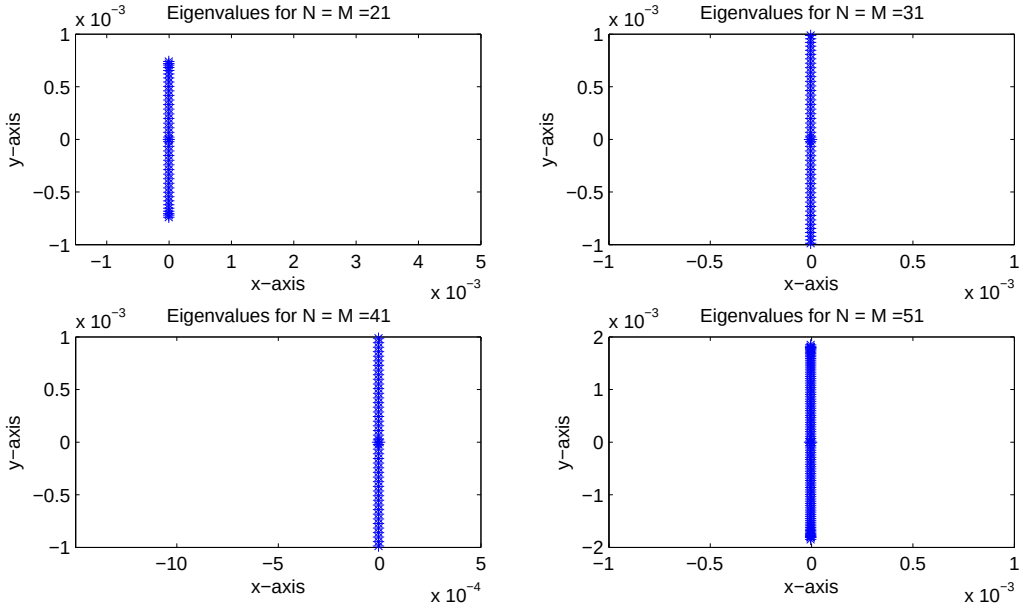

FIGURE 4.1: Eigenvalues of the matrix B for sine-Gordon equation with different partitions of the domain in two dimensions

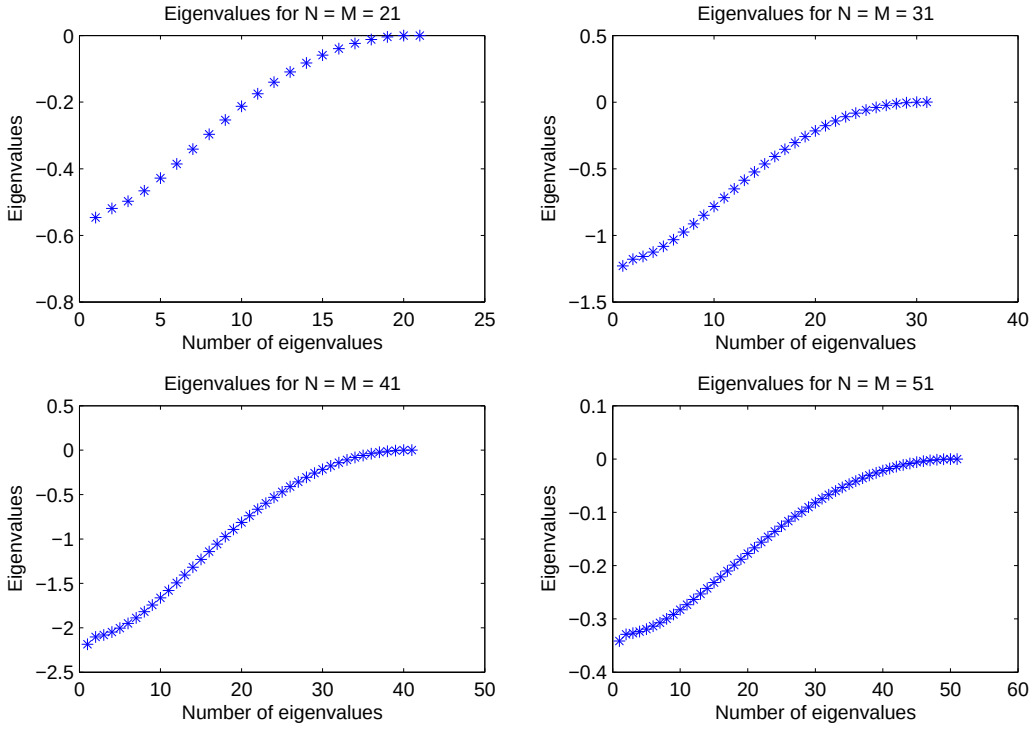

FIGURE 4.2: Eigenvalues of the matrix B for Burgers' equation with different partitions of the domain in two dimensions

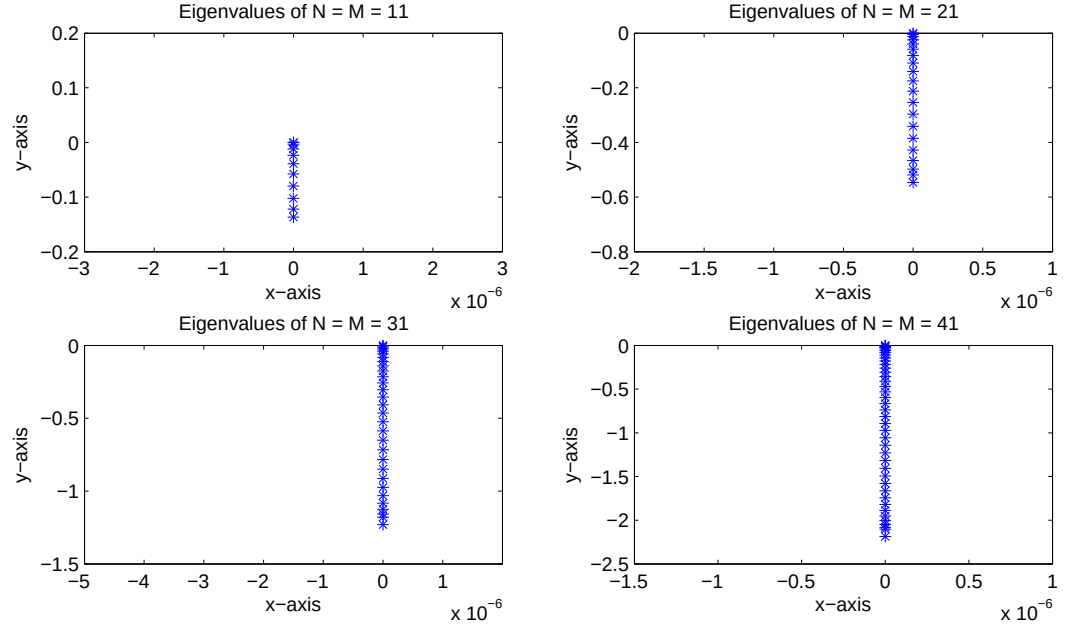

FIGURE 4.3: Eigenvalues of the matrix B for Schrödinger equation with different partitions of the domain in two dimensions

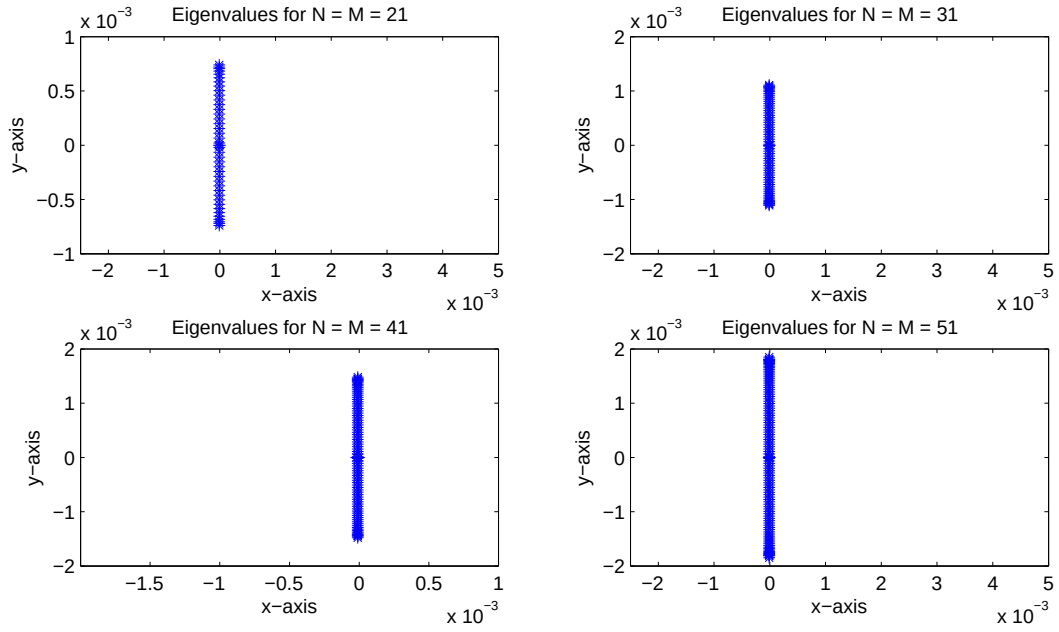

FIGURE 4.4: Eigenvalues of the matrix B for Telegraph equation with different partitions of the domain in two dimensions

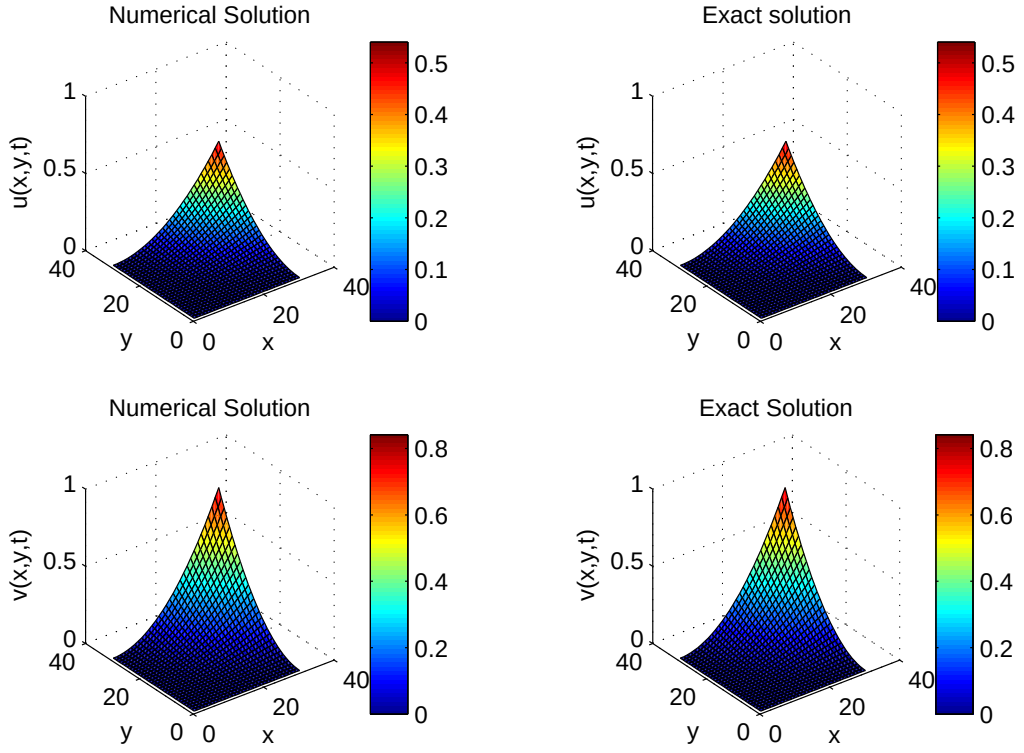

FIGURE 4.19: The surface plot of real and imaginary parts of the solution for example 4.7 at time $t = 1$ and $N = M = 30$

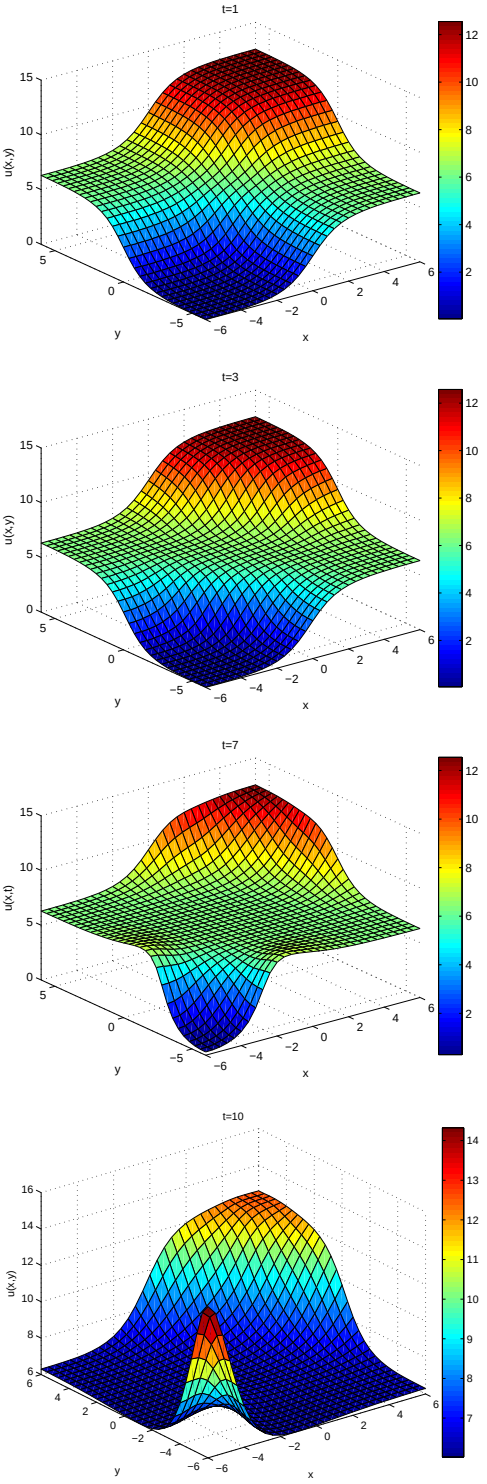

FIGURE 4.5: The surface plot of example 4.1 at time $t = 1, 3, 7, 10$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 31$

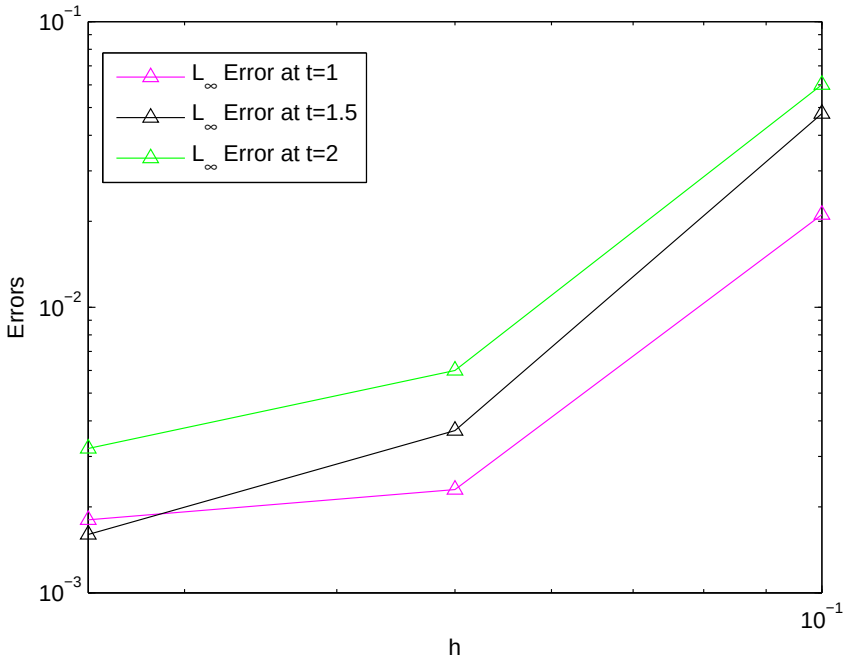

FIGURE 4.6: L_∞ of example 4.1 at different time $t = 1, 1.5, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 40, 20, 10$

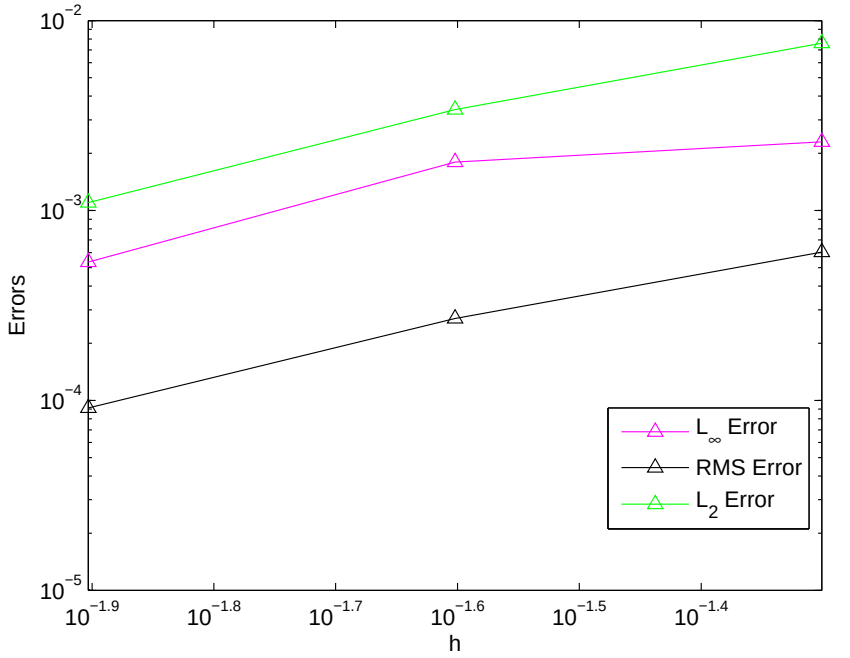

FIGURE 4.7: Errors and convergence with respect to the nodal spacing h with $N_x = N = N_y = M = 80, 40, 20$ for Example 4.1 at time $t = 1$ and $\Delta t = 0.001$

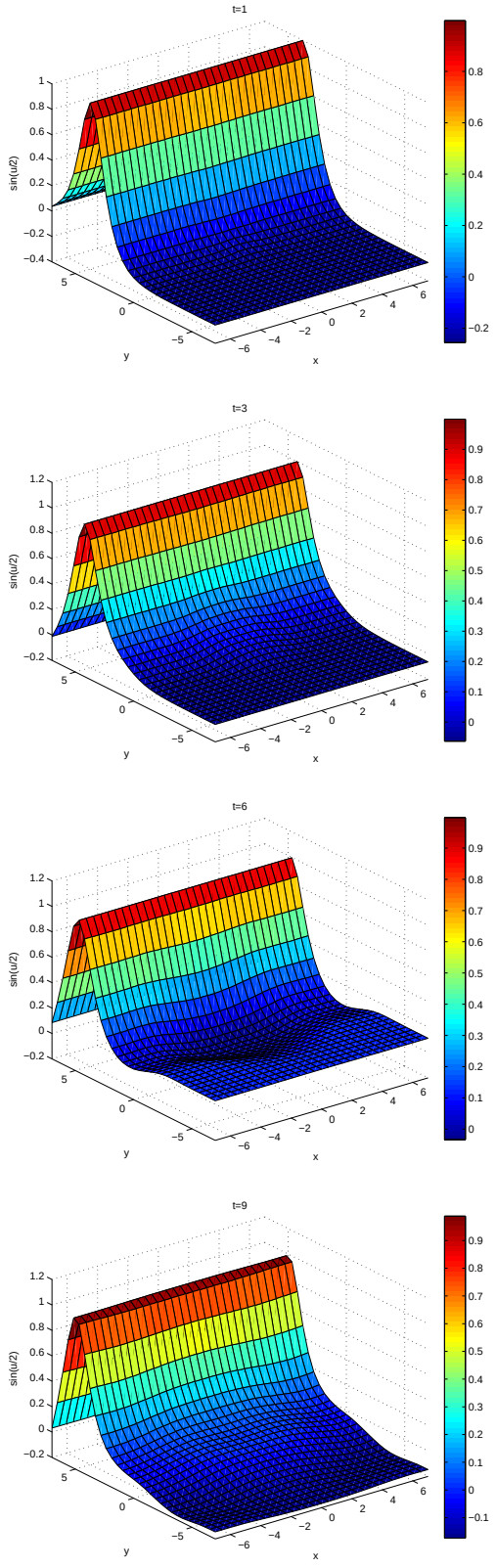

FIGURE 4.8: The surface plot of example 4.2 at time $t = 1, 3, 6, 9$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 31$

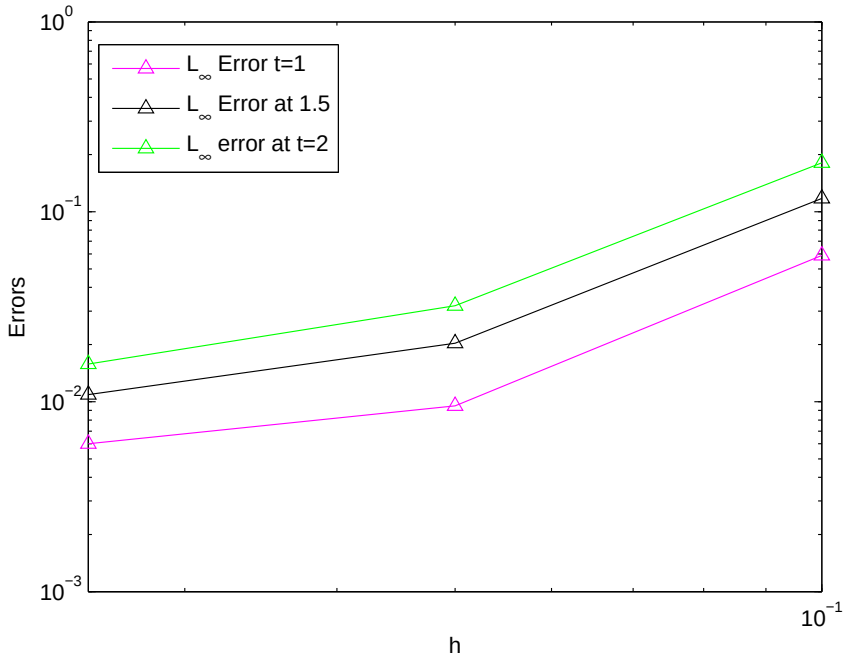

FIGURE 4.9: L_∞ of example 4.2 at different time $t = 1, 1.5, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 40, 20, 10$

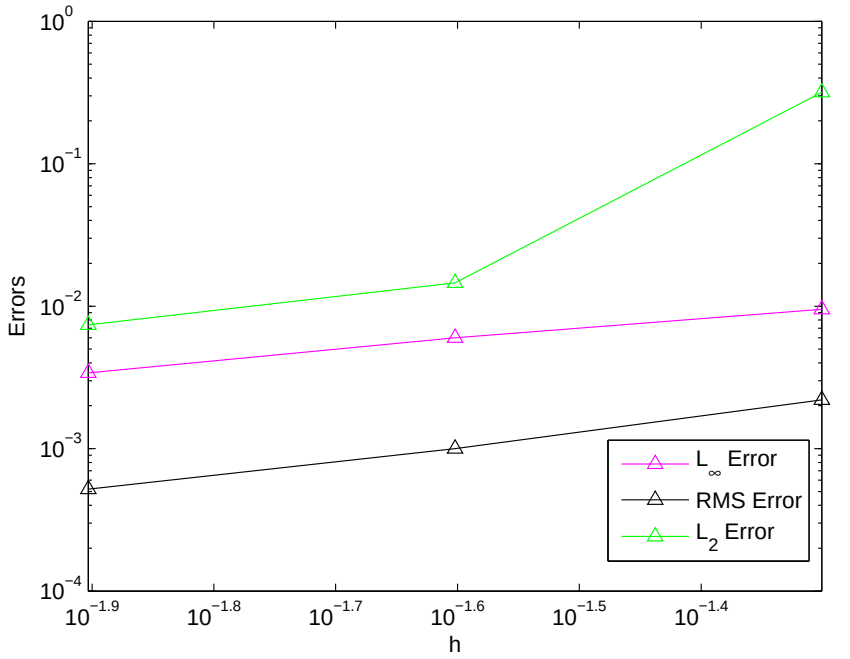

FIGURE 4.10: Errors and convergence with respect to the nodal spacing h with $N_x = N = N_y = M = 80, 40, 20$ for example 4.2 at time $t = 1$ and $\Delta t = 0.001$

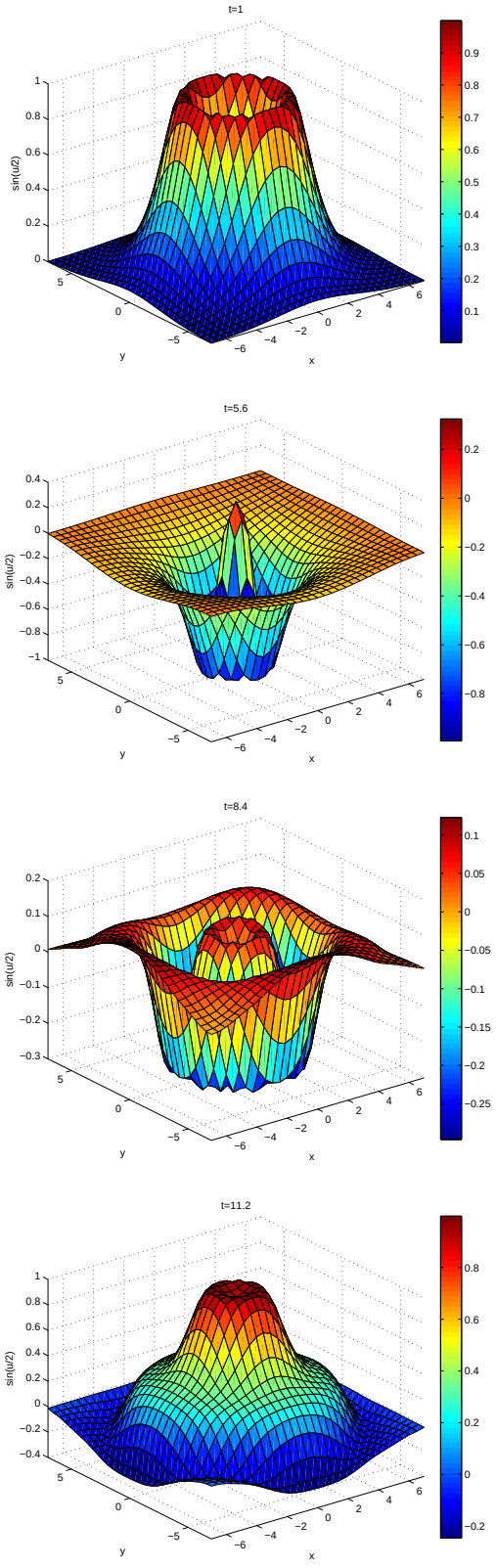

FIGURE 4.11: The surface plot of example 4.3 at time $t = 1, 5.6, 8.4, 11.2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 31$

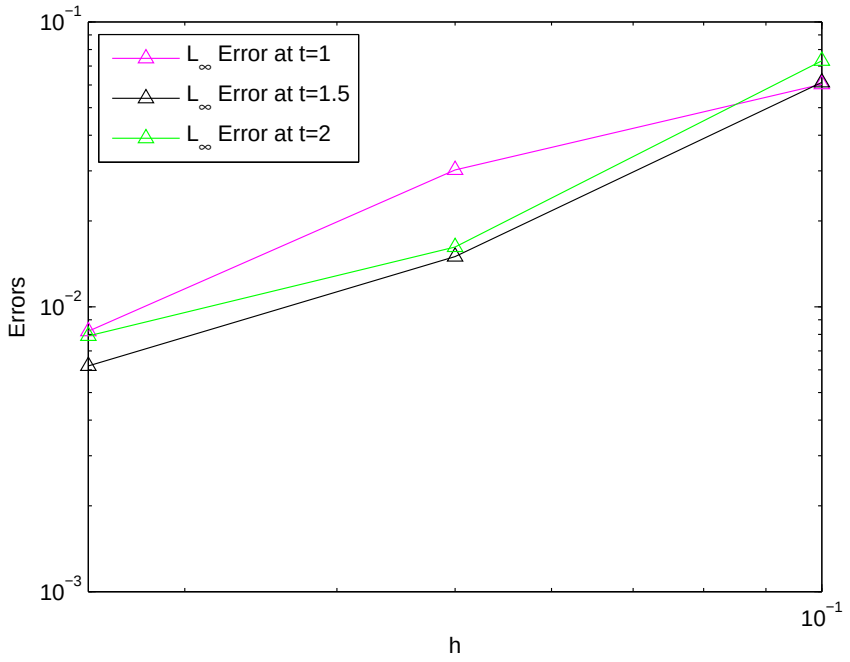

FIGURE 4.12: L_∞ of example 4.3 at different time $t = 1, 1.5, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 40, 20, 10$

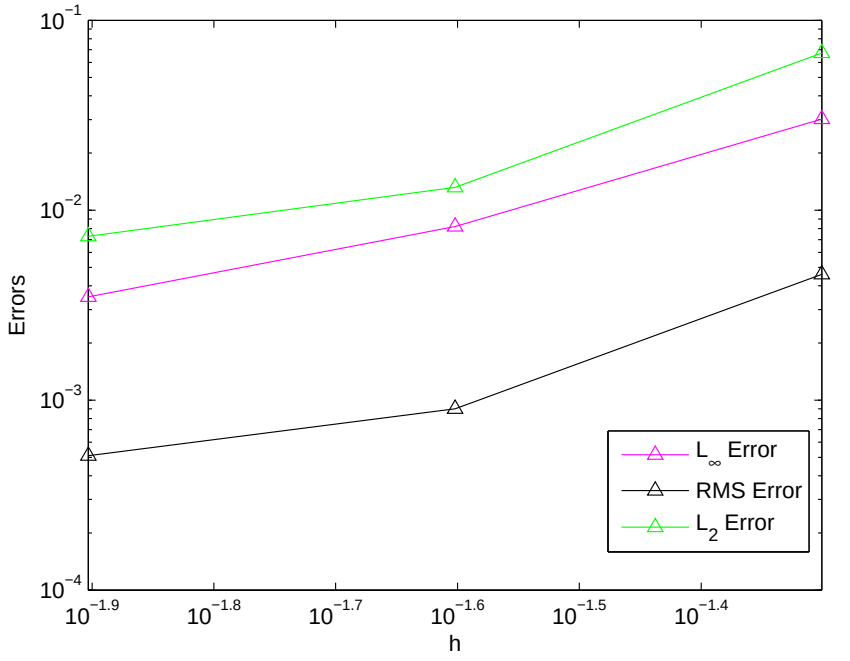

FIGURE 4.13: Errors and convergence with respect to the nodal spacing h with $N_x = N = N_y = M = 80, 40, 20$ for example 4.3 at time $t = 1$ and $\Delta t = 0.001$

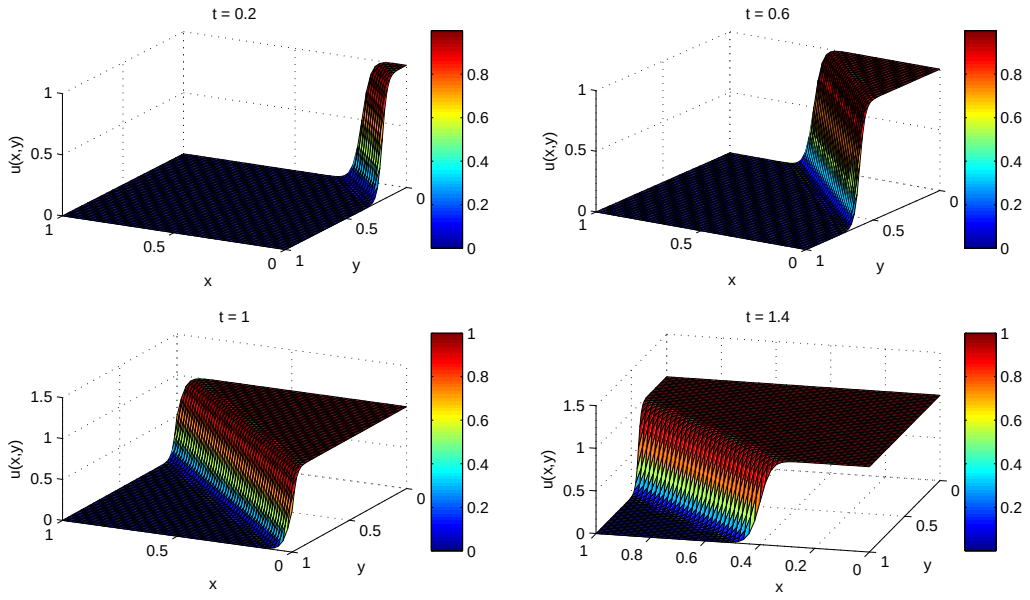

FIGURE 4.14: The physical behaviour of numerical solution in example 4.4 at time $t = 0.2, 0.6, 1, 1.4$ for $N = M = 50$ and $k = \Delta t = 0.01$

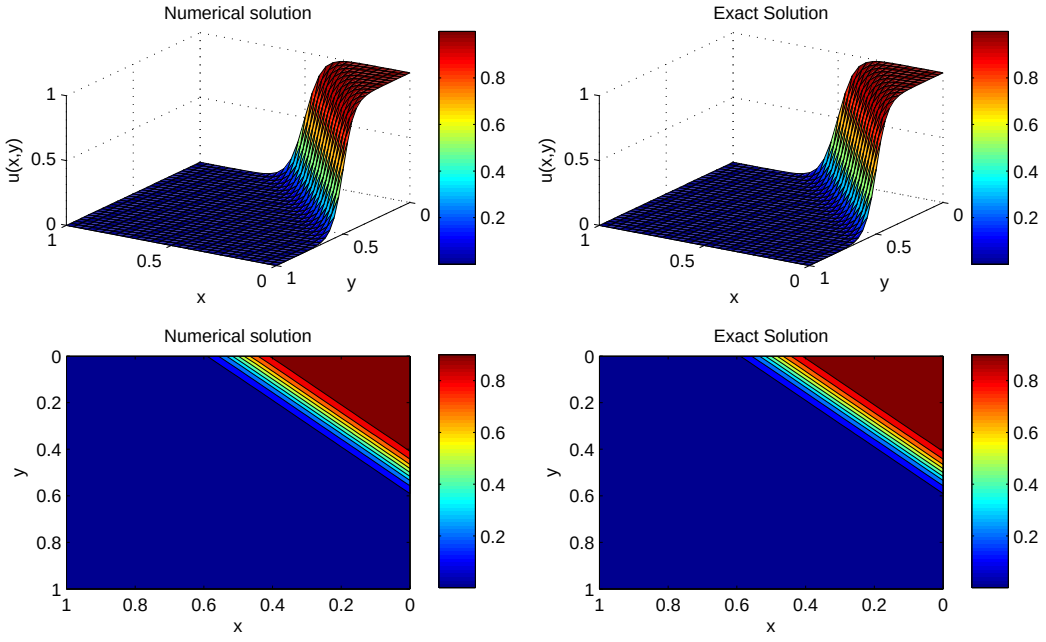

FIGURE 4.15: The comparison of numerical and the exact solution for example 4.4 at time $t = 0.5$ for $N = M = 30$, $k = \Delta t = 0.0001$ and R=50

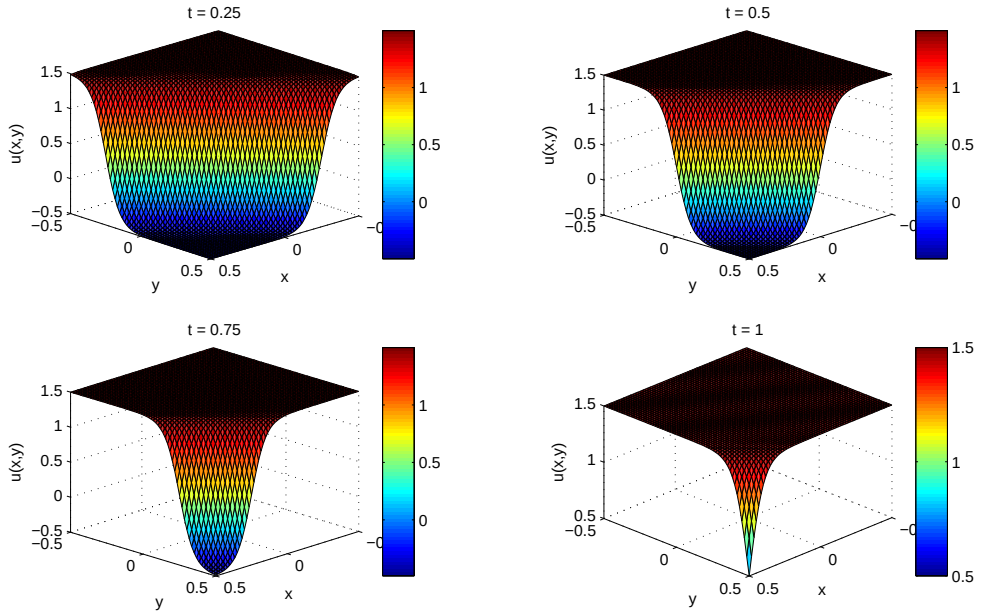

FIGURE 4.16: The physical behaviour of numerical solution in example 4.5 at time $t = 0.25, 0.5, 0.75, 1$ for $N = M = 60$ and $k = \Delta t = 0.0001$

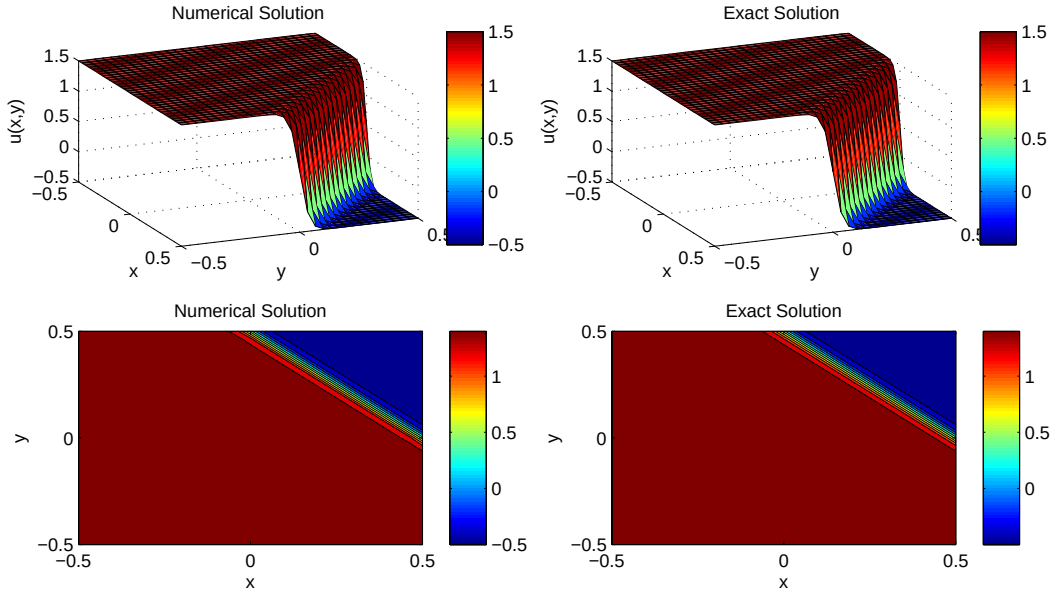

FIGURE 4.17: The comparison of numerical and the exact solution for example 4.5 at time $t = 0.5$ for $N = M = 30$, $k = \Delta t = 0.0001$ and $R = 50$

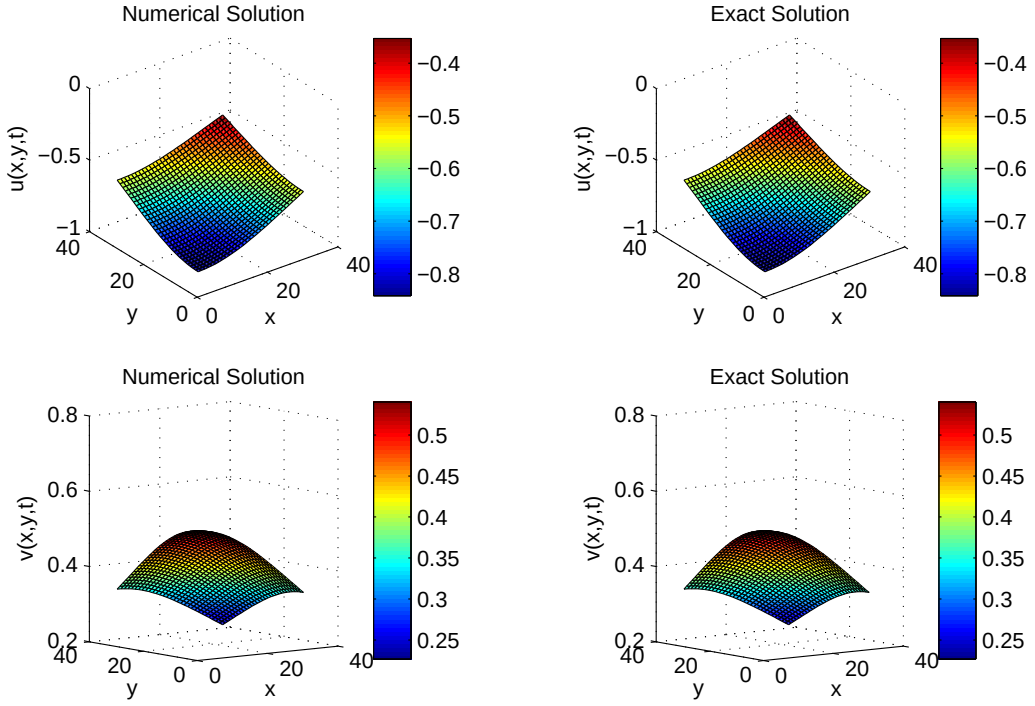

FIGURE 4.18: The surface plot of real and imaginary parts of the solution for example 4.6 at time $t = 1$ and $N = M = 30$

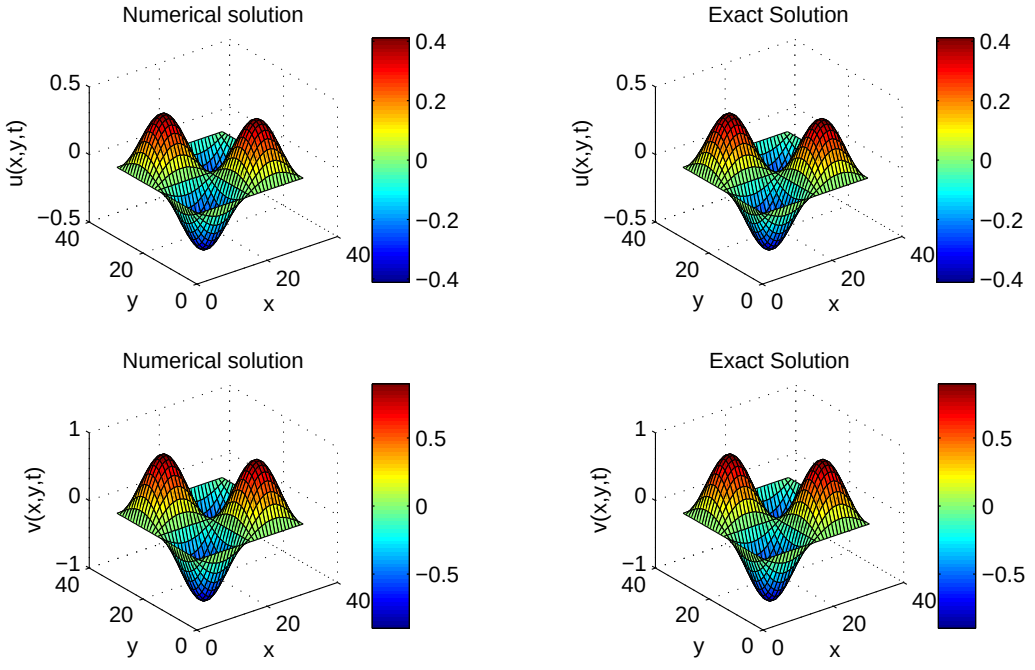

FIGURE 4.20: The surface plot of real and imaginary parts of the solution for example 4.8 at time $t = 1$ and $N = M = 30$

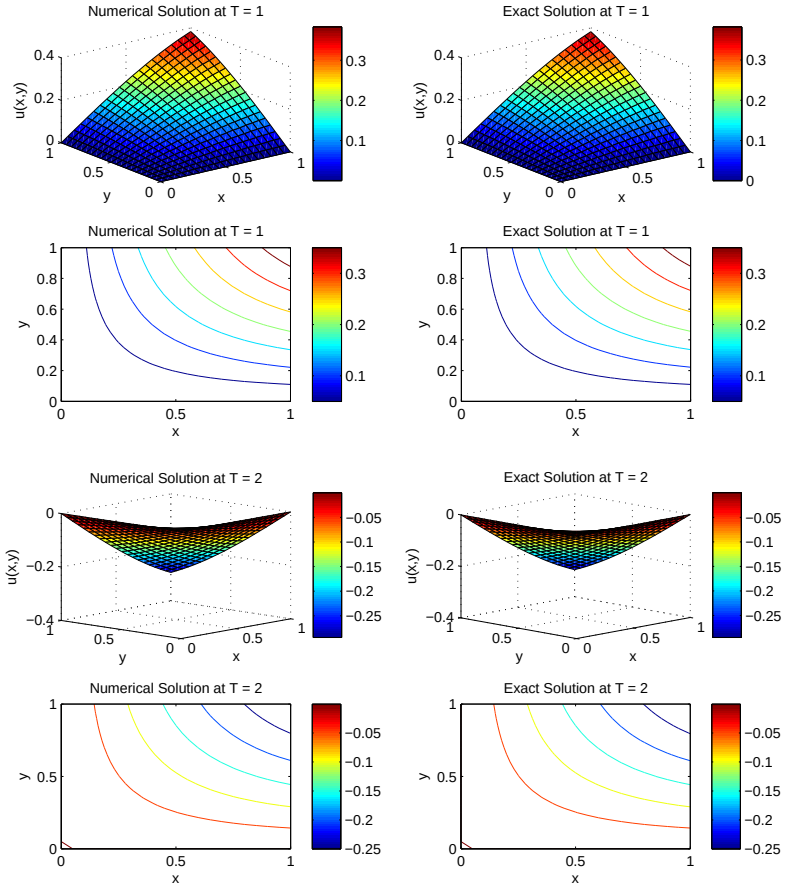

FIGURE 4.21: The surface and contour plot of example 4.9 at time $t = 1, 2$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$

TABLE 4.1: L_∞ error in the MCTB-DQM solution of example 4.1 for $\Delta t = 0.001$ for different values of $N_x = N = N_y = M = 10, 20, 40$ at different time levels

N	t=1	t=1.5	t=2
10	0.0211	0.0476	0.0601
20	2.30E-3	3.70E-3	6.00E-3
40	1.80E-3	1.60E-3	3.20E-3

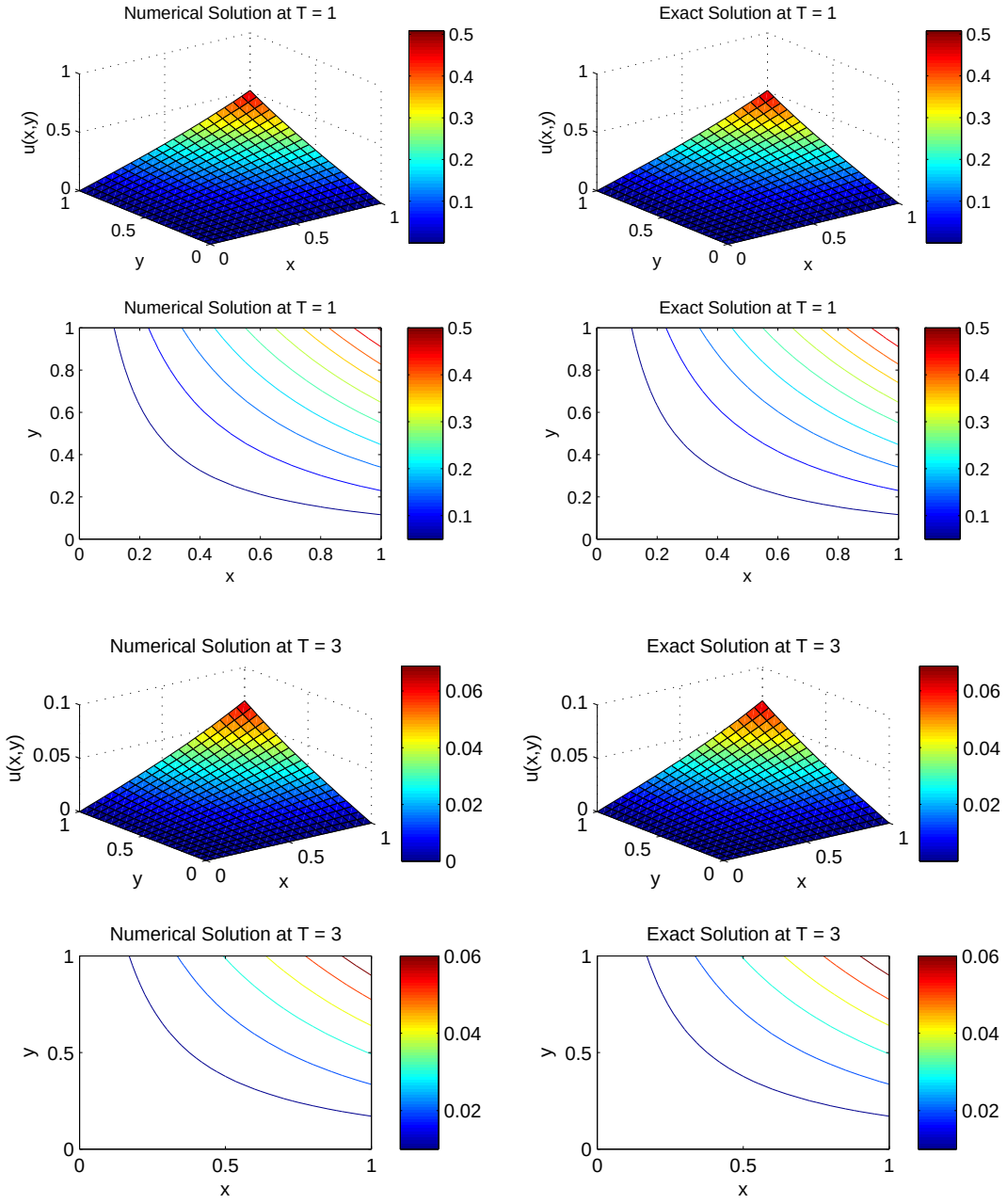

FIGURE 4.22: The surface and contour plot of example 4.10 at time $t = 1, 3$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$

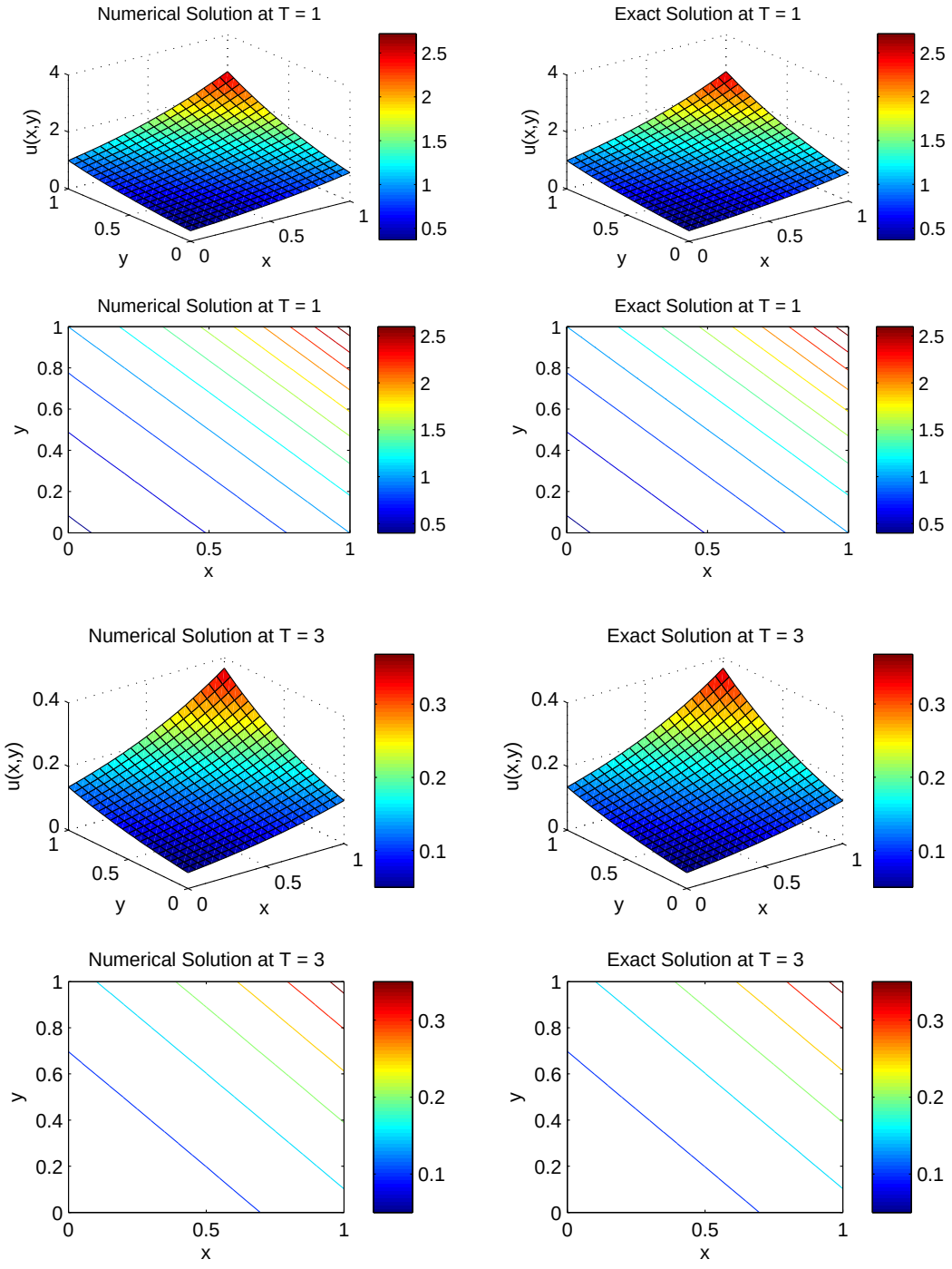

FIGURE 4.23: The surface and contour plot of example 4.11 at time $t = 1, 3$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$

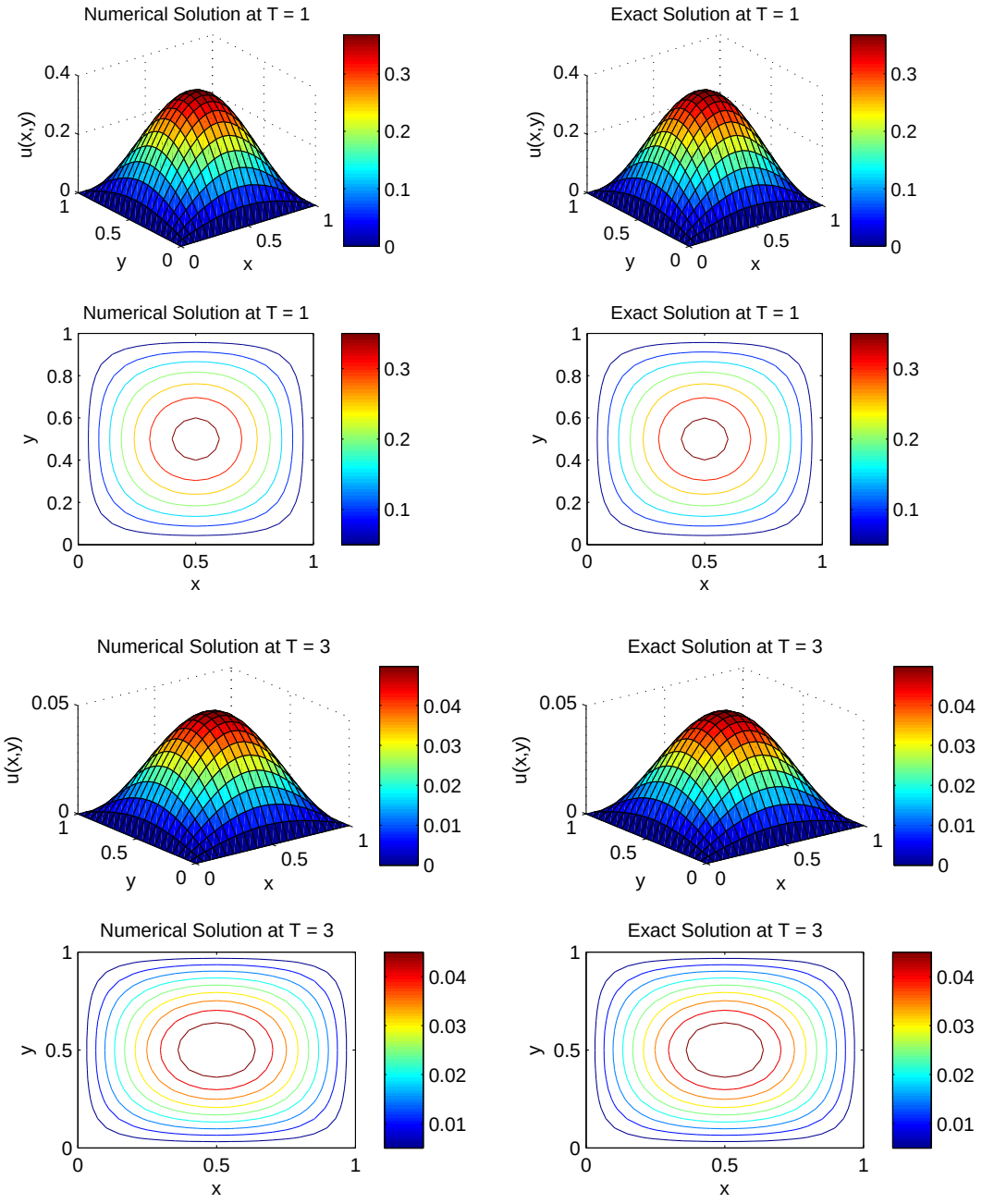

FIGURE 4.24: The surface and contour plot of example 4.12 at time $t = 1, 3$ with $\Delta t = 0.001$ and $N_x = N = N_y = M = 21$

TABLE 4.2: L_∞, RMS, and L_2, errors in the MCTB-DQM solution of example 4.1 for $\Delta t = 0.001$ and for different values of $N_x = N = N_y = M = 20, 40, 80$ at t=1

N	L_∞	RMS	L_2
20	2.30E-3	6.03E-4	7.60E-3
40	1.80E-3	2.70E-4	3.40E-3
80	5.35E-4	9.12E-5	1.10E-3

TABLE 4.3: L_∞ error in the MCTB-DQM solution of example 4.2 for $\Delta t = 0.001$ for different values of $N_x = N = N_y = M = 10, 20, 40$ at t=1

N	t=1	t=1.5	t=2
10	0.0589	0.1176	0.1810
20	9.50E-3	0.0203	0.0320
40	6.00E-3	0.0109	0.0158

TABLE 4.4: L_∞, RMS, and L_2, errors in the MCTB-DQM solution of example 4.2 for $\Delta t = 0.001$ and for different values of $N_x = N = N_y = M = 20, 40, 80$ at t=1

N	L_∞	RMS	L_2
20	0.0095	0.0022	0.3170
40	6.00E-3	1.00E-3	0.0146
80	3.40E-3	5.20E-4	7.40E-3

TABLE 4.5: L_∞ error in the MCTB-DQM solution of example 4.3 for $\Delta t = 0.001$ for different values of $N_x = N = N_y = M = 10, 20, 40$ at different time levels

N	t=1	t=1.5	t=2
10	0.0604	0.0613	0.0730
20	0.0302	0.0150	0.0162
40	8.20E-3	6.20E-3	7.90E-3

TABLE 4.6: L_∞, RMS, and L_2, errors in the MCTB-DQM solution of example 4.3 for $\Delta t = 0.001$ and for different values of $N_x = N = N_y = M = 20, 40, 80$ at t=1

N	L_∞	RMS	L_2
20	0.0302	4.60E-3	0.0672
40	8.20E-3	9.00E-4	0.0132
80	3.50E-3	5.10E-4	7.30E-3

TABLE 4.7: Errors norms in the MTCB-DQM solution of example 4.4 at different time levels with $N = M = 60$, and $\Delta t = 0.0001$

	R = 1, Time = 0.05		R = 1, Time = 0.25		R = 1, Time = 0.5	
Method	L_∞	L_2	L_∞	L_2	L_∞	L_2
MTCB-DQM	8.67E-6	5.62E-6	4.99E-6	3.88E-6	5.02E-6	3.89E-6
	R = 10, Time = 0.05		R = 10, Time = 0.25		R = 10, Time = 0.5	
Method	L_∞	L_2	L_∞	L_2	L_∞	L_2
MTCB-DQM	6.92E-5	1.16E-5	5.30E-5	1.51E-5	5.26E-5	1.88E-5

TABLE 4.8: Errors norms in the MTCB-DQM solution of example 4.5 at different time levels with $N = M = 60$, and $\Delta t = 0.0001$

	R = 1, Time = 0.05		R = 1, Time = 0.25		R = 1, Time = 0.5	
Method	L_∞	L_2	L_∞	L_2	L_∞	L_2
MTCB-DQM	3.67E-5	2.29E-5	1.93E-5	1.46E-5	2.00E-5	1.34E-5
	R = 10, Time = 0.05		R = 10, Time = 0.25		R = 10, Time = 0.5	
Method	L_∞	L_2	L_∞	L_2	L_∞	L_2
MTCB-DQM	4.99E-4	1.43E-4	4.91E-4	1.12E-4	4.66E-4	7.14E-5

TABLE 4.9: Comparison of errors obtained by applying both of the methods in example 4.6 for $N = M = 30$, $\Delta t = 0.0001$ at different time levels

	Method 1		Method 2		Mohebbi[177]		Dehghan[163]	
t	L_∞ for u	L_∞ for v	Max. Err.	Average Err.	L_∞ for u	L_∞ for v	L_∞ for u	L_∞ for v
0.1	6.64E-5	1.14E-4	1.32E-4	3.59E-5	4.61E-7	5.08E-7	2.44E-5	2.99E-5
0.3	1.47E-4	1.04E-4	1.55E-4	6.15E-5	4.89E-7	4.25E-7	2.94E-5	2.94E-5
0.5	9.29E-5	1.09E-4	1.19E-4	4.08E-5	4.33E-7	5.30E-7	2.74E-5	3.40E-5
0.7	1.04E-4	7.19E-5	1.10E-4	5.72E-5	5.00E-7	3.03E-7	2.54E-5	1.86E-5
1	1.02E-4	1.52E-4	1.68E-4	5.65E-5	4.08E-7	4.76E-7	2.94E-5	2.42E-5

TABLE 4.10: Comparison of errors obtained by applying both of the methods in example 4.7 for $h = 0.033$, $\Delta t = 0.0001$ at different time levels

t	Method 1		Method 2		Abdur[178]		Dehghan[163]	
	L_∞ for u	L_∞ for v	Max. Error	Average Error	Max. Error in u	Max. Error in v	Max. Error in u	Max. Error in v
0.1	3.79E-5	5.11E-5	5.55E-5	1.36E-5	5.15E-5	4.68E-5	4.04E-4	3.57E-4
0.3	4.14E-5	3.87E-5	5.17E-5	1.08E-5	6.23E-5	4.16E-5	5.12E-4	8.05E-4
0.5	4.64E-5	5.61E-5	6.78E-5	1.58E-5	5.74E-5	4.06E-5	4.63E-4	3.95E-4
0.7	3.87E-5	4.73E-5	4.81E-5	1.32E-5	4.90E-5	5.27E-5	3.89E-4	4.16E-4
1	4.53E-5	5.01E-5	5.01E-5	1.45E-5	4.83E-5	5.23E-5	3.72E-4	4.12E-4

TABLE 4.11: Comparison of errors obtained by applying both of the methods in example 4.8 for $N = 40$, $\Delta t = 0.0001$ at different time levels

	Method 1		Method 2		Wang [176]
t	L_∞ for u	L_∞ for v	Max. Error	Average Err.	Max. Error
4	3.22E-4	1.06E-4	3.34E-4	9.40E-5	8.11E-4
8	2.40E-4	6.74E-4	6.89E-4	2.61E-4	1.62E-3
12	9.65E-4	4.68E-4	1.04E-3	4.34E-4	2.43E-3
16	8.22E-4	1.29E-3	1.53E-3	6.010E-4	3.24E-3

TABLE 4.12: Comparison of error norms obtained by present method in example 4.9 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels

	Present Method		R. C. Mittal [207]	
t	L_2	L_∞	L_2	L_∞
1	1.76E-4	5.95E-4	9.88E-5	2.49E-4
2	2.02E-4	6.43E-4	1.21E-4	3.22E-4
3	4.96E-5	1.15E-4	3.76E-5	9.93E-5
5	2.12E-4	6.79E-4	1.27E-4	3.32E-4
7	1.29E-4	4.64E-4	6.76E-5	1.76E-4
10	1.03E-4	3.84E-4	5.17E-5	1.35E-4

TABLE 4.13: Comparison of error norms obtained by present method in example 4.10 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels

	Present Method		R. C. Mittal [207]	
t	L_2	L_∞	L_2	L_∞
0.5	1.96E-4	3.80E-4	1.06E-4	2.47E-4
1	1.27E-4	5.08E-4	1.52E-5	3.30E-4
2	4.86E-5	1.87E-4	4.64E-5	1.13E-5
3	1.80E-5	6.88E-5	2.19E-5	4.35E-5
5	2.44E-6	9.31E-6	2.71E-6	5.41E-6

TABLE 4.14: Comparison of error norms obtained by present method in example 4.11 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels

	Present Method		R. C. Mittal [207]	
t	L_2	L_∞	L_2	L_∞
0.5	2.26E-3	4.79E-3	3.48E-3	9.51E-3
1	1.31E-3	2.71E-3	3.23E-3	7.47E-3
2	5.21E-4	1.00E-3	2.85E-4	1.03E-3
3	1.30E-4	3.68E-4	3.10E-4	5.78E-4
4	5.01E-5	1.35E-4	9.08E-5	2.76E-4
5	2.89E-5	5.69E-5	2.44E-5	6.72E-5
7	1.90E-6	6.74E-6	2.53E-6	8.22E-6
10	1.00E-7	3.36E-7	3.65E-6	8.58E-6

TABLE 4.15: Comparison of error norms obtained by present method in example 4.12 for $N_x = N_y = 20$, $\Delta t = 0.001$ at different time levels

	Present Method		R. C. Mittal [207]	
t	L_2	L_∞	L_2	L_∞
0.5	3.31E-4	6.36E-4	3.58E-4	9.51E-4
1	1.61E-4	2.75E-4	3.23E-4	7.47E-4
2	8.52E-5	1.68E-4	2.85E-5	1.03E-4
3	5.89E-6	9.94E-6	3.10E-5	5.78E-4
5	4.52E-6	9.35E-6	2.44E-6	6.72E-5
7	4.14E-8	9.88E-8	2.53E-7	8.22E-7
10	2.94E-9	6.61E-9	3.65E-9	8.58E-8

Chapter 5

Numerical solution of fourth order partial differential equations using quintic trigonometric differential quadrature method

5.1 Introduction

In this chapter, we have gone through the numerical simulation of the generalized nonlinear fourth order partial differential equation:

$$u_t + \epsilon u u_x + \alpha u_{xx} + \mu u_{xxx} + \beta u_{xxxx} + f(u) = 0. \tag{5.1}$$

This type of nonlinear partial differential equation arises in many applications of theoretical, engineering and environmental sciences. Because of the involvement of nonlinearity and high order of derivatives, it is very difficult to find the exact solution of this equation, hence solving this equation numerically is a good option. Many researchers already did a few attempts to find the numerical solution of this type of equation. But there is no single method which can be used to solve a general fourth order partial differential equation that leads to some important equations such as Korteweg-de Vries, Kuramoto-Sivashinsky and extended

Fisher-Kolmogorov equations which contain the nonlinear terms with higher order derivatives. Because there is the presence of the high order derivatives in these equations cubic trigonometric B-spline basis function is not sufficient for commutating approximation of space derivatives so in this chapter we are using quintic trigonometric B-spline. On substituting the different values to the real constants $\epsilon, \alpha, \beta, \mu$ and function $f(u)$ many well known equations can be framed for example Fisher's, Burgers', convection-diffusion, sine-Gordon, Korteweg-de Vries, Kuramoto-Sivashinsky and extended Fisher-Kolmogorov equations *etc.* In this chapter partial differential equations containing fourth and third order derivatives are considered as the particular cases of the above mentioned generalized equation as discussed in the coming section.

5.1.1 Korteweg-de Vries (KdV) equation

By substituting $\alpha = \beta = 0$ and $f(u) = 0$ in equation (5.1) we get the KdV equation:

$$u_t + \epsilon u u_x + \mu u_{xxx} = 0 \tag{5.2}$$

where u is a field variable and μ stands for the dispersion parameter with $u \to 0$ as $x \to \infty$, and ϵ is real positive constant. This equation was utilized to depict the bearing of small amplitude shallow water waves in one space dimension. The vital property of equation (5.2) is that it has singular waves named solitons with the properties such as (i) it represents a wave of permanent form; (ii) it is localized, i.e. it decays or approaches a constant value at infinity; and (iii) it can interface emphatically with different solitons and hold its form, in particular, solitons spread without changing its shape and speed and are steady in shared connection simply like the marvel of absolutely versatile impact in energy. The third property is fairly astonishing. It is important to note that the nonlinear term $\epsilon u u_x$ in the equation makes the wave steep and unsteady while the dispersion term μu_{xxx} makes the wave changing its shape for eternity. In any case, the third property reveals that neither does the arrangement of equation (5.2) spread out nor steepens i.e. the two impacts keep up an adjustment and hence deliver a soliton. Throughout the years, the Korteweg-de Vries (KdV) equation has a wide application in many fields, for example, inside gravity waves in a stratified liquid, waves in a turning atmosphere, ion-acoustic waves in a plasma, pressure waves in a liquid-gas bubble mixture, and

wave phenomena in harmonic crystals. In recent years many researchers attempted to compute the exact and numerical solution to Korteweg-de Vries (KdV) equation using various methods, some of them are enlisted as [208–222].

5.1.2 Kuramoto-Sivashinsky (KS) equation

The generalized Kuramoto-Sivashinsky equation briefly known as KS equation is a model of the nonlinear partial differential equation which arises by mathematical modelling continuous media which exhibits a chaotic behavior and of the form:

$$u_t + \epsilon u u_x + \alpha u_{xx} + \mu u_{xxx} + \beta u_{xxxx} = 0 \tag{5.3}$$

where ϵ, α, μ, and β are real constant and is obtained from equation (5.1)with $f(u) = 0$. This equation was originally derived in the context of plasma instabilities, phase turbulence in reaction diffusion system and flame front propagation. It appears in the context of long waves on the interface between two viscous fluids, unstable drift waves in plasmas, and in flame front instability. The term u_{xx} describes instability at large scales, whereas the dissipative term u_{xxxx} gives damping at small scales. The nonlinear term uu_x act as a stabilizer by transferring energy between large and small scales. With $\alpha, \mu > 0$, the linear terms of the equation describes a balance between short-wave stability and long-wave instability while the nonlinear term provides a mechanism for transferring energy between wave modes. For $\alpha = \beta = 1$ and $\mu = 0$, this equation represents models of pattern formation in unstable flame fronts and thin hydrodynamic films. Many researchers have worked on the numerical, exact and analytic solution of this equation [223–242].

5.1.3 Extended Fisher-Kolmogorov equation

By substituting $\epsilon = \mu = 0, \alpha = -1$ and $f(u) = (u^3 - u)$ in the equation (5.1), we can obtain the extended Fisher-Kolmogorov equation given as:

$$u_t - u_{xx} + \beta u_{xxxx} + (u^3 - u) = 0. \tag{5.4}$$

This equation was introduced by Coullet et al. [243], and Dee and Van saarlos [244–246] was named as extended Fisher-Kolmogorov equation because this equation is a generalized form which gets converted to Fisher-Kolmogorov equation in standard form for $\beta = 0$. This well-known extended Fisher-Kolmogorov equation arises in a variety of applications such as pattern formation, spatiotemporal chaos in bi-stable systems and in phase transition near a Lifshitz point [247, 248]. It has been derived as an amplitude equation at the onset of instabilities near certain degenerate points [249]. Recently, attention has been given to the steady state of the equation which displays periodic, homoclinic or heteroclinic solutions, depending on β. Peletier and Troy [250, 251] studied the solution of an extended Fisher-Kolmogorov equation in steady state by adopting shooting method. Peletier et al. [250, 251], Bengurai et al. [252] and Dhanmjaya et al. [253] discussed the existence of a solution, its uniqueness, and regularity for the equation. Aghamohamadi et al. [254] worked on nonlinear Fisher's equation using tension spline method. These papers have differentiated and analyzed two different cases depending on the value of β, for $\beta \leq \frac{1}{8}$ and $\beta \geq \frac{1}{8}$, where the behavior of solutions is variable. It has been shown that for $\beta > \frac{1}{8}$ monotonicity is lost but there exists a unique solution for $\beta \leq \frac{1}{8}$.

The target of our employment is to explore the practicality of differential quadrature scheme with the quintic trigonometric B-spline basis to solve the concerned equations numerically, using different initial and boundary conditions in one dimension. The outcomes were observed to be empowering.

5.1.4 Implementation of the numerical scheme

On replacing the obtained first and high order space derivatives by their approximate values in equation (5.1) we get the following system:

$$u_t = -\epsilon u \sum_{j=1}^{N} a_{ij}\, u_j - \alpha \sum_{j=1}^{N} b_{ij}\, u_j - \mu \sum_{j=1}^{N} c_{ij}\, u_j - \beta \sum_{j=1}^{N} d_{ij}\, u_j + f(u_j) = R(u) \quad (5.5)$$

This system of ordinary differential equations is then solved by using SSP-RK43 scheme [72], which provides the numerical solution at different time levels.

5.2 Stability of the scheme

To check the stability of the scheme one can convert the given equations (5.1) to the system of ODEs and stability of the obtained system can be verified using the matrix method. On substituting the approximate values of derivatives in the equation (5.1) and taking the nonlinear terms as constant. This equation can be written in simplified form as follows:

$$u_t = Bu + F(u) \tag{5.6}$$

where $F(u)$ represent the nonlinear terms in the equation (5.1) and B takes different values as follows:

1. In Korteweg-de Vries (KdV) equation: $B = -k\mu \sum_{j=1}^{N} c_{ij}$
2. In extended Fisher-Kolmogorov equation: $B = -k\beta \sum_{j=1}^{N} d_{ij}$
3. In Kuramoto-Sivashinsky (KS) equation:$B = -k(\beta \sum_{j=1}^{N} d_{ij} + \mu \sum_{j=1}^{N} c_{ij} + \alpha \sum_{j=1}^{N} b_{ij})$

The stability of the system (5.6) depends on the eigenvalues of B. The system is said to be stable if the eigenvalues of the matrix B satisfy the conditions (discussed in chapter 1 section 1.6). Since all the eigenvalues of the matrix B presented in Figures 5.1, 5.2, 5.3 satisfy the stability conditions, hence the scheme is stable.

5.3 Numerical results and discussion

In this section, we have been simulated eight test problems of equation (5.1), and also examined the numerical results. The accuracy of obtained numerical results is measured by evaluating the L_2, L_∞, GRE and RMS errors.

Example 5.1. *Consider the KDV equation given by (5.2) with* $\epsilon = 1$ *and* $\mu = 4.84 \times 10^{-4}$ *with the domain* $[0, 2]$ *and the initial condition given by:*

$$u(x, 0) = 3csech^2(Ax + D)$$

and boundary conditions

$$u(0,t) = u(2,t) = 0,\ t > 0$$

The exact solution [222] for this equation is given as:

$$u(x,t) = 3csech^2(Ax - Bt + D)$$

where $A = \frac{1}{2}(\epsilon c/\mu)^{\frac{1}{2}}$, $B = \frac{1}{2}\epsilon cA$, $c = 0.3$ *and* $D = -6$. *Numerical results are computed at different time* $t = 0.5, 11.5, 2, 2.5, 3$ *and comparison of the obtained solution with the exact solution is shown in the form of Figures 5.4 and 5.5. Comparison of obtained solution with numerical solution computed in [222] is given in Table 5.1 at different times. It can be concluded from the comparison that results obtained by the present method are better than the results obtained in the literature.*

Example 5.2. *In the following examples, KDV equation (5.2) is considered with* $\epsilon = 6$ *and* $\mu = 1$ *for various space domains with different initial and boundary conditions taken from the exact solution given in [222]. Obtained results are also compared with results given in literature [222]. The different examples are given as follows:*

(a) *Consider the initial condition with the above mentioned parameters:*

$$u(x,0) = \frac{r}{2}sech^2(\frac{\sqrt{r}}{2}x - 7),\quad x \in [0, 40]$$

and boundary conditions taken from the exact solution given as:

$$u(x,t) = \frac{r}{2}sech^2(\frac{\sqrt{r}}{2}(x - rt) - 7),$$

where $r = 0.5$. *Numerical results are computed at different times* $t = 1$ *to* $t = 5$ *and comparison is made with the exact solution shown in form of Figures 5.6 and 5.7. Comparison of errors in numerical solution is given in Table 5.2 at different times.*

(b) *Consider the initial condition:*

$$u(x,0) = 12\Big(\frac{3 + 4cosh(2x) + cosh(4x)}{(3cosh(x) + cosh(3x))^2}\Big),\quad x \in [-5, 15]$$

and boundary conditions taken from the exact solution:

$$u(x,t) = 12\Big(\frac{3 + 4cosh(2x - 8t) + cosh(4x - 64t)}{(3cosh(x - 28t) + cosh(3x - 36t))^2}\Big).$$

Numerical results are computed at different times $t = 0.01, 0.05, 0.1, 0.2, 0.3$ *and comparison with the exact solution is shown in the form of Figures 5.8 and 5.9. Comparison of errors in numerical solution is given in Table 5.3 at different times.*

(c) *Consider the initial condition for the equation (5.2) as:*

$$u(x,0) = 2sech^2(x), \quad x \in [-5, 10]$$

and boundary conditions taken from the exact solution given as:

$$u(x,t) = 2sech^2(x - 4t).$$

Numerical results are computed at different times $t = 0.1, 0.3, 0.6, 0.9, 1.2, 1.5$ *and comparison with the exact solution is shown in the form of Figures 5.10 and 5.11. Error norms for this test problem are given in Table 5.4 at different times.*

(d) *Consider the initial condition:*

$$u(x,0) = \frac{r}{2}sech^2(\frac{\sqrt{r}}{2}x - 10), \quad x \in [30, 80]$$

and boundary conditions taken from the exact solution given as:

$$u(x,t) = \frac{r}{2}sech^2(\frac{\sqrt{r}}{2}(x - rt) - 10),$$

where $r = 0.14$. *Numerical results are computed at different times* $t = 1, 3, 5, 7, 10$ *and comparison with the exact solution is shown in the form of Figures 5.12 and 5.13. Comparison with numerical solutions obtained by other researchers is given in Table 5.5 at different times.*

(e) *Consider the initial condition for the equation (5.2) as:*

$$u(x,0) = 5\Big(\frac{4.5csch^2(1.5(x + 14.5)) + 2sech^2(x + 12)}{(3coth(1.5(x + 14.5)) - 2tanh(x + 12))^2}\Big), \quad x \in [-20, 0]$$

and boundary conditions taken from the exact solution given as:

$$u(x,t) = 5\Big(\frac{4.5csch^2(1.5(x-9t+14.5)) + 2sech^2(x-4t+12)}{(3coth(1.5(x-9t+14.5)) - 2tanh(x-4t+12))^2}\Big).$$

Numerical results are computed at different times $t = 0.1, 0.2, 0.3, 0.7, 1$ *and comparison with the exact solution is shown in form of Figures 5.14 and 5.15. Comparison with numerical solution is given in Table 5.6 at different times.*

Example 5.3. *Consider the KS equation (5.3) with parameter* $\epsilon = \alpha = \beta = 1$, $\mu = 4$. *The initial and boundary conditions are obtained from the exact solution:*

$$u(x,t) = c+9-15tanh(k(x-ct-x_0))-15tanh^2(k(x-ct-x_0))+15tanh^3(k(x-ct-x_0)),$$

where $x \in [-30, 30]$, $c = 6$, $k = \frac{1}{2}$ *and* $x_0 = -10$. *Numerical solution by the present scheme is obtained at different times* $t = 1, 2, 3, 4$ *and presented in Figures 5.20 and 5.21. These figures show the solitary wave propagation for KS-equation from time* $t = 1$ *to* $t = 4$. *For comparison of the obtained numerical results with results presented in literature, different error norms are computed and enlisted in Table 5.7. The obtained errors show that the present proposed scheme is better than the schemes proposed in [233] and [242].*

Example 5.4. *Consider the KS equation (5.3) with parameter* $\epsilon = \alpha = \beta = 1$, $\mu = 4$. *The initial and boundary conditions are obtained from the exact solution:*

$$u(x,t) = c-9-15tanh(k(x-ct-x_0))+15tanh^2(k(x-ct-x_0))+15tanh^3(k(x-ct-x_0)),$$

where $x \in [-30, 30]$, $c = -6$, $k = \frac{1}{2}$ *and* $x_0 = -10$. *Numerical solution by the present scheme is obtained at different times* $t = 0.5, 1, 1.5, 2$ *and presented in Figures 5.22 and 5.23. These figures show the solitary wave propagation for KS-equation from time* $t = 0.5$ *to* $t = 2$. *Different error norms like* L_2, L_∞ *and GRE are enlisted in Table 5.8.*

Example 5.5. *Consider the KS equation (5.3) with parameter* $\epsilon = \alpha = \beta = 1$, $\mu = 0$ *with initial condition:*

$$u(x,0) = exp(-x^2),\ x \in [-30, 30]$$

and boundary conditions:

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

Numerical solution by the present scheme is obtained at different times $t = 1, 5, 10, 20$ *and presented in Figure 5.24. This figure shows the solitary wave propagation for KS-equation from time* $t = 1$ *to* $t = 20$*. Obtained numerical results match with the results obtained by [242].*

Example 5.6. *Consider extended Fisher-Kolmogorov equation (5.4) with the initial condition in the space domain* $[-4, 4]$ *:*

$$u(x,0) = -sin(\pi x),$$

and with the boundary conditions:

$$u(-4,t) = 0,\ u(4,t) = 0.$$

Numerical solution of this test problem is computed at different times $t = 0, 0.05, 0.10, 0.15, 0.20$ *with* $k = 0.0001$*, and* $N = 251$*. The obtained results are presented in Figures 5.16 and 5.17 for* $\beta = 0$ *and* $\beta = 0.0001$ *respectively. The obtained numerical results are found in good match with results available in the literature [255].*

Example 5.7. *Consider extended Fisher-Kolmogorov equation (5.4) with the initial condition in the space domain* $[-4, 4]$*:*

$$u(x,0) = 10^{-3}exp(-x^2),$$

and with the boundary conditions:

$$u(-4,t) = 1,\ u(4,t) = 1.$$

Numerical solution of this test problem is computed at different times $t = 0.25, 1, 1.75, 2.50, 3.50, 4.50$ *with* $k = 0.001$ *and* $N = 121$*. The obtained results are presented in Figure 5.18 for* $\beta = 0.0001$*. The obtained numerical results from the proposed method are found in good match with results available in the literature [255].*

Example 5.8. *Consider extended Fisher-Kolmogorov equation (5.4) with the initial condition in the space domain* $[-4, 4]$ *:*

$$u(x, 0) = -10^{-3} exp(-x^2),$$

and with the boundary conditions:

$$u(-4, t) = -1,\ u(4, t) = -1.$$

Numerical solution of this test problem is computed at different times $t = 0.25, 1, 1.75, 2.50, 3.50, 4.50$ *with* $k = 0.001$ *and* $N = 121$*. The obtained results are presented in Figure 5.19 for* $\beta = 0.0001$*. The obtained numerical results from the proposed method are found in good match with results available in the literature [255].*

5.4 Summary

In this chapter, the numerical solution of a generalized fourth order partial differential equation is computed using developed hybrid scheme in which differential quadrature method is implemented with quintic trigonometric B-spline basis function. The considered general equation reduces to three well-known equations namely: Korteweg-de Vries equation extended Fisher-Kolmogorov equation and Kuramoto-Sivashinsky equation for different choices of constants. The previous section of numerical results and discussion demonstrate the good accuracy of the proposed method. Various numerical examples are taken from each obtained equation. While implementing the proposed scheme, the generalized equation is converted to a system of ordinary differential equations which is further solved using the SSP-RK43 method. Since errors are the best way to justify the working and ability of a method, the L_2, L_∞ and RMS errors are calculated and presented to show the robustness of the method. Stability of the scheme is also evaluated with the help of eigenvalues and the scheme is found to be unconditionally stable. As a summary, this method proved to be a significant tool in solving the nonlinear partial differential equations of fourth order.

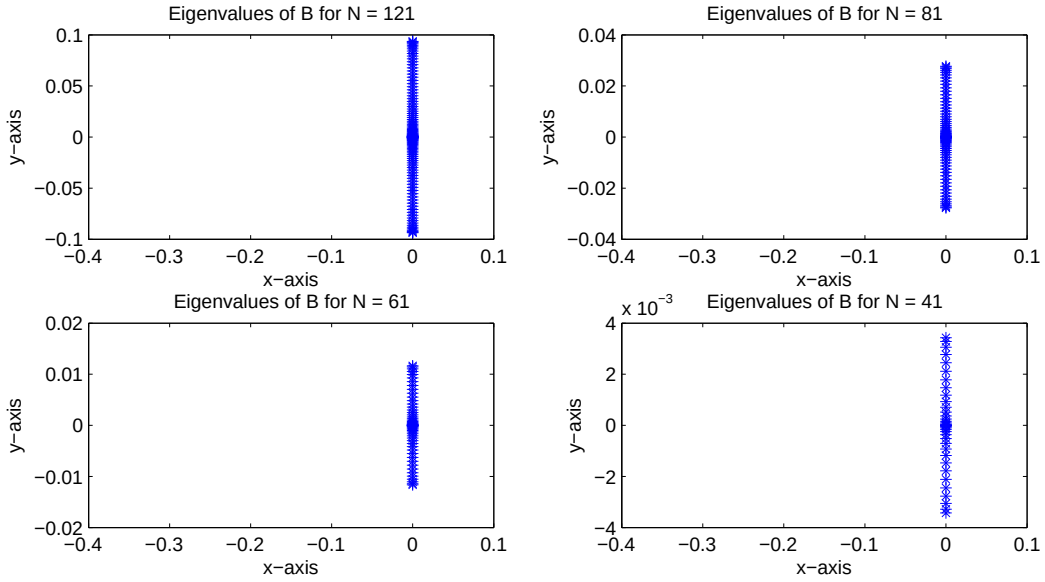

FIGURE 5.1: Eigenvalues of the matrix B for KDV equation with different partitions of the domain in one dimension

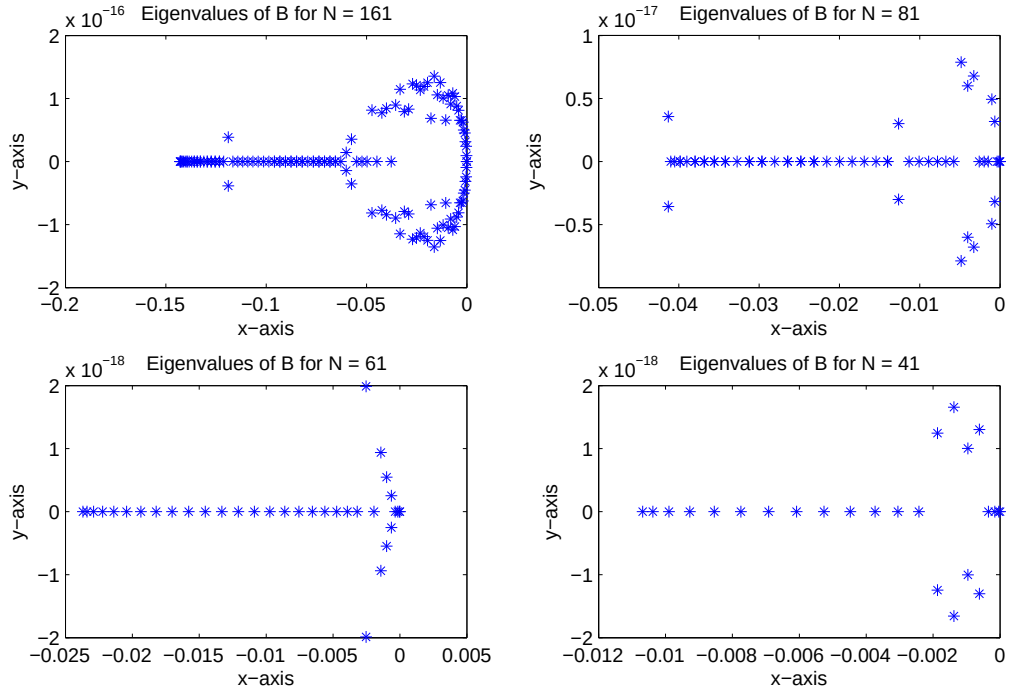

FIGURE 5.2: Eigenvalues of the matrix B for extended Fisher's-Kolmogorov equation with different partitions of the domain in one dimension

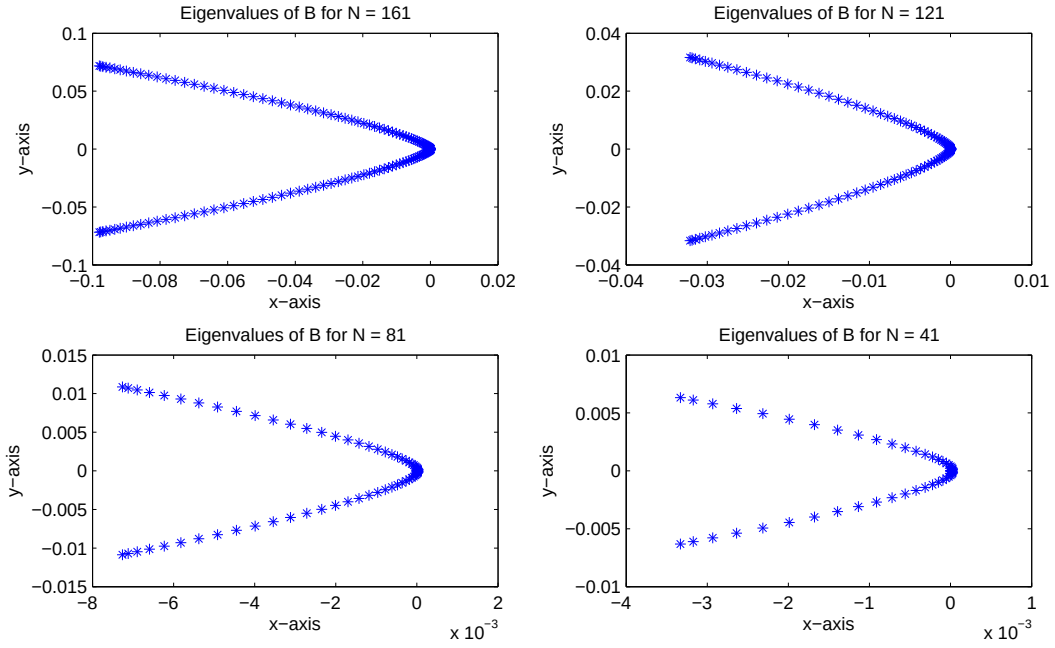

FIGURE 5.3: Eigenvalues of the matrix B for KS equation with different partitions of the domain in one dimension

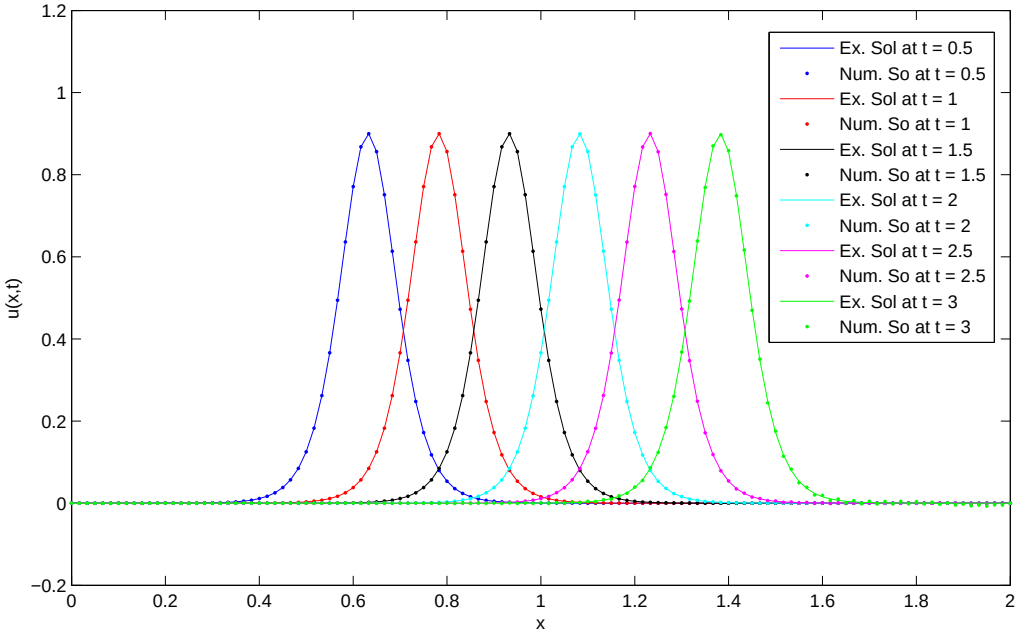

FIGURE 5.4: Comparison of the exact and numerical solutions of example 5.1 for time $t \leq 3$ with $\Delta t = 0.001$ and $N = 121$

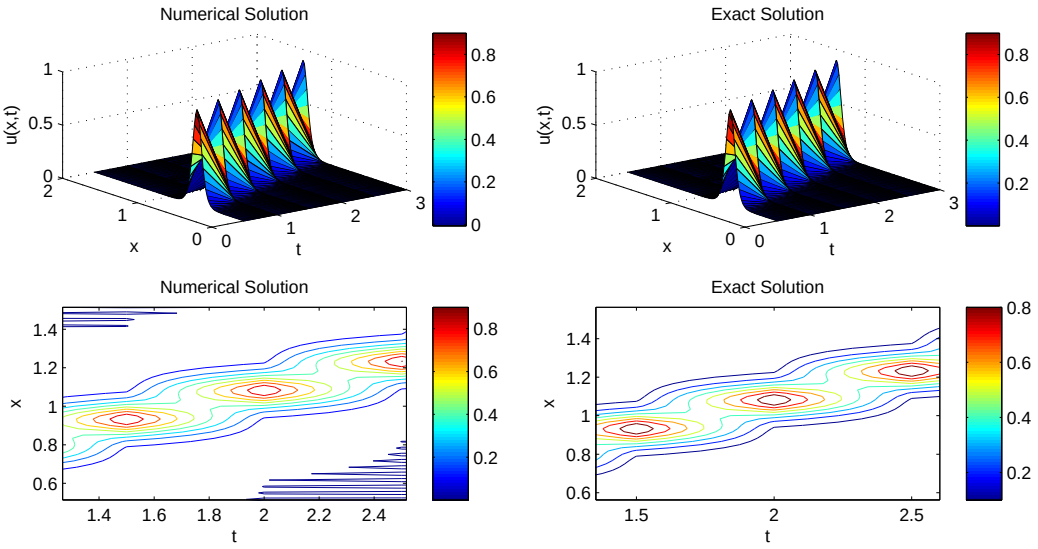

FIGURE 5.5: Surface and contour plot of the exact and numerical solution of example 5.1 for time $t \leq 3$ with $\Delta t = 0.001$ and $N = 121$

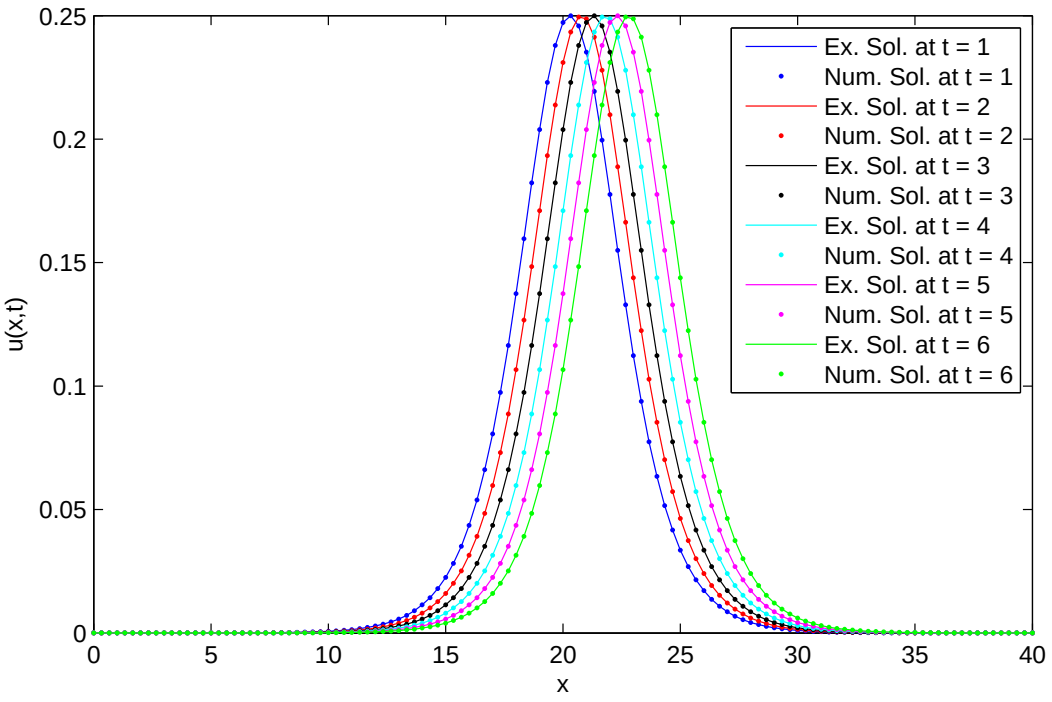

FIGURE 5.6: Comparison of the exact and numerical solutions of example 5.2(a) for time $t \leq 6$ with $\Delta t = 0.0001$ and $N = 121$

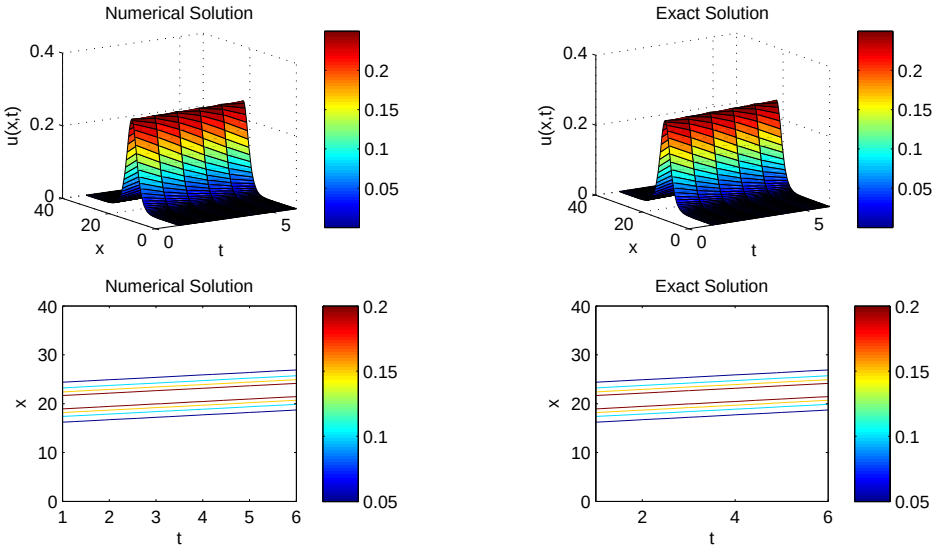

FIGURE 5.7: Surface and contour plot of the exact and numerical solution of example 5.2(a) for time $t \leq 6$ with $\Delta t = 0.0001$ and $N = 121$

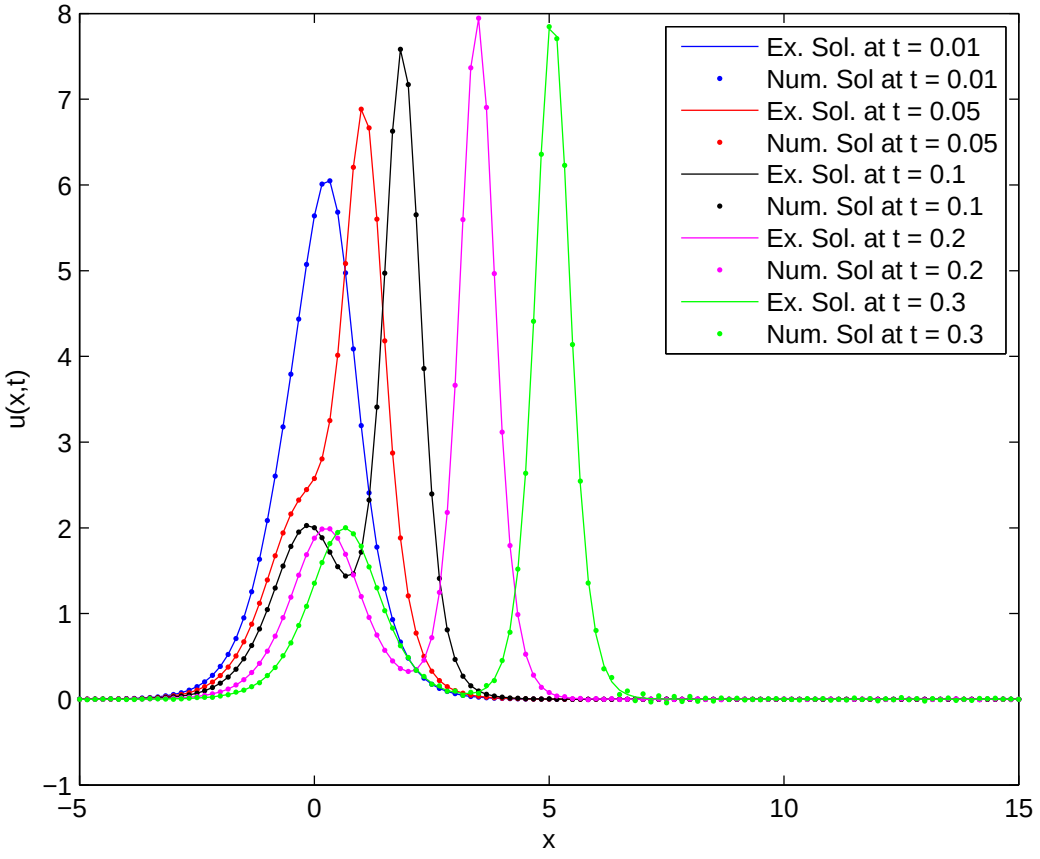

FIGURE 5.8: Comparison of the exact and numerical solutions of example 5.2(b) for time $t \leq 0.3$ with $\Delta t = 0.00001$ and $N = 121$

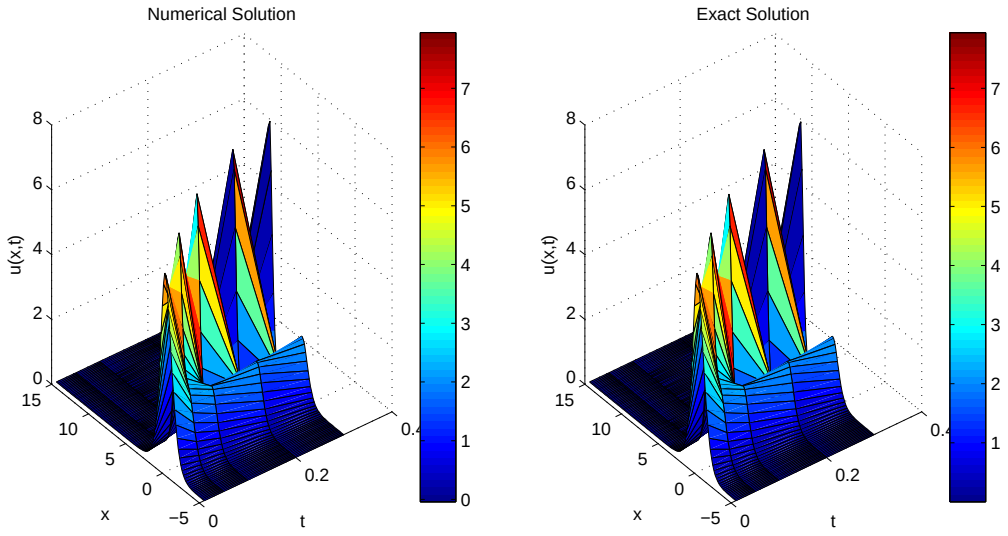

FIGURE 5.9: Surface and contour plot of the exact and numerical solution of example 5.2(b) for time $t \leq 0.3$ with $\Delta t = 0.00001$ and $N = 121$

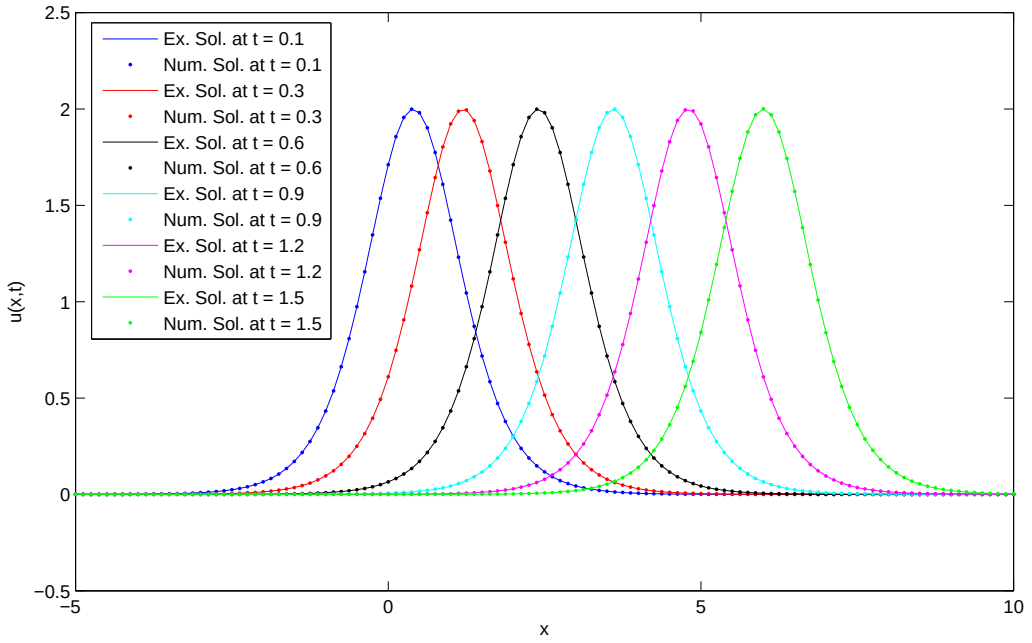

FIGURE 5.10: Comparison of the exact and numerical solutions of example 5.2(c) for time $t \leq 1.5$ with $\Delta t = 0.0001$ and $N = 121$

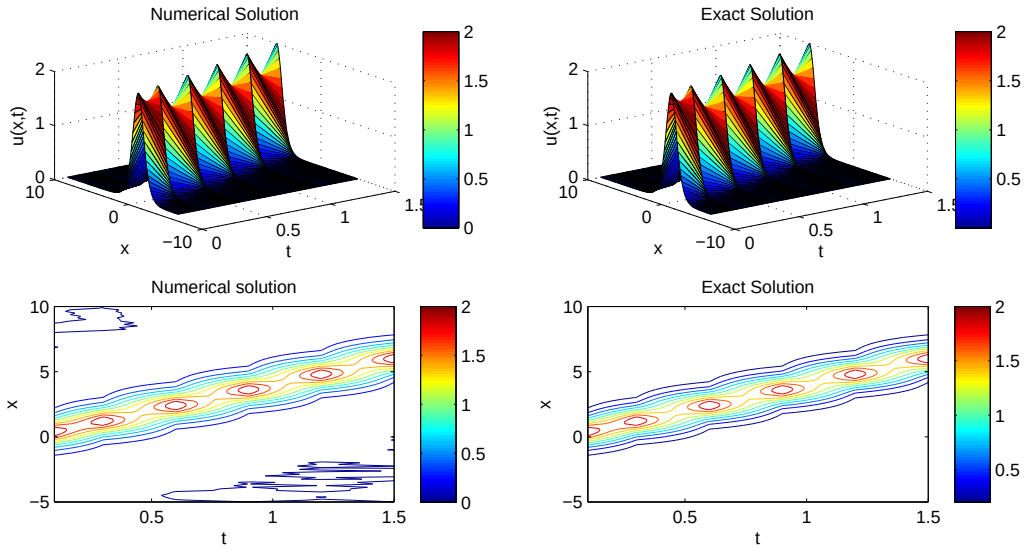

FIGURE 5.11: Surface and contour plot of the exact and numerical solution of example 5.2(c) for time $t \leq 1.5$ with $\Delta t = 0.0001$ and $N = 121$

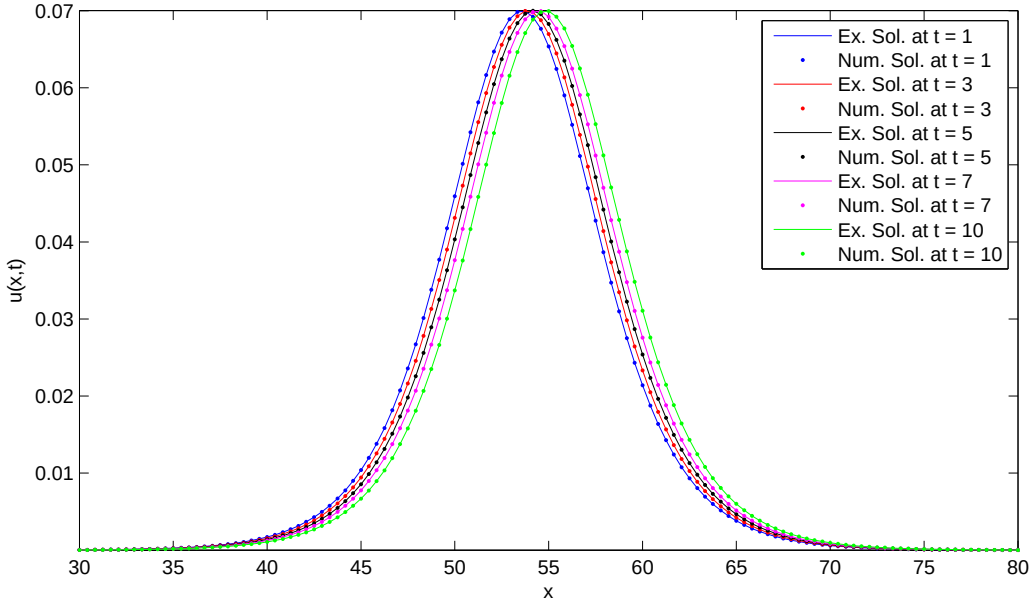

FIGURE 5.12: Comparison of the exact and numerical solutions of example 5.2(d) for time $t \leq 10$ with $\Delta t = 0.001$ and $N = 121$

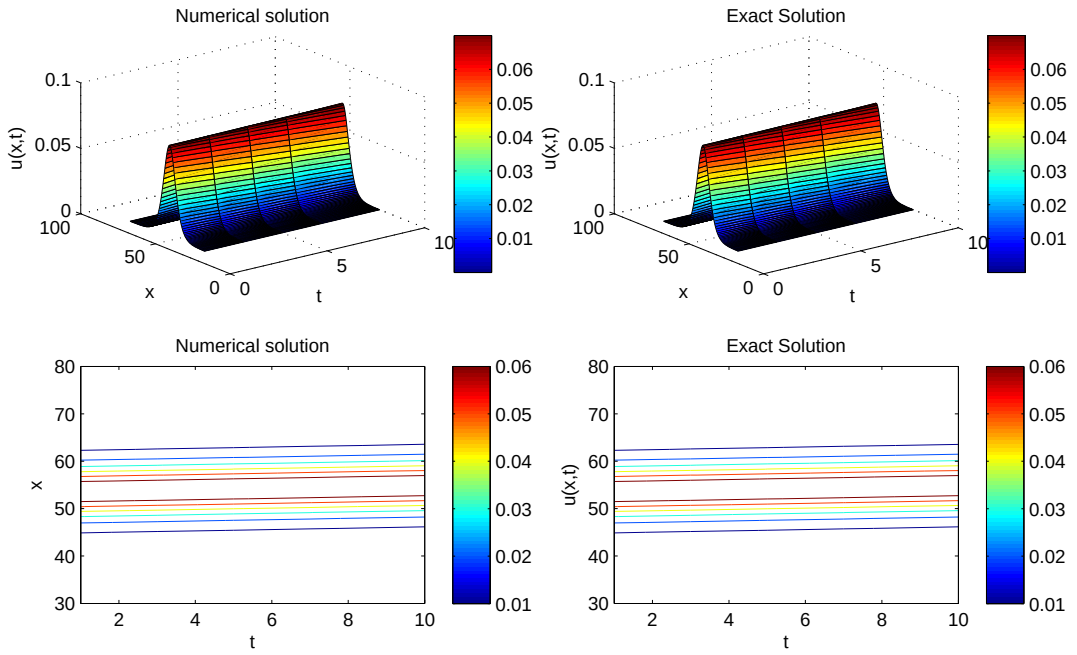

FIGURE 5.13: Surface and contour plot of the exact and numerical solution of example 5.2(d) for time $t \leq 10$ with $\Delta t = 0.001$ and $N = 121$

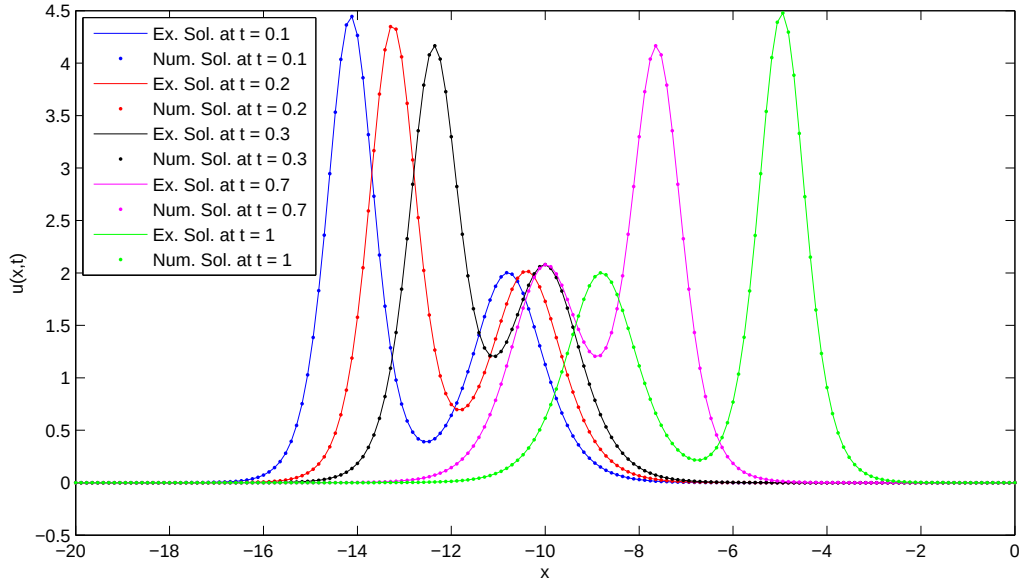

FIGURE 5.14: Comparison of the exact and numerical solutions of example 5.2(e) for time $t \leq 1$ with $\Delta t = 0.0001$ and $N = 171$

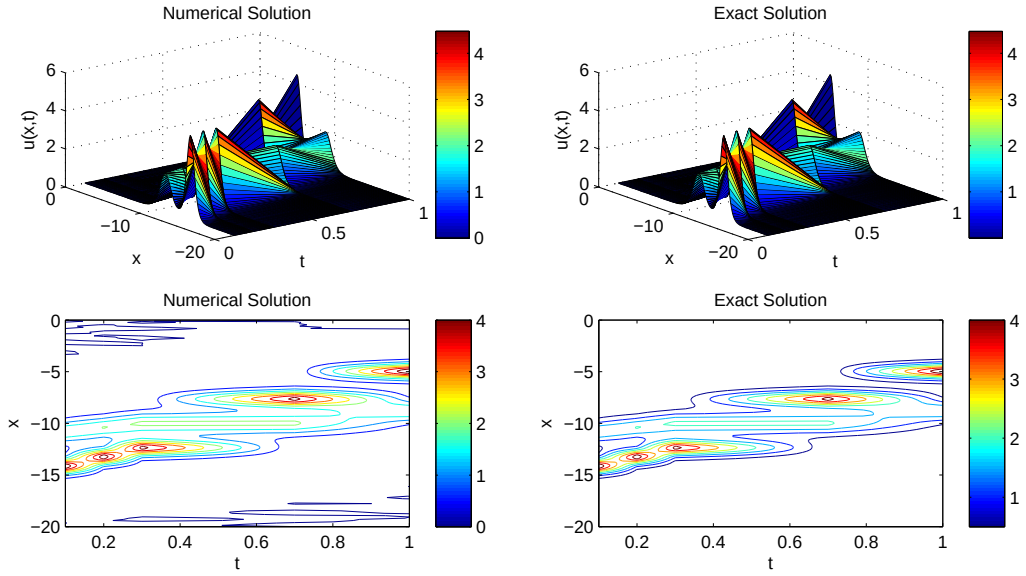

FIGURE 5.15: Surface and contour plot of the exact and numerical solution of example 5.2(e) for time $t \leq 1$ with $\Delta t = 0.0001$ and $N = 171$

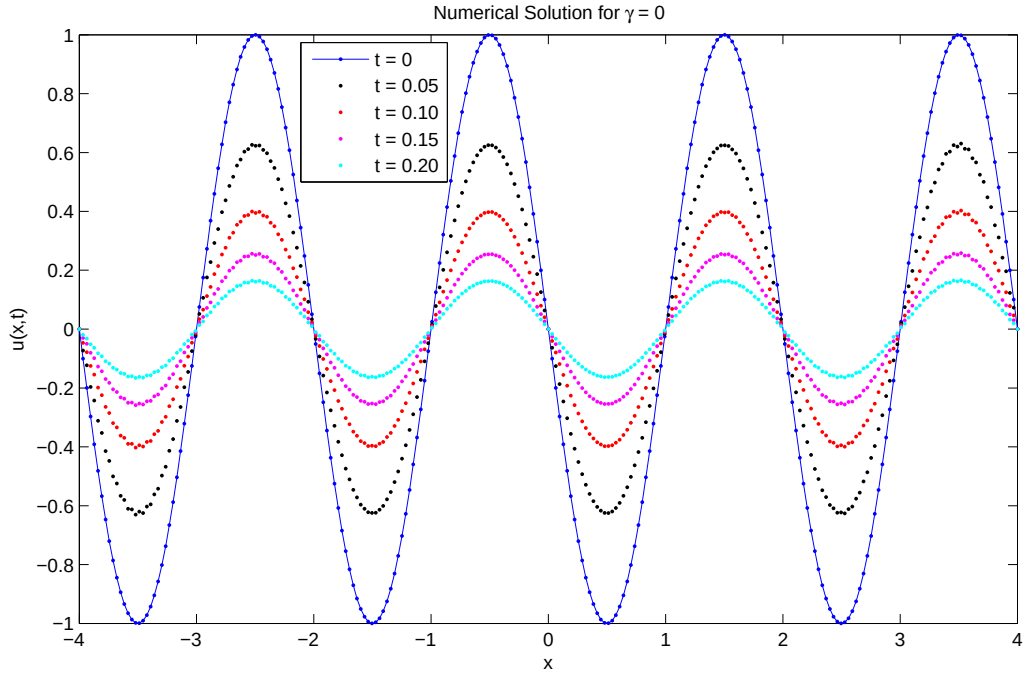

FIGURE 5.16: Comparison of numerical solutions of example 5.6 for different time $t \leq 0.20$ with $\Delta t = 0.0001$, $\gamma = 0$ and $N = 251$

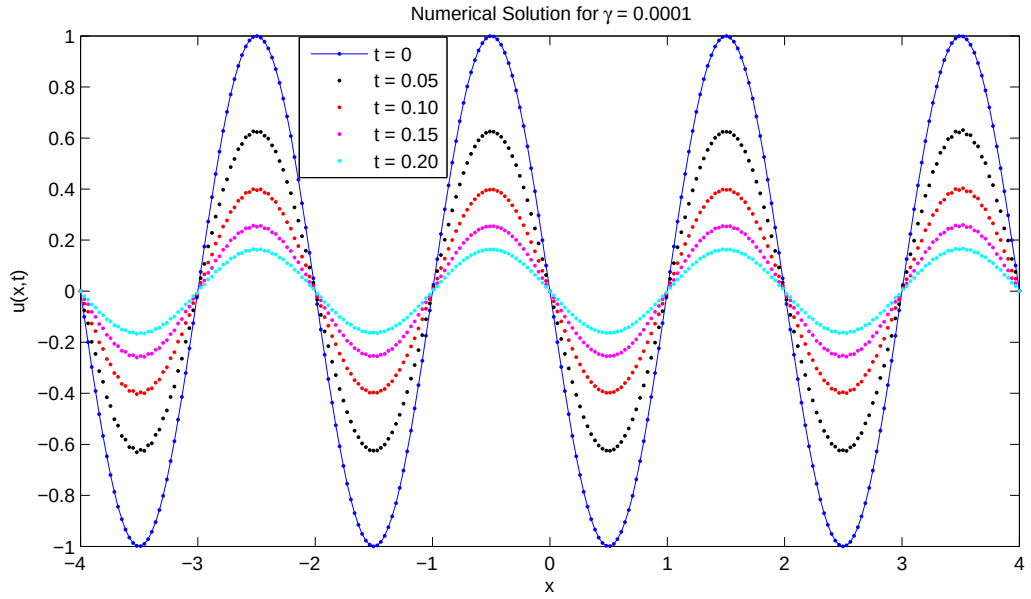

FIGURE 5.17: Comparison of numerical solutions of example 5.6 for different time $t \leq 0.20$ with $\Delta t = 0.0001$, $\gamma = 0.0001$ and $N = 251$

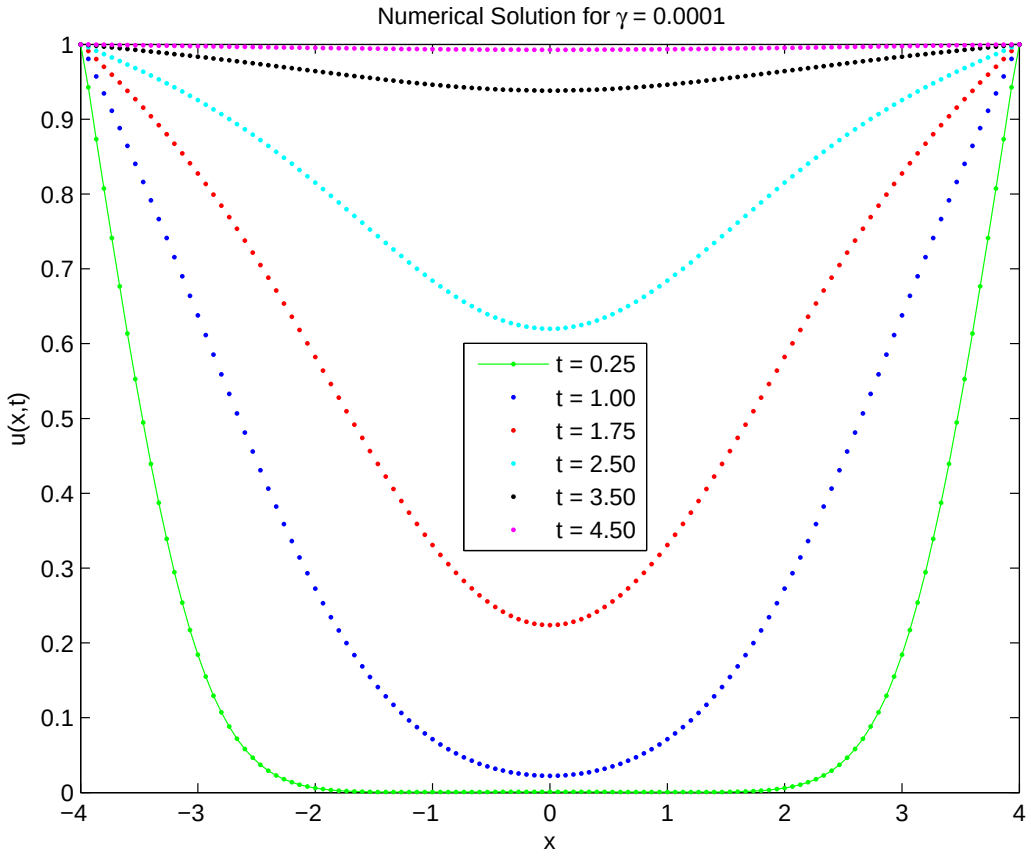

FIGURE 5.18: Comparison of numerical solutions of example 5.7 for different time $t \leq 4.50$ with $\Delta t = 0.001$, $\gamma = 0.0001$ and $N = 121$

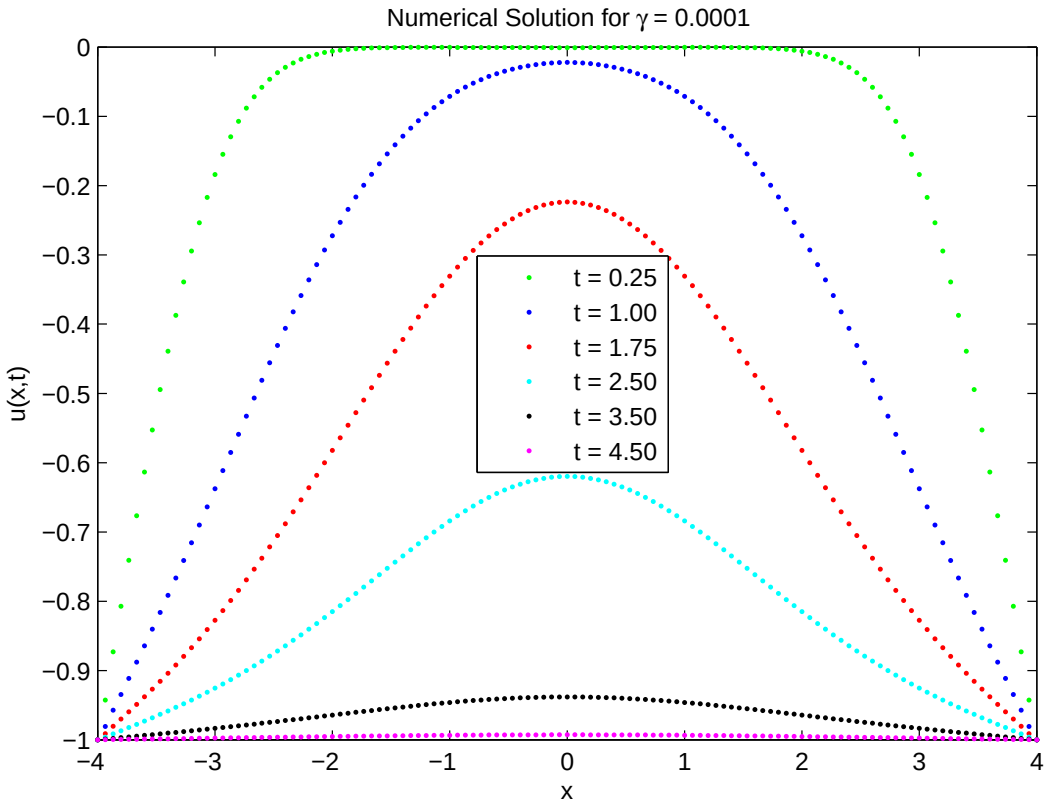

FIGURE 5.19: Comparison of numerical solutions of example 5.8 for different time $t \leq 4.50$ with $\Delta t = 0.001$, $\gamma = 0.0001$ and $N = 121$

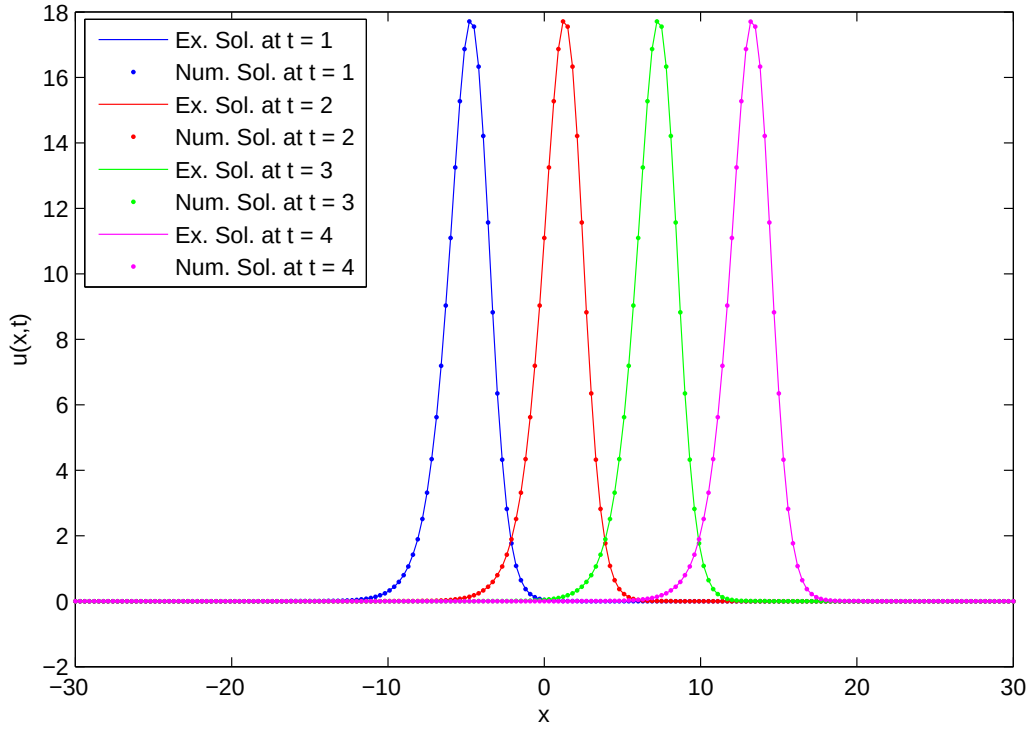

FIGURE 5.20: Comparison of numerical solutions of example 5.3 for different time $t \leq 4$ with $\Delta t = 0.001$, and $N = 201$

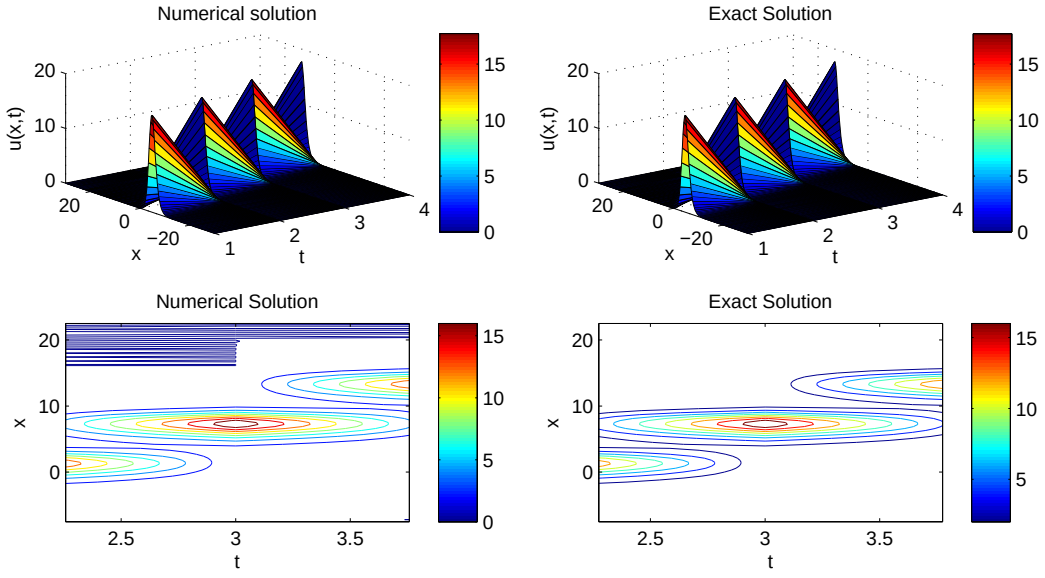

FIGURE 5.21: Surface and contour plot of the exact and numerical solution of example 5.4 for different time $t \leq 4$ with $\Delta t = 0.001$, and $N = 201$

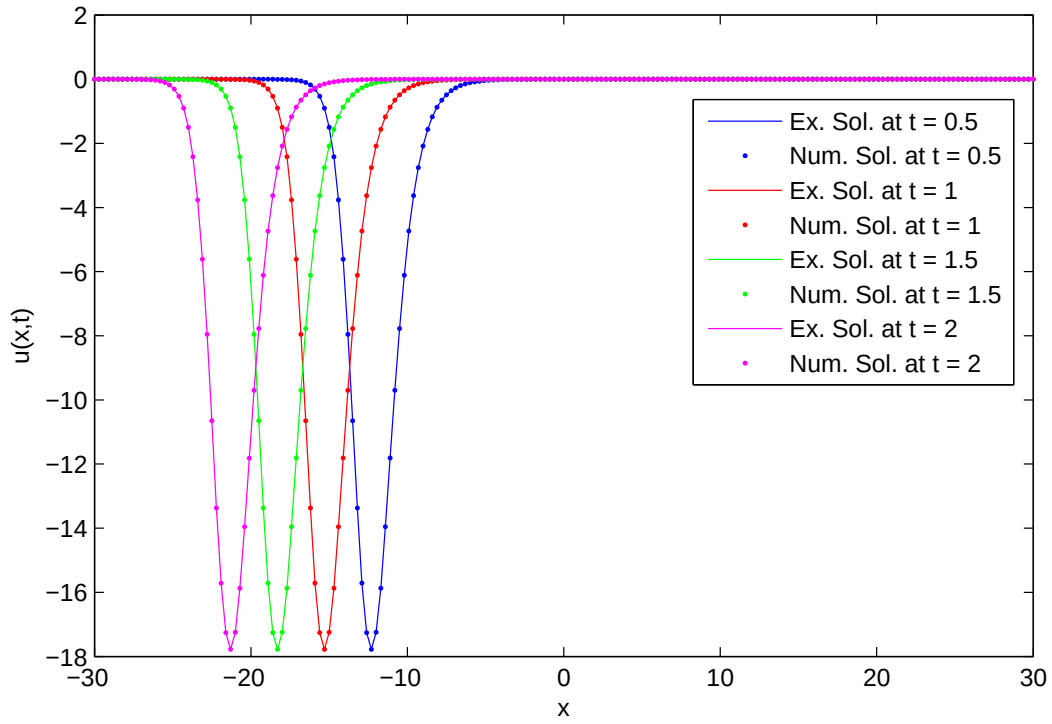

FIGURE 5.22: Comparison of numerical solutions of example 5.4 for different time $t \leq 2$ with $\Delta t = 0.001$, and $N = 201$

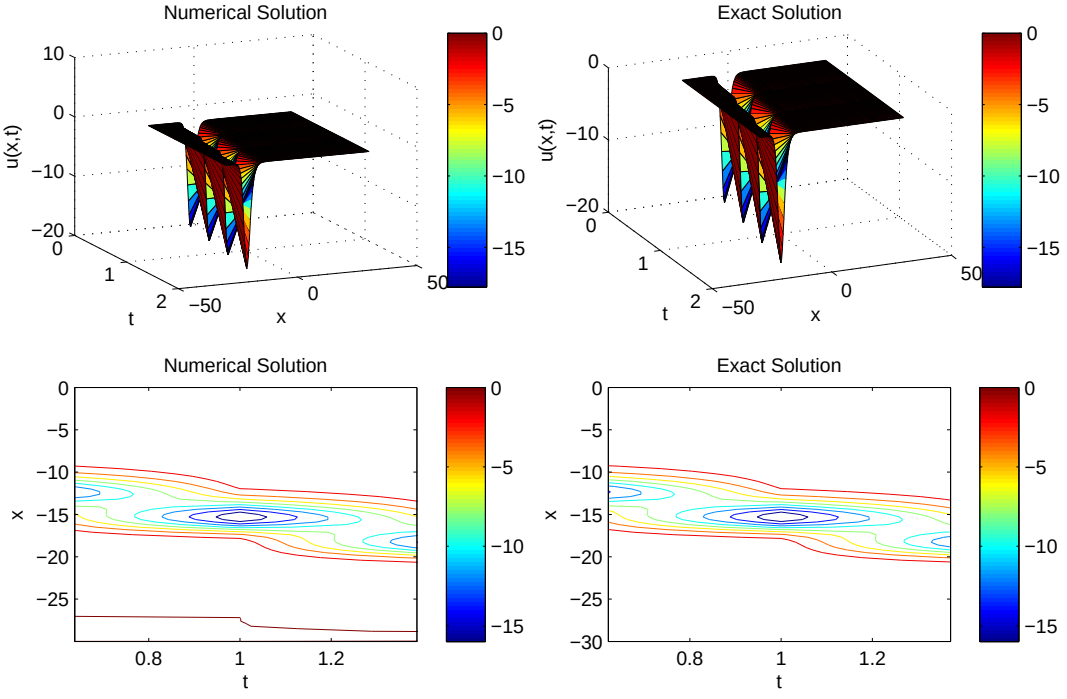

FIGURE 5.23: Surface and contour plot of the exact and numerical solution of example 5.4 for different time $t \leq 2$ with $\Delta t = 0.001$, and $N = 201$

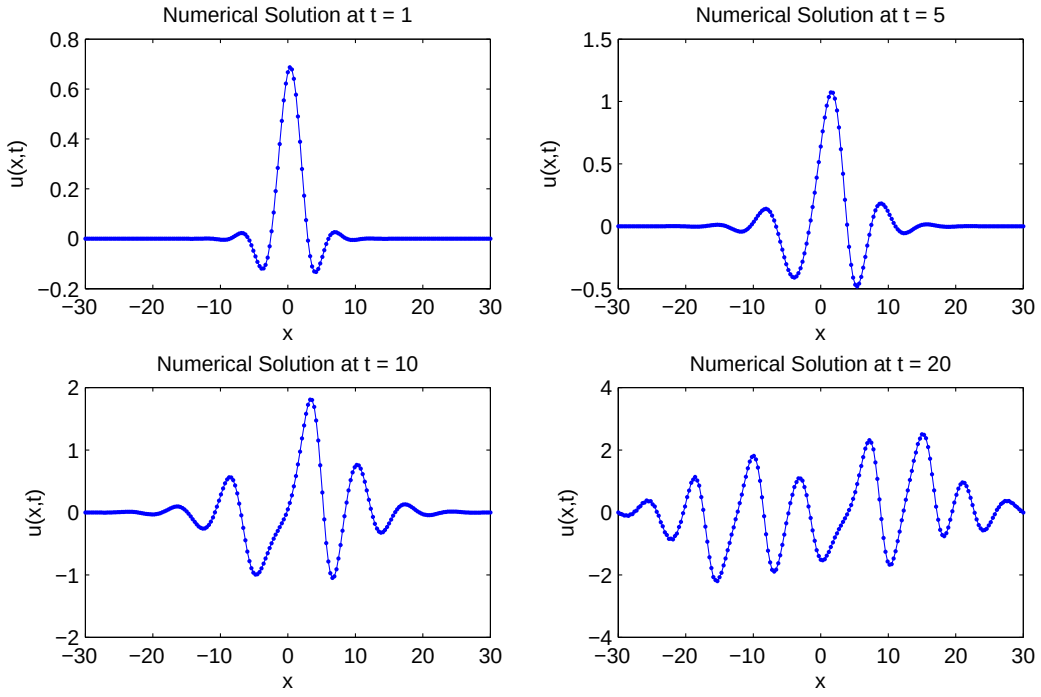

FIGURE 5.24: Numerical solution of example 5.5 for different times $t = 1, 5, 10, 20$ with $\Delta t = 0.001$, and $N = 201$

TABLE 5.1: Comparison of errors norms in the numerical solutions computed by present method for Example 5.1 at different time levels with $N = 121$, $\Delta t = 0.001$

	Present Method			Dehghan and Shokri [222]		
Time	L_2	L_∞	RMS	L_2	L_∞	RMS
0.5	6.35E-6	2.29E-5	4.47E-6	2.52E-3	7.23E-4	1.78E-4
1	1.14E-5	3.35E-5	8.04E-6	6.21E-3	1.79E-3	4.38E-4
1.5	3.34E-5	7.29E-5	2.35E-5	1.30E-2	3.89E-3	9.17E-4
2	1.32E-4	2.00E-4	9.35E-05	2.29E-2	6.67E-3	1.61E-3
2.5	5.82E-4	8.29E-4	4.10E-4	-	-	-
3	2.89E-3	7.13E-3	2.03E-3	-	-	-

TABLE 5.2: Comparison of errors norms in the numerical solutions computed by present method for example 5.2(a) at different time levels with $N = 121$, $\Delta t = 0.0001$

	Present Method			Dehghan and Shokri[222]		
Time	L_2	L_∞	RMS	L_2	L_∞	RMS
1	3.19E-7	1.17E-7	5.03E-8	6.23E-5	1.80E-5	4.39E-6
2	5.29E-7	2.94E-7	8.33E-8	1.12E-4	3.03E-5	7.94E-6
3	7.45E-7	4.82E-7	1.17E-7	1.55E-4	4.00E-5	1.09E-5
4	9.86E-7	6.35E-7	1.55E-7	1.94E-4	4.83E-5	1.36E-5
5	1.31E-6	8.54E-7	2.07E-7	2.29E-4	5.60E-5	1.61E-5
6	1.81E-6	1.22E-6	2.85E-7	-	-	-

TABLE 5.3: Comparison of errors norms in the numerical solutions computed by present method for example 5.2(b) at different time levels with $N = 121$, $\Delta t = 0.00001$

	Present Method			Dehghan and Shokri[222]		
Time	L_2	L_∞	RMS	L_2	L_∞	RMS
0.01	1.00E-4	1.09E-4	2.24E-5	-	-	-
0.05	6.93E-4	9.42E-4	1.54E-4	-	-	-
0.1	2.86E-3	3.12E-3	6.37E-4	1.54E-2	5.63E-3	1.08E-3
0.2	1.35E-2	8.38E-3	3.02E-3	6.22E-2	2.33E-2	4.39E-3
0.3	8.49E-2	5.21E-2	1.89E-2	1.63E-1	5.94E-2	1.15E-2

TABLE 5.4: Comparison of errors obtained by applying the present methods in example 5.2(c) for $N = 121$, $\Delta t = 0.0001$ at different time levels

t	L_2	L_∞	RMS
0.1	5.25E-5	3.68E-5	1.35E-5
0.3	7.25E-5	4.50E-5	1.87E-5
0.6	1.48E-4	1.05E-4	3.83E-5
0.9	4.39E-4	4.11E-4	1.12E-4
1.2	1.02E-3	9.74E-4	2.62E-4
1.5	1.56E-3	1.39E-3	4.03E-4

TABLE 5.5: Comparison of errors norms in the numerical solutions computed by present method for example 5.2(d) at different time levels with $N = 121$, $\Delta t = 0.001$

	Present Method			Dehghan and Shokri [222]		
Time	L_2	L_∞	RMS	L_2	L_∞	RMS
1	1.30E-5	7.16E-6	1.82E-6	2.13E-5	6.88E-6	1.35E-6
3	1.69E-5	7.52E-6	2.38E-6	3.49E-5	8.59E-6	2.20E-6
5	2.01E-5	9.58E-6	2.84E-6	4.10E-5	8.39E-6	2.59E-6
7	2.02E-5	8.04E-6	2.85E-6	4.27E-5	9.20E-6	2.70E-6
10	2.02E-5	7.69E-6	2.84E-6	4.54E-5	8.56E-6	2.87E-6

TABLE 5.6: Comparison of errors norms in the numerical solution computed by present method for example 5.2(e) at different time levels with $N = 171$, $\Delta t = 0.0001$

	Present Method			Dehghan and Shokri[222]		
Time	L_2	L_∞	RMS	L_2	L_∞	RMS
1	1.88E-5	2.42E-5	4.18E-6	1.35E-2	4.12E-3	1.34E-3
3	2.03E-5	2.28E-5	4.53E-6	1.96E-2	5.02E-3	1.95E-3
5	1.72E-5	1.70E-5	3.82E-6	2.25E-2	5.92E-3	2.24E-3
7	2.37E-5	2.09E-5	5.28E-6	2.98E-2	1.14E-2	2.97E-3
10	1.59E-4	1.51E-4	3.55E-5	6.88E-2	2.48E-2	6.84E-3

TABLE 5.7: Comparison of errors norms in the numerical solution computed by present method for example 5.3 at different time levels with $N = 171$, $\Delta t = 0.0001$

	Present Method			Mittal [242]		Lai [233]
Time	L_2	L_∞	$GRE(N = 201)$	$GRE(N = 150)$	$GRE(N = 200)$	$GRE(N = 150)$
1	3.13E-5	2.62E-5	1.75E-6	2.56E-3	5.04E-4	1.59E-4
2	3.82E-5	3.74E-5	2.25E-6	4.90E-3	1.00E-3	3.20E-4
3	1.19E-4	7.51E-5	9.80E-6	1.11E-2	2.33E-3	7.30E-4
4	1.54E-3	4.69E-4	1.25E-4	1.91E-2	4.00E-3	1.26E-3

TABLE 5.8: Comparison of errors norms in the numerical solution computed by present method for example 5.4 at different time levels with $N = 171$, $\Delta t = 0.0001$

	Present Method		
Time	L_2	L_∞	$GRE(N = 201)$
0.5	3.03E-5	2.36E-5	1.58E-6
1	3.13E-5	2.41E-5	1.75E-6
1.5	3.36E-5	2.88E-5	1.91E-6
2	3.76E-5	3.40E-5	2.09E-6

Chapter 6

Numerical solution of extended Fisher-Kolmogorov equation in two dimensions using quintic trigonometric differential quadrature method

6.1 Introduction

In this chapter, we will concentrate on the numerical solution of the extended Fisher-Kolmogorov partial differential equation in two dimensions:

$$u_t + \gamma(u_{xxxx} + u_{yyyy}) - (u_{xx} + u_{yy}) + f(u) = g(x, y, t), \tag{6.1}$$

or this equation can be written as:

$$u_t + \gamma\Delta^2 u - \Delta u + f(u) = g(x, y, t),\ (x, y) \in [a, b] \times [c, d], \tag{6.2}$$

with initial and boundary conditions

$$u(x, y, t) = u_0(x, y),$$

$$u = 0, \Delta u = 0, \text{ on boundary}$$

where γ is a positive constant and $f(u) = (u^3 - u)$. In chapter 5, one dimensional extended Fisher-Kolmogorov equation has been solved as the particular case of generalized fourth order. This equation was introduced by Coullet et al. [243], Dee and Van saarlos [244–246] and named as extended Fisher-Kolmogorov equation because this equation is a generalized form which gets converted to Fisher-Kolmogorov equation in standard form for $\gamma = 0$. Its applications, properties and literature review has been discussed in detail in the previous chapter where this equation is solved in one dimension.

The target of our employment is to explore the practicality of differential quadrature scheme with the trigonometric B-spline basis to solve the concerned equation in two dimensions numerically, using different initial and boundary conditions. In the previous chapter, extended Fisher-Kolmogorov equations in one dimension have been solved with quintic trigonometric differential quadrature method. This is an extension of the previous work in two dimensions.

6.2 Implementation of the numerical scheme

On replacing the obtained first and high order space derivatives in equation (6.1) by their approximation as given in first chapter, we get the following system:

$$u_t + \gamma(\sum_{j=1}^{N} d_{ij}\ u_j(x,y) + \sum_{j=1}^{N} d'_{ij}\ u_j(x,y)) - (\sum_{j=1}^{N} b_{ij}\ u_j + \sum_{j=1}^{N} b'_{ij}\ u_j) + f(u_j) = g(x,y,t) \tag{6.3}$$

where b_{ij}, d_{ij}, b'_{ij} and d'_{ij} are the coefficient matrices as defined in section 1.4 of chapter 1, to calculate the approximations of space derivatives in x and y directions. Now to solve the above system of ODE we use strong stability-preserving time-stepping RungeKutta (SSP-RK43) scheme [72].

6.3 Stability of the scheme

To check the stability of the scheme one can convert the given equations (6.1) to system of ODEs and verify its stability using the matrix method as discussed in

the section 1.6 of chapter 1. On substituting the approximate values of derivatives in equation (6.1) and taking the nonlinear terms as constant. The equation can be written in simplified form as follows:

$$u_t = Bu + F(u) \tag{6.4}$$

where $F(u)$ represent the nonlinear terms in the equation and $B = -k\gamma \sum_{j=1}^{N} d_{ij}$. The stability of the system (6.4) depends on the eigenvalues of B. The system is said to be stable if the eigenvalues of the matrix B satisfy the conditions given by [73]. From this discussion, it can be concluded that the stability of the obtained system depends upon the eigenvalues of matrix B. Figure 6.1 presents the eigenvalues for different domain partitions. As visualized from this figure, all the eigenvalues of matrix B are laying in the stability region. Hence it can be concluded that the scheme is stable.

6.4 Numerical results and discussion

In this section we have simulated two test problems of equation (6.1), and examined the numerical results. The accuracy of obtained numerical results is measured by evaluating the numerical errors.

Example 6.1. *Consider the EFK equation given by (6.1) with the domain* $(x, y) \in [0, 1] \times [0, 1]$ *and the initial condition given by [256]:*

$$u(x, y, 0) = x^3(1 - x)^3 y^3(1 - y)^3$$

and on boundary

$$u = 0, \Delta u = 0$$

Numerical results are calculated of this equation for $\gamma = 0.0001$, $f(u) = (u^3 - u)$, $g(x, y, t) = 0$ *with* $\Delta t = 0.00001$, *and* $N = M = 31$ *at different time levels. Physical behaviour of the solution is presented in Figure 6.2.*

Example 6.2. *In this example, EFK equation (6.1) in two dimensions is considered with* $\gamma = 0.00001$ *and space domain* $(x, y) \in [0, 1] \times [0, 1]$ *with initial and*

boundary conditions given by the exact solution [256]:

$$u(x, y, t) = sin(2\pi x)sin(2\pi y)e^{-t}$$

and the value of $g(x, y, t)$ *in the equation (6.1) taken as:*

$$sin(2\pi x)sin(2\pi y)e^{-t}(sin(2\pi x)^2 sin(2\pi y)^2 e^{(-t)^2} + 64\gamma(\pi)^4 + 8(\pi)^2)$$

Numerical solution of this equation is computed at different time levels (t=0.25, 0.75, 1, 1.25) and are presented in the Figures 6.3, 6.4, 6.5, 6.6 respectably. In Table 6.1, different error norms are calculated at different times. Comparison of the error norms is given in Table 6.2 at different domain partition.

6.5 Summary

In this chapter, the numerical solution of the extended Fisher-Kolmogorov equation in two dimensions is computed by developing and hence implementing a hybrid scheme, in which differential quadrature method is used with quintic trigonometric B-spline basis function. The previous section of numerical results and discussion demonstrates the good accuracy of the proposed method. The outcomes are observed to be empowering. Two test examples are taken from the literature to simulate the numerical scheme for the equation. While implementing the proposed scheme, the considered equation is converted to a system of ordinary differential equations which is further solved using the SSP-RK43 method. Since errors are the best way to justify working and ability of a method, numerical errors are calculated and presented to show the robustness of the method. Stability of the scheme is also evaluated with the help of eigenvalues and the scheme is found to be unconditionally stable. As a summary, this method proved to be a significant tool in solving the nonlinear partial differential equations of fourth order in two dimensions.

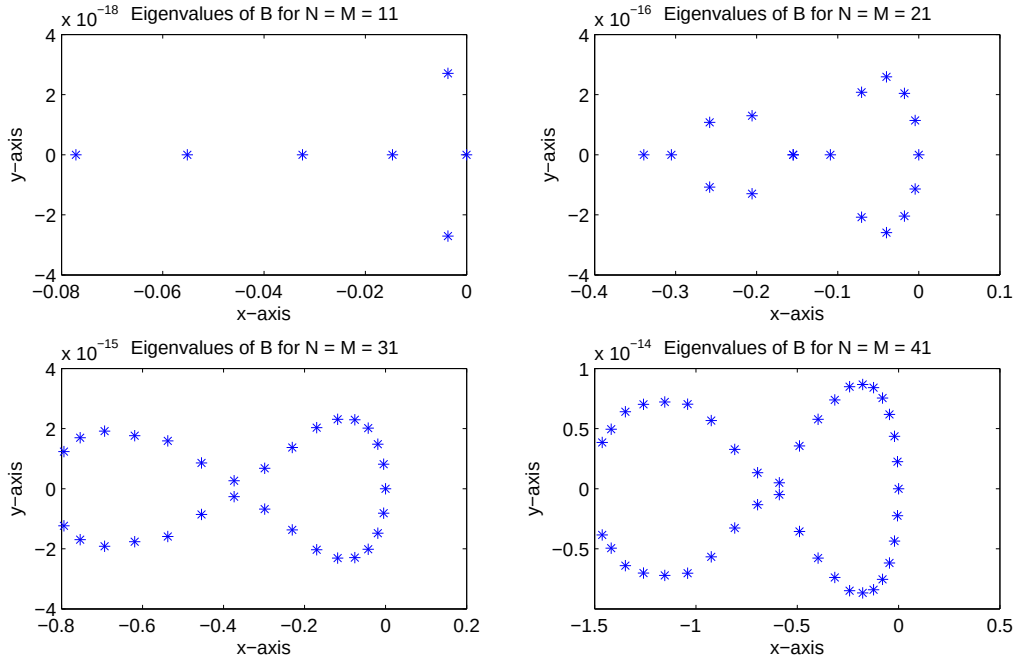

FIGURE 6.1: Eigenvalues of the matrix B for an extended Fisher's-Kolmogorov equation with different partitions of the domain in two dimensions

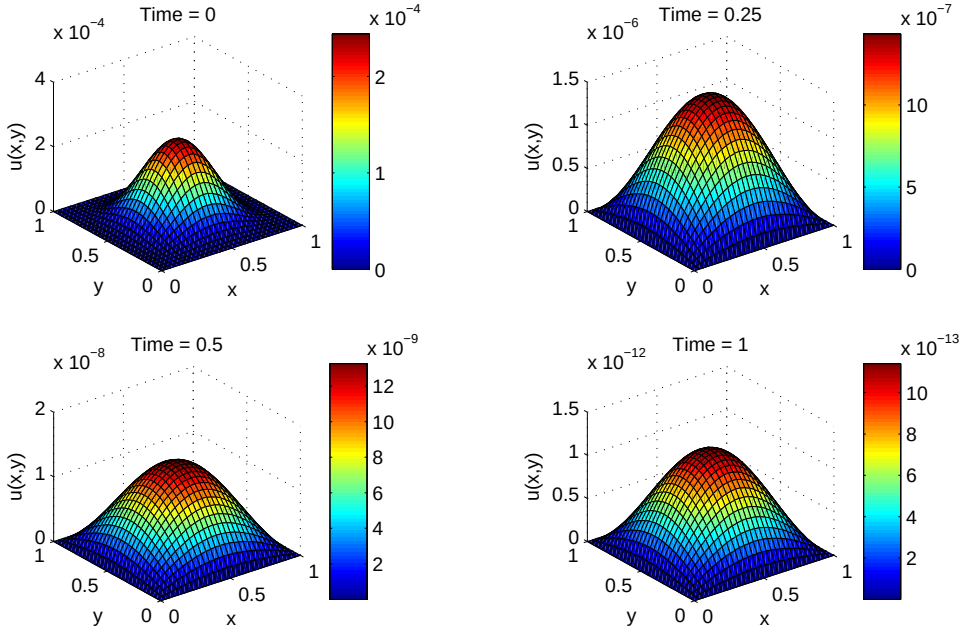

FIGURE 6.2: Surface plot of the numerical solution of example 6.1 for time $t \leq 1$ with $\Delta t = 0.00001$, $\gamma = 0.0001$ and $N = M = 31$

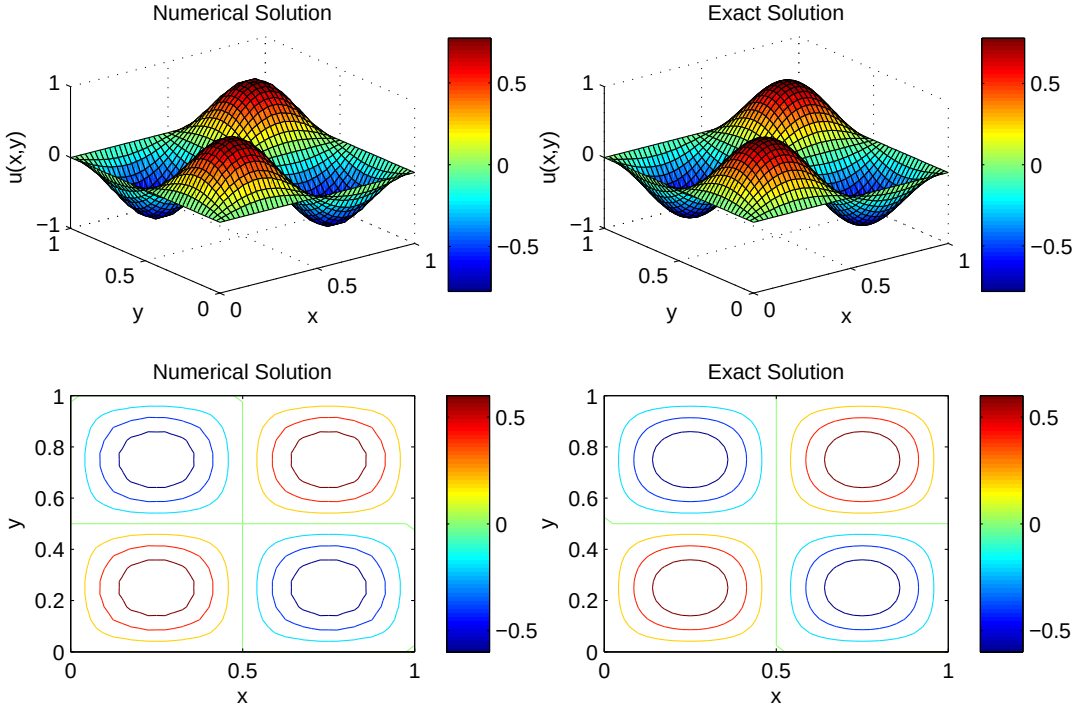

FIGURE 6.3: Surface and contour plot of the numerical and the exact solutions of example 6.2 at time $t = 0.25$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$

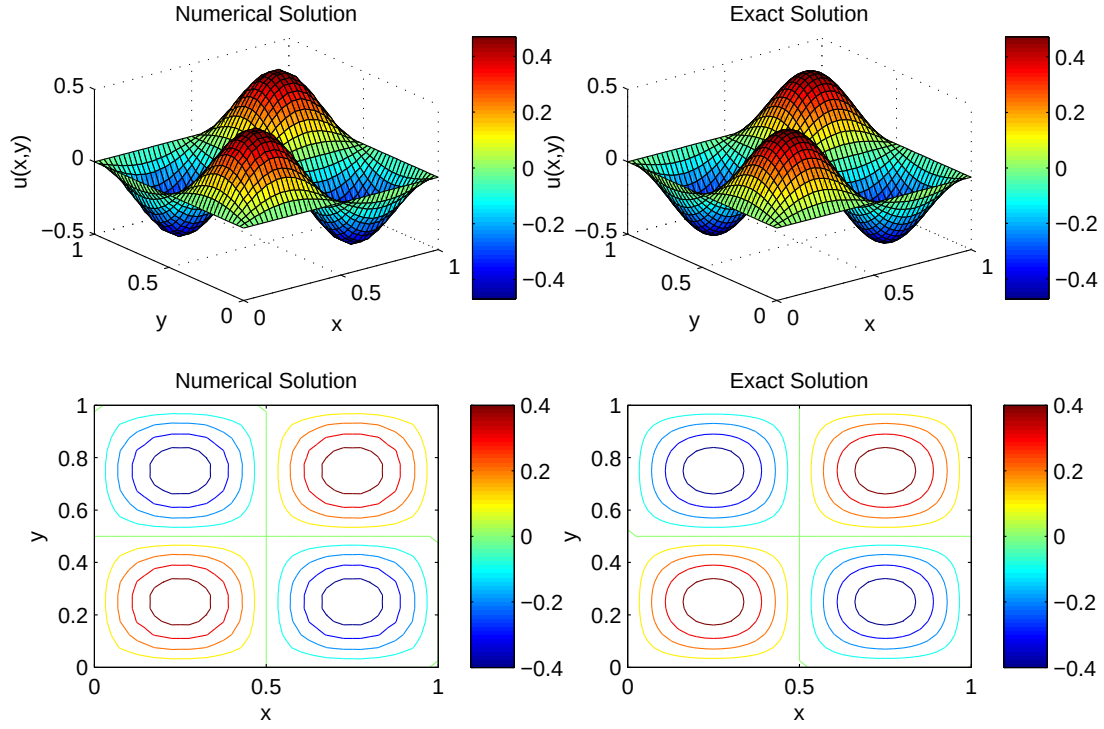

FIGURE 6.4: Surface and contour plot of the numerical and the exact solution of example 6.2 at time $t = 0.75$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$

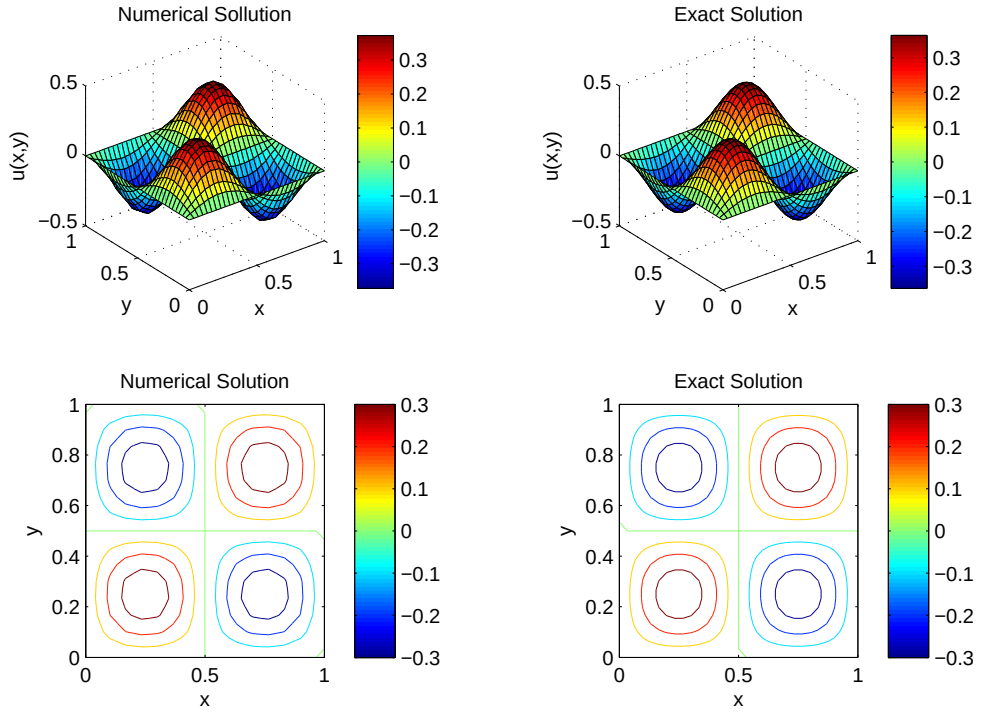

FIGURE 6.5: Surface and contour plot of the numerical and the exact solution of example 6.2 at time $t = 1$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$

TABLE 6.1: Error norms in the numerical solution computed by present method for example 6.2 at different time levels with $N = M = 41$, $\Delta t = 0.0001$ and $\gamma = 0.00001$

Time	L_∞	L_2	*Aver.Error*
0.25	1.60E-2	6.70E-3	4.64E-3
0.75	9.72E-3	4.07E-3	2.81E-3
1	7.58E-3	3.17E-3	2.19E-3
1.25	5.90E-3	2.47E-3	1.70E-3
1.5	4.60E-3	1.92E-3	1.33E-3

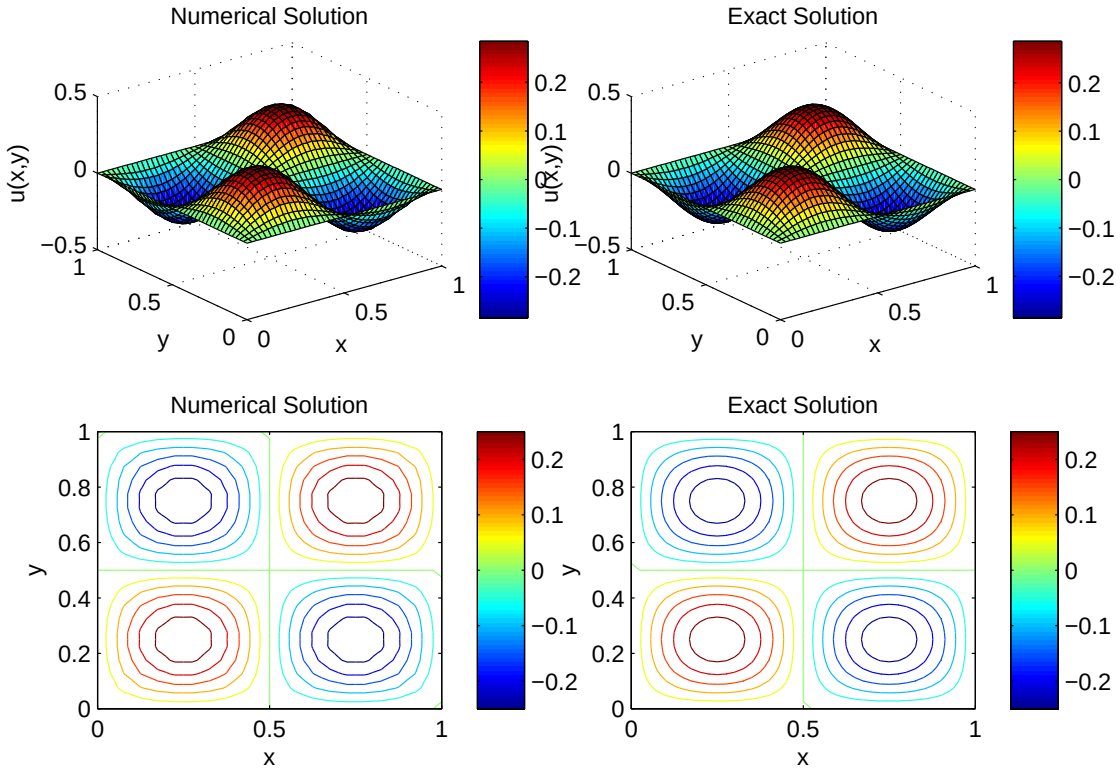

FIGURE 6.6: Surface and contour plot of the numerical and the exact solution of example 6.2 at time $t = 1.25$ with $\Delta t = 0.0001$, $\gamma = 0.001$ and $N = M = 41$

TABLE 6.2: Comparison of errors norms computed by present method for example 6.2 at time $t = 1$ with different values of N and M ($N = M = 30, 40, 50$)

N=M	L_∞	L_2	$Aver.Error$	L_∞[256]
30	9.62E-3	3.99E-3	2.69E-3	0.0016
40	7.58E-3	3.17E-3	2.19E-3	8.13E-4
50	6.23E-3	2.62E-3	1.84E-3	4.45E-4

Chapter 7

Conclusion

This thesis aims to explore the implementation of trigonometric B-spline basis function with collocation and differential quadrature methods. These two methods are used with third and fifth degree basis functions to compute the numerical solution of some important second, third and fourth order linear and nonlinear time dependent partial differential equations. The considered basis function approximate the given function in the referred range of interest and also provide a complete set of basis functions.

Trigonometric B-spline and standard B-spline basis functions have a few distinct advantages as both generate a piecewise-continuous and closed form of a numerical solution. But trigonometric B-spline basis functions has a few distinct advantages over standard B-spline basis functions such as C^{∞} continuity, more accuracy, and efficiency to approximate any function as shown in Table 3.6, 3.7, 3.14 to 3.18 in chapter three. Moreover, it shows better results with less computation timing as given in Table 3.3 in chapter 3, when implemented with collocation and differential quadrature methods as compared to standard B-spline basis function. This basis function also reduces the computational complexity in comparison to the finite element method as this is easy to implement with collocation and differential quadrature methods.

In collocation method, quasilinearization technique is used to resolve the nonlinearity in the differential equation while in differential quadrature method no such treatment is required to handle nonlinearity. While implementation of the method

the selection for the degree of trigonometric B-spline basis function depends on the order of the differential equation to be solved. For instance, in this work cubic trigonometric B-spline basis is used for second order equation and quintic trigonometric B-spline basis is used for third and fourth order differential equations.

The considered problems in the chapters arise in mathematical modeling of many physical, chemical and biological phenomena such as plasma instabilities, flame front propagation, phase turbulence in reaction diffusion system, pattern formation in bi-stable systems, in wave phenomena and in many areas related to engineering. The conclusion of the chapters are as follows:

In chapter two, Fisher's equation is considered for numerical solution by trigonometric cubic B-spline collocation method. Reason for the selection of this nonlinear equation is its numerous application in chemical and biological processes. The stability and convergence of the numerical scheme have been proved using theoretical results. The computed results are compared with the exact solutions by obtaining the absolute, L_2 and L_∞ errors. It is evident from the obtained errors that the numerical results are in good conformity with the exact solutions. It can be concluded that the method is an economical and efficient technique for finding numerical solutions for a variety of linear and nonlinear physical models.

Chapter three is dedicated to obtaining the numerical solution of second order time dependent linear and nonlinear partial differential equations in one dimension. To compute the numerical solution cubic trigonometric B-spline basis function is used with differential quadrature method. Four different equations have been considered in this chapter for obtaining their numerical solution. On implementing the scheme and substituting the derivatives, set of ODEs are attained, which is solved using SSP RK43 method. The efficiency and precision of the proposed method are revealed by many test problems. The comparison of obtained numerical results and different error norms with the numerical solutions from the literature are found to be in decent agreement with formerly obtained results. The advantage of the developed method includes it is easy to implement characteristic with reduced data complexity as compared to the old schemes present in the literature.

Chapter four is the extension of chapter three in context of space dimensions. All the four equations of second order solved in one dimension in the previous chapter, are considered again for the numerical solution in two dimensions in this chapter. Stability of the numerical method is given using the concept of eigenvalues. The efficiency and robustness of the method are shown using the error norms. The proposed scheme in this chapter is in good conformity with the exact solutions and found to be better than the numerical results present in the literature.

In chapter five, the numerical solution of a generalized fourth order partial differential equation is computed using proposed hybrid scheme in which differential quadrature method is implemented using quintic trigonometric B-spline basis function. The considered general equation reduces to three well-known equations namely: Korteweg-de Vries equation extended Fisher-Kolmogorov equation and Kuramoto-Sivashinsky equation. Numerical results demonstrate the good accuracy of the proposed method. Various numerical examples are taken for each considered equation. While implementing the proposed scheme, the generalized equation is converted to a system of ordinary differential equations which is further solved using the SSP-RK43 method. The L_2, L_∞ and RMS errors are calculated and presented to show the robustness of the method. Stability of the scheme is also evaluated with the help of eigenvalues and the scheme is found to be unconditionally stable. As a summary, this method proved to be a significant tool in solving the nonlinear partial differential equations of fourth order.

In chapter six, numerical solution of an extended Fisher-Kolmogorov equation in two dimensions is computed by quintic trigonometric differential quadrature method. Proposed method demonstrate good accuracy and the computed numerical results are observed to be empowering. Two test problems are taken from the literature for simulation of the extended Fisher-Kolmogorov equation. Different error norms are computed and presented to show the efficiency of the method. Stability of the method is also discussed for this equation.

While implementing the trigonometric B-spline basis functions with collocation and differential quadrature methods, following observations are made:

1. Collocation method is easy to apply in comparison to other numerical technique such as finite element methods.

2. Differential quadrature method is better and easy than collocation method as it reduces the computational complexity and needs less number of grid points when collocation fails to give good results.

3. While implementing both the methods with trigonometric B-spline basis functions obtained numerical results are in good coordination with the exact solution and found to be better than the numerical results in the literature.

4. In the context of stability for the considered equations, both the proposed methods satisfies the stability conditions and found to be unconditionally stable. The order of convergence of both the methods is around two, which is quite good.

5. While implementing the differential quadrature method nonlinearity of the differential equation requires no special treatment.

6. In trigonometric differential quadrature method partial differential equation is converted into a set of ordinary differential equations by substituting the derivatives as a linear combination of basis functions without any discretizing the time derivative, this results in the reduced computational effort.

For future work, following are the suggestions for consideration:

1. In this thesis only initial and boundary problems are considered for the numerical solutions, these methods can be tested for boundary value problems.

2. While using trigonometric collocation method nonlinear problems are linearized by using quasilinearization formula as discussed in the thesis. This work can be extended to deal with the nonlinearity without using quasilinearization.

3. In differential quadrature, the time derivative can be approximated by finite difference approximation that will result in a system of linear equations which can be further solved by using any numerical method.

4. Most of the problems solved in this research work are in one and two dimensions. These methods can be applied to solve higher dimensional equations appearing in various applications of science and engineering.

5. In this research work, collocation and differential quadrature methods are applied with trigonometric B-spline basis functions. This basis function can also be implemented with other numerical legacy methods.

Bibliography

[1] M. Abbas, A. A. Majid, A. I. M. Ismail, A. Rashid, *The application of cubic trigonometric B-spline to the numerical solution of the hyperbolic problems*, Appl. Math. Comput. **239**(2014), 74-88.

[2] O. Botella, *On a collocation B-spline method for the solution of the Navier-Stokes equations*, Comput. Fluids, **31**(2002), 397-420.

[3] A. Can, I. Dag, *Taylor-Galerkin and Taylor-collocation methods for the numerical solutions of Burgers' equation using B-splines*, Commun. Nonlinear Sci. Numer. Simul. **16**(2011), 2696-2708.

[4] R. I. Fernandes, G. Fairweather, *An ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems*, J. Comput. Phys. **231**(2012), no. 19, 6248-6267.

[5] N. Govindarajan, C. C. D. Visser, K. Krishnakumar, *A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines*, Automatica, **50**(2014), 2234-2244.

[6] B. Gupta, V. K. Kukreja, *Numerical approach for solving diffusion problems using cubic B-spline collocation method*, Appl. Math. Comput. **219**(2012), no. 4, 2087-2099.

[7] E. Ideon, P. Oja, *Linear / linear rational spline collocation for linear boundary value problems*, J. Comput. Appl. Math. **263**(2014), 32-44.

[8] S. Jator, Z. Sinkala, *A high order B-spline collocation method for linear boundary value problems*, Appl. Math. Comput. **191**(2007), 100-116.

[9] M. K. Kadalbajoo, D. Kumar, *Fitted mesh B-spline collocation method for singularly perturbed differential-difference equations with small delay*, Appl. Math. Comput. **204**(2008), 90-98.

[10] M. K. Kadalbajoo, A. S. Yadaw, D. Kumar, *Comparative study of singularly perturbed two-point BVPs via: Fitted-mesh finite difference method, B-spline collocation method and finite element method*, Appl. Math. Comput. **204**(2008), no. 2, 713-725.

[11] M. K. Kadalbajoo, P. Arora, *B-spline collocation method for the singular-perturbation problem using artificial viscosity*, Comput. Math. Appl. **57**(2009), no. 4, 650-663.

[12] M. K. Kadalbajoo, V. K. Aggarwal, *Fitted mesh B-spline collocation method for solving self-adjoint singularly perturbed boundary value problems*, Appl. Math. Comput. **161**(2005), 973-987.

[13] M. K. Kadalbajoo, V. Gupta, *Numerical solution of singularly perturbed convection-diffusion problem using parameter uniform B-spline collocation method*, J. Math. Anal. Appl. **355**(2009), no. 1, 439-452.

[14] M. K. Kadalbajoo, V. Gupta, A. Awasthi, *A uniformly convergent B-spline collocation method on a nonuniform mesh for singularly perturbed one-dimensional time-dependent linear convection-diffusion problem*, J. Comput. Appl. Math. **220**(2008), 271-289.

[15] M. K. Kadalbajoo, L. P. Tripathi, A. Kumar, *A cubic B-spline collocation method for a numerical solution of the generalized Black-Scholes equation*, Math. Comput. Model. **55**(2012), no. 34, 1483-1505.

[16] M. K. Kadalbajoo, A. S. Yadaw, *B-Spline collocation method for a two-parameter singularly perturbed convection-diffusion boundary value problems*, Appl. Math. Comput. **201**(2008), 504-513.

[17] M. Lakestani, M. Dehghan, *Numerical solutions of the generalized Kuramoto-Sivashinsky equation using B-spline functions*, Appl. Math. Model. **36**(2012), no. 2, 605-617.

[18] F. G. Lang, X. P. Xu, *Quintic B-spline collocation method for second order mixed boundary value problem*, Comput. Phys. Commun. **183**(2012), no. 4, 913-921.

[19] C. Li, T. Zhao, W. Deng, Y. Wu, *Orthogonal spline collocation methods for the sub diffusion equation*, J. Comput. Appl. Math. **255**(2014), 517-528.

[20] R. C. Mittal, G. Arora, *Quintic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation*, Commun. Nonlinear Sci. Numer. Simul. **15**(2010), no. 10, 2798-2808.

[21] R. C. Mittal, R. Bhatia, *Numerical solution of second order one dimensional hyperbolic telegraph equation by cubic B-spline collocation method*, Appl. Math. Comput. **220**(2013), 496-506.

[22] R. C. Mittal, R. K. Jain, *Redefined cubic B-splines collocation method for solving convection-diffusion equations*, Appl. Math. Model. **36**(2012), no. 11, 5555-5573.

[23] R. Mohammadi, *Sextic B-spline collocation method for solving Euler-Bernoulli Beam Models*, Appl. Math. Comput. **241**(2014), 151-166.

[24] Y. Morinishi, S. Tamano, K. Nakabayashi, *A DNS algorithm using B-spline collocation method for compressible turbulent channel flow*, Comput. Fluids, **32**(2003), 751-776.

[25] S. C. S. Rao, M. Kumar, *Optimal B-spline collocation method for self-adjoint singularly perturbed boundary value problems*, Appl. Math. Comput. **188**(2007), no. 1, 749-761.

[26] S. S. Siddiqi, S. Arshed, *Quintic B-spline for the numerical solution of the good Boussinesq equation*, J. Egypt. Math. Soc. **22**(2013), no. 2, 209-213.

[27] R. C. Mittal, R. K. Jain, *Cubic B-splines collocation method for solving nonlinear parabolic partial differential equations with Neumann boundary conditions*, Commun. Nonlinear Sci. Numer. Simul. **17**(2012), 4616-4625.

[28] X. Li, *Numerical solution of fractional differential equations using cubic B-spline wavelet collocation method*, Commun. Nonlinear Sci. Numer. Simul. **17**(2012), no. 10, 3934-3946.

[29] X. Yang, H. Zhang, D. Xu, *Orthogonal spline collocation method for the two-dimensional fractional sub-diffusion equation*, J. Comput. Phys. **256**(2014), 824-837.

[30] H. Zhang, X. Han, X. Yang, *Quintic B-spline collocation method for fourth order partial integro-differential equations with a weakly singular kernel*, Appl. Math. Comput. **219**(2013), no. 12, 6565-6575.

[31] P. K. Sahu, S. Saha Ray, *Numerical solutions for the system of Fredholm integral equations of second kind by a new approach involving semiorthogonal B-spline wavelet collocation method*, Appl. Math. Comput. **234**(2014), 368-379.

[32] A. Pedas, E. Tamme, *Numerical solution of nonlinear fractional differential equations by spline collocation methods*, J. Comput. Appl. Math. **255**(2014), 216-230.

[33] R. E. Bellman, J. Casti, *Differential quadrature and long-term integration*, J. Math. Anal. Appl. **34**(1971), 235-238.

[34] R. E. Bellman, B. G. Kashef, J. Casti, *Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations*, J. Comput. Phys. **10**(1972), 40-52.

[35] C. W. Bert, S. K. Jang, A. G. Striz, *Two new approximate methods for analyzing free vibration of structural components*, AIAA J. **26**(1988), 612-618.

[36] C. W. Bert, M. Malik, *Differential quadrature in computational mechanics: A review*, Appl. Mech. Rev. **49**(1996), no. 1, 1-27.

[37] C. W. Bert, X. Wang, A. G. Striz, *Differential quadrature for static and free vibration analyses of anisotropic plates*, Int. J. Solids Struct. **30**(1993), no. 13, 1737-1744.

[38] Y. Wang, *Differential quadrature method and differential quadrature element method-theory and applications*, Ph.D. Dissertation, Nanjing University of Aeronautics and Astronautics, China, 2001.

[39] C. Shu, *Differential Quadrature and Its Application in Engineering*, Springer-Verlag London Ltd.(2000).

[40] J. R. Quan, C. T. Chang, *New insights in solving distributed system equations by the quadrature methods-I*, Comput. Chem. Eng. **13**(1989),779-788.

[41] J. R. Quan, C. T. Chang, *New insights in solving distributed system equations by the quadrature methods-II*, Comput. Chem. Eng. **13**(1989), 1017-1024.

[42] A. Korkmaz, *İ. Dağ*, *Shock wave simulations using sinc differential quadrature method*, Eng. Comput. Int. J. Comput. Aided Eng. Software, **28**(2011), no. 6, 654-674.

[43] A. Korkmaz, *İ. Dağ*, *Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation*, J. Franklin Inst. **348** (2011), no. 10, 2863-2875.

[44] A. Korkmaz, *İ. Dağ*, *Cubic B-spline differential quadrature methods and stability for Burgers' equation*, Eng. Comput. Int. J. Comput. Aided Eng. Software, **30**(2013), no. 3, 320-344.

[45] A. Korkmaz, A.M. Aksoy, *İ. Dağ*, *Quartic B-spline differential quadrature method*, Int. J. Nonlinear Sci. **11**(2011), no. 4, 403-411.

[46] R. Bellman, B. Kashef, R. Vasudevan, *The inverse problem of estimating heart parameters from cardiograms*, Math. Biosci. **19**(1974), 221-230.

[47] N. Bellomo, *Nonlinear models and problems in applied sciences from differential quadrature to generalized collocation methods*, Math. Comput. Model. **26**(1980), 13-34.

[48] W. Chen, C. Shu, W. He, T. Zhong, *The DQ solution of geometrically nonlinear bending of orthotropic rectangular plates by using Hadamard and SJT product*, Comput. Structures, **74**(2000), no. 1, 65-74.

[49] W. Chen, Y. Yong, X. Wang, *Reducing the computational effort of the differential quadrature method*, Numer. Methods Partial Differ. Equ. **12**(1996), 565-577.

[50] W. Chen, T. X. Zhong, C. Shu, *A Lyapunov formulation for efficient solution of the Poisson and convection-diffusion equations by the differential quadrature method*, J. Comput. Phys. **141**(1998), no. 1, 78-84.

[51] F. Civan, *Solving multivariable models by the quadrature and cubature methods*, Numer. Methods Partial Differ. Equ. **10**(1994), 545-567.

[52] M. D. Nam, T. C. Thanh, *Numerical solution of differential equation using multiquadric radial basis function networks*, Neural Networks **14**(2001), 185-199.

[53] M. D. Nam, T. C. Thanh, *Approximation of function and its derivatives using radial basis function networks*, Appl. Math. Model. **27**(2003), 197-220.

[54] C. Shu, B. E. Richards, *Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations*, Internat. J. Numer. Methods Fluids, **15**(1992), 791-798.

[55] C. Shu, Y. T. Chew, *Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems*, Commun. Numer. Methods Eng. **13**(1997), no. 8, 643-653.

[56] C. Shu, H. Xue, *Explicit computation of weighting coefficients in the harmonic differential quadrature*, J. Sound Vib. **204**(1997), no. 3, 549-555.

[57] M. Tanaka, W. Chen, *Dual reciprocity BEM applied to transient elastodynamic problems with differential quadrature method in time*, Comput. Methods Appl. Mech. Engrg. **190**(2001), 2331-2347.

[58] X. Wu, L. Chen, *Solving the driven flow by linearized elimination with differential quadrature*, Numer. Methods Partial. Differ. Equ. **22**(2006), 540-577.

[59] X. Wu, S. Liu, *Differential quadrature domain decomposition method for problems on a triangular domain*, Numer. Methods Partial. Differ. Equ. **21**,(2005), 574-585.

[60] Y. Wu, X. Wu, *Linearized and rational approximation method for solving nonlinear Burgers' equation*, Internat. J. Numer. Meth. Fluids, **45**(2004), 509-525.

[61] R. C. Mittal, S. Dahiya, *Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method*, Appl. Math. Comput. **313**(2017), 442-452

[62] X. Wang, *Novel differential quadrature element method for vibration analysis of hybrid nonlocal Euler-Bernoulli beams*, Appl. Math. Lett. **77**(2018), 94-100.

[63] S. R. K. Iyengar, P. Jain, *Spline difference methods for singular two point boundary valueproblems*, Numer. Math. **500** (1987), 363-376.

[64] S. A. Khuri, A. Sayfy, *A numerical approach for solving an extended Fisher-Kolomogrov-Petrovskii-Piskunov equation*, J. Comput. Appl. Math. **233**(2010), 2081-2089.

[65] A. S. V. R. Kanth, Y. N. Reddy, *Cubic spline for a class of singular two-point boundary value problems*, Appl. Math. Comput. **170**(2005), 733-740.

[66] S. S. Siddiqi, E. H. Twizell, *Spline solution of linear eighth-order boundary value problems*, Comput. Methods Appl. Mech. Eng. **131**(1996), 309-325.

[67] G. Akram, S. S. Siddiqi, *Nonic spline solutions of eighth order boundary value problems*, Appl. Math. Comput. **182**(2006), 829-845.

[68] I. J. Schoenberg, *Contribution to the problem of approximation of equidistant data by analytical functions*, Quart. Appl. Math. **4**(1946), 45-99.

[69] I. J. Schoenberg, *On trigonometric spline interpolation*, J. Math. Mech. **13**(1964), 795-825.

[70] G. Arora, B. K. Singh, *Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method*, Appl. Math. Comput. **224**(2013), 166-177.

[71] X. Wang, *Differential quadrature and differential quadrature based element methods*, Butterworth-Heinemann publications, (2015).

[72] J. R. Spiteri, S. J. Ruuth, *A new class of optimal high-order strongstability-preserving time-stepping schemes*, SIAM J. Numer. Anal. **40**(2002), no. 2, 469-491.

[73] M. K. Jain, *Numerical Solution of Differential Equations*, second ed., Wiley, New York, NY, 1983.

[74] R. A. Fisher, *The wave of advance of advantageous genes*, Ann. Eugen. **7**(1937), 355-369.

[75] J. Canosa, *On a nonlinear diffusion equation describing population growth*, IBM J. Res. Develop. **17**(1973), 307-313.

[76] S. K. Aggarwal, *Some numerical experiments on Fisher's equation*, Int. Commun. Heat Mass Trans. **12**(1985), 417-430.

[77] D. G. Aronson, H. F. Weinberger, *Nonlinear diffusion in population genetics combustion and nerve pulse propagation, Partial Differential Equations and Related Topics*, Lecture Notes in Mathematics, **446**(1975), 5-49.

[78] P. K. Maini, D. L. S. Mcelwain, D. Leavesley, *Travelling waves in a wound healing assay*, Appl. Math. Lett. **17**(2004), 575-580.

[79] B. G. Sengers, C. P. Please, R. O. C. Oreffo, *Experimental characterization and computational modelling of two-dimensional cell spreading for skeletal regeneration*, J. R. Soc. Interface, **4**(2007), 1107-1117.

[80] J. Gazdag, J. Canosa, *Numerical solution of Fisher's equation*, J. Appl. Probab. **11**(1974), 445-457.

[81] D. Olmos, B. D. Shizgal, *A pseudospectral method of solution of Fisher's equation*, J. Comput. Appl. Math. **193**(2006), 219-242.

[82] Y. Qiu, D. M. Sloan, *Numerical solution of Fisher's equation using a moving mesh method*, J. Comput. Phys. **146**(1998), 726-746.

[83] A. M. Wazwaz, A. Gorguis, *An analytic study of Fisher's equation by using Adomian decomposition method*, Appl. Math. Comput. **154**(2004), 609-620.

[84] N. Parekh, S. Puri, *A new numerical scheme for the Fisher's equation*, J. Phys. A. **23**(1990), L1085-L1091.

[85] E. H. Twizell, Y. Wang, W. G. Price, *Chaos free numerical solutions of reaction-diffusion equations*, Proc. R. Soc. London Sci. A. **430**(1990), 541-576.

[86] R. C. Mittal, R. Jiwari, *Numerical study of Fisher's equation by using differential quadrature method*, Int. J. Inf. Syst. Sci. **5**(2009), 143-160.

[87] S. Tang, R. O. Weber, *Numerical study of Fisher's equation by a Petrov-Galerkin finite element method*, J. Aus. Math. Soc. Sci. B **33**(1991), 27-38.

[88] K. Al-Khaled, *Numerical study of Fisher's reaction-diffusion equation by the sinc collocation method*, J. Comput. Appl. Math. **137**(2001), 245-255.

[89] H. A. Abdusalam, E. S. Fahmy, *Exact solution for the generalized Telegraph Fisher's equation*, Chaos Solitons Fract. **41**(2009), 1550-1556.

[90] R. E. Mickens, *A best finite-difference scheme for Fisher's equation*, Numer. Meth. Part. Differ. Eqns. **10**(1994), 581-585.

[91] R. C. Mittal, S. Kumar, *Numerical study of Fisher's equation by wavelet Galerkin method*, Int. J. Comput. Math. **83**(2006), 287-298.

[92] M. S. El-Azab, *An approximation scheme for a nonlinear diffusion Fisher's equation*, Appl. Math. Comput. **186**(2007), 579-588.

[93] A. Sahin, I. Dag, B. Saka, *A B-spline algorithm for the numerical solution of Fisher's equation*, Kybernetes, **37**(2008), 326-342.

[94] R. C. Mittal, G. Arora, *Efficient numerical solution of Fisher's equation by using B-spline method*, Int. J. Comput. Math. **87**(2010), no. 13, 3039-3051.

[95] G. Hariharan, K. Kannan, K. R. Sharma, *Haar wavelet method for solving Fisher's equation*, Appl. Math. Comput. **211**(2009), 284-292.

[96] M. Aghamohamadi, J. Rashidinia, R. Ezzati, *Tension spline method for solution of non-linear Fisher's equation*, Appl. Math. Comput. **249**(2014), 399-407.

[97] M. Alquran, K. Al-Khaled, T. Sardar, J. Chattopadhyay, *Revisited Fisher's equation in a new outlook: A fractional derivative approach*, Physica A **438**(2015), 81-93.

[98] G. Faye, M. Holzer, *Modulated traveling fronts for a nonlocal Fisher's-KPP equation: A dynamical systems approach*, J. Differ. Equ. **258**(2015), 2257-2289.

[99] J. Stoer, R. Bulrisch, *An introduction to numerical analysis*, Springer-verlag, 1991.

[100] X. Y. Wang, *Exact and explicit solitary wave solutions for the generalized Fisher's equation*, Phys. Lett. A, **131**(1988), no. 4, 277-279.

[101] R. K. Dodd, J. C. Eilbeck, J. D. Gibbon, H. C. Morris, *Solitons and nonlinear wave equations*, Academic Press, London, 1982, ISBN 978-0-12-219122-0.

[102] D. Kaya, *An application of the modified decomposition method for two dimensional sine-Gordon equation*, Appl. Math. Comput. **159**(2004), 1-9.

[103] S. S. Ray, *A numerical solution of the coupled sine-Gordon equation using the modified decomposition method*, Appl. Math. Comput. **175**(2006), 1046-1054.

[104] Q. Wang, *An application of the modified Adomian decomposition method for (N+1)-dimensional sine-Gordon field*, Appl. Math. Comput. **181**(2006), 147-152.

[105] U. Yucel, *Homotopy analysis method for the sine-Gordon equation with initial conditions*, Appl. Math. Comput. **203**(2008), 387-395.

[106] A. Sadighi, D. D. Ganji, *Traveling wave solutions of the sine-Gordon and the coupled sine-Gordon equations using the homotopy-perturbation method*, Scientia Iranica Transaction B Mechanical Engineering, **16**(2009), 189-195.

[107] N. H. Kuo, C. D. Hu, *A study of the solutions of the combined sine-cosine-Gordon equation*, Appl. Math. Comput. **215**(2009), 1015-1019.

[108] M. Dehghan, D. Mirzaei, *The dual reciprocity boundary element method (DRBEM) for two-dimensional Sine-Gordon equation*, Comput. Methods Appl. Mech. Engrg. **197**(2008), 476-486.

[109] M. Dehghan, A. Shokri, *A numerical method for one dimensional nonlinear sine-Gordon equation using collocation and radial basis functions*, Numer. Methods Partial Differ. Equs. **24**(2008), no. 2, 687-698.

[110] M. Dehghan, A. Shokri, *A numerical method for solution of the two dimensional sine-Gordon equation using the radial basis functions*, Math. Comput. Simulat. **79**(2008), 700-715.

[111] R. Jiwari, S. Pandit, R. C. Mittal, *Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method*, Comput. Phy. Commun. **183**(2012), 600-616.

[112] R. C. Mittal, R. Bhatia, *Numerical Solution of nonlinear sine-Gordon equation by modified cubic B-spline collocation method*, Internat. J. Partial Differ. Equs. (2014), doi:10.1155/2014/343497.

[113] F. Yin, T. Tian, J. Song, M. Zhu, *Spectral methods using Legendre wavelets for nonlinear Klein/Sine-Gordon equations*, J. Comput. Appl. Math. **275**(2015), 321-334.

[114] M. Li-Min, W. Zong-Min, *A numerical method for one dimensional nonlinear sine-Gordon equation using multiquadric quasi-interpolation*, Chinese Physics B, **18**(2009), no. 8, 3099-3103.

[115] J. M. Burgers, *A mathematical model illustrating the theory of turbulence*, Adv. Appl. Mech. **1**(1948), 171-199.

[116] E. Benton, G. W. Platzman, *A table of solutions of the one dimensional Burgers' equations*, Q. Appl. Math. **30**(1972) 195-212.

[117] M. M. Cecchi, R. Nociforo, P. P. Grego, *Space-time finite elements numerical solution of Burgers' problems*, Le Matematiche LI (Fasc. I) (1996), 43-57.

[118] T. Ozis, E. N. Aksan, A. Ozdes, *A finite element approach for solution of Burgers' equation*, Appl. Math. Comput. **139**(2003), 417-428.

[119] A. Dogan, A. Galerkin, *Finite element approach to Burgers' equation*, Appl. Math. Comput. **157**(2004), 331-346.

[120] I. Dag, D. Irk, A. Sahin, *B-Spline collocation methods for numerical solutions of the Burgers' equation*, Math. Probl. Eng. **5**(2005), 521-538.

[121] A. Asaithambi, *Numerical solution of the Burgers' equation by automatic differentiation*, Appl. Math. Comput. **216**(2010), 2700-2708.

[122] R. C. Mittal, R. K. Jain, *Numerical solutions of nonlinear Burgers' equation with modied cubic B-splines collocation method*, Appl. Math. Comput. **218**(2012), 7839-7855.

[123] A. K. Khalifa, K. I. Noor, M. A. Noor, *Some numerical methods for solving Burgers' equation*, Int. J. Phys. Sci. **6**(2011), no. 7, 1702-1710.

[124] Z. Jiang, R. Wang, *An improved numerical solution of Burgers' equation by cubic B-spline Quasi-interpolation*, J. Inf. Comput. Sci. **7**(2010), no. 5, 1013-1021.

[125] B. Saka, *İ. Dağ*, *Quartic B-spline collocation method to the numerical solutions of the Burgers' equation*, Chaos Solitons Fract. **32**(2007), 1125-1137.

[126] E. N. Aksan, A. Ozdes, *A numerical solution of Burgers' equation*, Appl. Math. Comput. **156**(2004), 395-402.

[127] T. *Özis*, A. Esen, S. Kutluay, *Numerical solution of Burgers' equation by quadratic B-spline finite elements*, Appl. Math. Comput. **165**(2005), 237-249.

[128] I. A. Hassanien, A. A. Salama, H. A. Hosham, *Fourth-order finite difference method for solving Burgers' equation*, Appl. Math. Comput. **170**(2005), 781-800.

[129] K. Altparmak, *Numerical solution of Burgers' equation with factorized diagonal Pad approximation*, Int. J. Numer. Methods Heat Fluid Flow, **21**(2011), no. 3, 310-319.

[130] M. A. Ramadan, T. S. El-Danaf, F. E. I. Abd Alaal, *Application of the non-polynomial spline approach to the solution of the Burgers' equation*, Open Appl. Math. J. **1**(2007), 15-20.

[131] M. Xu, R. H. Wang, J. H. Zhang, Q. Fang, *A novel numerical scheme for solving Burgers' equation*, Appl. Math. Comput. **217**(2011), 4473-4482.

[132] S. Kutluay, A. R. Bahadir, A. Ozdes, *Numerical solution of one-dimensional Burgers' equation: explicit and exact-explicit finite difference methods*, J. Comput. Appl. Math. **103**(1999), 251-261.

[133] J. D. Cole, *On a quasi-linear parabolic equations occurring in aerodynamics*, Quart. Appl. Math. **9**(1951), 225-236.

[134] S. Kutulay, A. Esen, *İ. Dağ*, *Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method*, J. Comput. Appl. Math. **167**(2004), 21-33.

[135] S. Kumar, P. Singh, *Higher-order MUSCL scheme for transport equation originating in a neuronal model*, Comput. Appl. Math. **70**(2015), 2838-2853.

[136] S. Xie, S. Heo, S. Kim, G. Woo, S. Yi, *Numerical solution of one-dimensional Burgers' equation using reproducing kernel function*, J. Comput. Appl. Math. **214**(2008), 417-434.

[137] W. Liao, *An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation*, Appl. Math. Comput. **206**(2008), 755-764.

[138] R. Jiwari, R. C. Mittal, K. K. Sharma, *A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation*, Appl. Math. Comput. **219**(2013), 6680-6691.

[139] M. K. Kadalbajoo, K. K. Sharma, A. Awasthi, *A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equation*, Appl. Math. Comput. **170**(2005), 1365-1393.

[140] R. C. Mittal, P. Singhal, *Numerical solution of Burgers' equation*, Commun. Numer. Methods Eng. **9**(1993), 397-406.

[141] B. Ay, I. Dag, M. Z. Gorgulu, *Trigonometric quadratic B-spline subdomain Galerkin algorithm for the Burgers' equation*, Open Physics, **13**(2015), no. 1, 400-406.

[142] A. Esen, O. Tasbozan, *Numerical solution of time fractional Burgers' equation by cubic B-spline finite elements*, Mediterr. J. Math. **13**(2015), no. 3, 1325-1337.

[143] A. Esen, O. Tasbozan, *Numerical solution of time fractional Burgers' equation*, Acta Universitatis Sapientiae, Mathematica, **7**(2015), no. 2, 167-185.

[144] S. Kutluay, Y. Ucar , N. M. Yagmurlu, *Numerical solutions of the modified Burgers' equation by a cubic B-spline collocation method*, Bull. Malays. Math. Sci. Soc., **39**(2016), 1603-1614.

[145] S. Kutluay, N. M. Yagmurlu, *The modified bi-quintic B-splines for solving the two-dimensional unsteady Burgers' equation*, Europ. Internat. J. Sci. Tech. **1**(2012), no. 2, 23-39.

[146] K. R. Raslan, *A collocation solution for Burgers' equation using quadratic B-spline finite elements*, Int. J. Comput. Math. **80**(2003), no. 7, 931-938.

[147] J. Chung, E. Kim, Y. Kim, *Asymptotic agreement of moments and higher order contraction in the Burgers' equation*, J. Diff. Equ. **248**(2010), no. 10, 2417-2434.

[148] A. H. Salas, *Symbolic computation of solutions for a forced Burgers' equation*, Appl. Math. Comput. **216**(2010), no. 1, 18-26.

[149] L. Zhang, J. Ouyang, X. Wang, X. Zhang, *Variational multiscale element-free Galerkin method for 2D Burgers' equation*, J. Comput. Phy. **229**(2010), no. 19, 7147-7161.

[150] S. Lin, C. Wang, Z. Dai, *New exact traveling and non-traveling wave solutions for (2+1) dimensional Burgers' equation*, Appl. Math. Comput. **216**(2010), no. 10, 3105-3110.

[151] A. Korkmaz, *Numerical solutions of some nonlinear partial differential equations using differential quadrature method*, Thesis of Master Degree, Eskisehir Osmangazi University, 2006.

[152] A. Korkmaz, *Numerical solutions of some one dimensional partial differential equations using B-spline differential quadrature method*, Doctoral Dissertation, Eskisehir Osmangazi University, 2010.

[153] M. Tamsir, V. K. Srivastava, R. Jiwari, *An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation*, Appl. Math. Comput. **290**(2016), 111-124.

[154] A. Arnold, *Numerically absorbing boundary conditions for quantum evolution equations*, VLSI Design, **6**(1998), 313-319.

[155] M. Lévy, *Parabolic equation methods for electromagnetic wave propagations*, IEEE, 2000.

[156] F. D. Tappert, *The parabolic approximation method in wave propagation and underwater acoustics(edited by J. B. Keller and J. S. Papadakis)*, Topics in Current Physics (Springer-Verlag, Berlin), 1977.

[157] Y. V. Kopylov, A. V. Popov, A. V. Vinogradov, *Applications of the parabolic wave equations to X-ray diffraction optics*, Optics Comm. **118**(1995), 619-636.

[158] W. Huang, C. Xu, S. T. Chu, S. K. Chaudhuri, *The finite-difference vector beam propagation method*, J. Lightwave Technol. **10**(1992), no. 3, 295-304.

[159] F. Y. Hajj, *Solution of the Schrdinger equation in two and three dimensions*, J. Phys. B **18**(1985), 1-11.

[160] L. Gr. Ixaru, *Operations on oscillatory functions*, Comput. Phys. Comm. **105**(1997), 1-9.

[161] M. Dehghan, *Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices*, Math. Comput. Simul. **71**(2006), 16-30.

[162] M. Subasi, *On the finite-difference schemes for the numerical solution of two dimensional Schrodinger equation*, Numer. Methods Partial Differ. Equ. **18** (2002), 752-758.

[163] M. Dehghan, A. Shokri, *A numerical method for two-dimensional Schrodinger equation using collocation and radial basis functions*, Comput. Math. Appl. **54**(2007), 136-146.

[164] J. C. Kalita, P. Chhabra, S. Kumar, *A semi-discrete higher order compact scheme for the unsteady two-dimensional Schrodinger equation*, J. Comput. Appl. Math. **197**(2006), 141-149.

[165] J. A. C. Weideman, B. M. Herbst, *Split-step methods for the solution of the nonlinear Schrodinger equation*, SIAM J. Numer. Anal. **23**(1986), no. 3, 485-507.

[166] P. L. Sulem, C. Sulem, A. Patera, *Numerical simulation of singular solutions to the two dimensional cubic Schrodinger equation*, Commun. Pure Appl. Math. **37**(1984), 755-778.

[167] T. R. Taha, *A numerical scheme for the nonlinear Schrodinger equation*, Comput. Math. Appl. **22**(1991), no. 9, 77-84.

[168] Q. Chang, E. Jia, W. Sun, *Difference schemes for solving the generalized nonlinear Schrodinger equation*, J. Comput. Phys. **148**(1999), 397-415.

[169] L. R. T. Gardner, G. A. Gardner, S. I. Zaki, Z. El Sahrawi, *B-spline finite element studies of the non-linear Schrodinger equation*, Comput. Methods Appl. Mech. Eng. **108**(1993), 303-318.

[170] P. Muruganandam, S. K Adhikari, *Bose-Einstein condensation dynamics in three dimensions by the pseudospectral and finite-difference methods*, J. Phys. B, **36**(2003), 2501-2513.

[171] V. M. Perez-Garcia, X. Y. Liu, *Numerical methods for the simulation of trapped nonlinear Schrodinger systems*, Appl. Math. Comput. **144**(2003), no. 23, 215-235.

[172] W. Bao, D. Jaksch, P. A. Markowich, *Numerical solution of the Gross-Pitaevskii equation for Bose-Einstein condensation*, J. Comput. Phys. **187**(2003), no. 1, 318-342.

[173] M. M. Cerimele, M. L. Chiofalo, F. Pistella, S. Succi, M. P. Tosi, *Numerical solution of the Gross-Pitaevskii equation using an explicit finite-difference scheme: an application to trapped Bose-Einstein condensates*, Phys. Rev. E. **62**(2000), 1382-1389.

[174] C. M. Dion, E. Cances, *Spectral method for the time-dependent Gross-Pitaevskii equation with a harmonic trap*, Phys.Rev.E, **67**(2003), 046706-1-046706-9.

[175] M. Javidi, A. Golbabai, *Numerical studies on nonlinear Schrodinger equations by spectral collocation method with preconditioning*, J. Math. Anal. Appl. **333**(2007), 1119-1127.

[176] H. Wang, *Numerical studies on the split-step finite difference method for nonlinear Schrodinger equations*, Appl. Math. Comput. **170**(2005), 17-35.

[177] A. Mohebbi, M. Dehghan, *The use of compact boundary value method for the solution of two-dimensional Schrodinger equation*, J. Comput. Appl. Math. **225**(2009), 124-134.

[178] R. Abdur, A. I. B. M. Ismail, *Numerical studies on two dimensional schrodinger equation by chebyshev spectral collocation method*, U.P.B. Sci. Bull., Series A, **73**(2011), no. 1, 101-110.

[179] R. K. Mohanty, *New unconditionally stable difference schemes for the solution of multi-dimensional telegraphic equations*, Int. J. Comput. Math. **86**(2009), no. 12, 2061-2071.

[180] P. M. Jordan, A. Puri, *Digital signal propagation in dispersive media*, J. Appl. Phys. **85**(1999), no. 3, 1273-1282.

[181] V. H. Weston, S. He, *Wave splitting of the telegraph equation in R^3 and its application to inverse scattering*, Inverse Probl. **9**(1993), 789-812.

[182] J. Banasiak, J. R. Mika, *Singularly perturved telegraph equations with applications in the random walk theory*, J. Appl. Math. Stoch. Anal. **11**(1998), no. 1, 9-28.

[183] A. C. Metaxas, R. J. Meredith, *Industrial microwave*, Heating. Peter Peregrinus, London, 1993.

[184] G. Roussy, J. A. Pearcy, *Foundations and industrial applications of microwaves and radio frequency fields*, Wiley, New York, 1995.

[185] M. S. El-Azab, El-Ghamel, *A numerical algorithm for the solution of telegraph equations*, Appl. Math. Comput. **190**(2007), 757-764.

[186] W. S. Kim, *Doubly-periodic boundary value problem for nonlinear dissipative hyperbolic equations*, J. Math. Anal. Appl. **145**(1990), 1-6.

[187] R. K. Mohanty, *An unconditionally stable difference scheme for the one space-dimensional linear hyperbolic equation*, Appl. Math. Lett. **17**(2004), no. 1, 101-105.

[188] R. K. Mohanty, *An unconditionally stable finite difference formula for a linear second order one space dimensional hyperbolic equation with variable coefficients*, Appl. Math. Comput. **165**(2005), 229-236.

[189] H. W. Liu, L. B. Liu, *An unconditionally stable spline difference scheme of $O(k^2 + h^4)$ for solving the second order $1D$ linear hyperbolic equation*, Math. Comput. Model. **49**(2009), 1985-1993.

[190] M. Dehghan, *On the solution of an initial-boundary value problem that combines Neumann and integral condition for the wave equation.* Numer. Methods Partial Differ. Eqns. **21**(2005), 24-40.

[191] R. K. Mohanty, M. K. Jain, K. George, *On the use of high order difference methods for the system of one space second order non-linear hyperbolic equations with variable coefficients*, J. Comput. Appl. Math. **72**(1996), 421-431.

[192] E. H. Twizell, *An explicit difference method for the wave equation with extended stability range*, BIT Numer. Math. **19**(1979), no. 3, 378-383.

[193] A. Mohebbi, M. Dehghan, *High order compact solution of the one-space-dimensional linear hyperbolic equation*, Numer. Methods Partial Differ. Eqns. **24**(2008), 1222-1235.

[194] M. Dehghan, M. Lakestani, *The use of Chebyshev cardinal functions for solution of the second-order one-dimensional telegraph equation*, Numer. Methods Partial Differ. Eqns. **25**(2009), 931-938.

[195] M. Lakestani, B. N. Saray, *Numerical solution of telegraph equation using interpolating scaling functions*, Comput. Math. Appl. **60**(2010), no. 7, 1964-1972.

[196] A. Saadatmandi, M. Dehghan, *Numerical solution of hyperbolic telegraph equation using the Chebyshev Tau method*, Numer. Methods Partial Differ. Eqns. **26**(2010), no. 1, 239-252.

[197] F. Gao, C. M. Chi, *Unconditionally stable difference schemes for a one space-dimensional linear hyperbolic equation*, Appl. Math. Comput. **187**(2007), no. 2, 1272-1276.

[198] M. Dehghan, A. Ghesmati, *Solution of the second-order one-dimensional hyperbolic telegraph equation by using the dual reciprocity boundary integral equation (DRBIE) method*, Engg. Anal. Boundary Elem. **34**(2010), 51-59.

[199] M. Dehghan, A. Shokri, *A numerical method for solving the hyperbolic telegraph equation*, Numer. Methods Partial Differ. Eqns. **24**(2008), 1080-1093.

[200] R. Jiwari, S. Pandit, R. C. Mittal, *A differential quadrature algorithm for the numerical solution of the second-order one dimensional hyperbolic telegraph equation*, Int. J. Nonlinear Sci. **13**(2012), no. 3, 259-266.

[201] M. Dosti, A. Nazemi, *Solving one-dimensional hyperbolic telegraph equation using cubic B-spline quasi-interpolation*, Int. J. Math. Comput. Sci. **52**(2011), 935-940.

[202] J. Rashidinia, S. Jamalzadeh, F. Esfahani, *Numerical solution of one dimensonal telegraph equation using cubic B-spline collocation method*, J. Interpo. Approx. Sci. Comput. (2014), 1-8

[203] M. Dosti, A. Nazemi, *Quartic B-Spline collocation method for solving one-dimensional hyperbolic telegraph equation*, J. Inform. Computing Sci. **7**(2012), no. 2, 83-90.

[204] R. C. Mittal, R. Bhatia, *Numerical solutions of second order one dimensonal hyperbolic telegraph equation by cubic B-spline collocation method*, Appl. Math. Comput. **222**(2013), 496-506.

[205] X. Li, S. Zhang, Y. Wang, H. Chen, *Analysis and application of the element-free Galerkin method for nonlinear sine-Gordon and generalized sinh-Gordon equations*, Comput. Math. Appl. **71**(2016), no. 8, 1655-1678.

[206] R. C. Mittal, A. Tripathi, *Numerical solutions of two-dimensional Burgers' equations using modified Bi-cubic B-spline finite elements*, Eng. Comput. **32**(2015), no. 5, 1275-1306.

[207] R. C. Mittal, R. Bhatia, *A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method*, Appl. Math. Comput. **244**(2014), 976-997.

[208] H. Liu, *A note on Jacobi elliptic function solutions for the modified Korteweg-de Vries equation*, J. King Saud Uni. Sci. **26**(2014), no. 2, 159-160.

[209] C. Hufford, Y. Xing, *Superconvergence of the local discontinuous Galerkin method for the linearized Korteweg-de Vries equation*, J. Comput. Appl. Math. **255**(2014), 441-455.

[210] T. Trogdon, B. Deconinck, *Numerical computation of the finite-genus solutions of the Korteweg-de Vries equation via Riemann-Hilbert problems*, Appl. Math. Lett. **26**(2013), no. 1, 5-9.

[211] T. Grava, C. Klein, *A numerical study of the small dispersion limit of the Korteweg-de Vries equation and asymptotic solutions*, Physica D: Nonlinear Phenomena, **241**(2012), no. 1, 2246-2264.

[212] J. A. Leach, *The large-time development of the solution to an initial-value problem for the generalized Korteweg-de Vries equation*, Appl. Math. Lett. **24**(2011), no. 2, 214-218.

[213] B. V. R. Kumar, M. Mehra, *Time-accurate solutions of Korteweg-de Vries equation using wavelet Galerkin method*, Appl. Math. Comput. **162**(2005), no. 1, 447-460.

[214] A. R. Bahadr, *Exponential finite-difference method applied to Korteweg-de Vries equation for small times*, Appl. Math. Comput. **160**(2005), no. 3, 675-682.

[215] E. N. Aksan, A. Ozde, *Numerical solution of Korteweg-de Vries equation by Galerkin B-spline finite element method*, Appl. Math. Comput. **175**(2006), no. 2, 1256-1265.

[216] S. Ozer, S. Kutluay, *An analytical-numerical method for solving the Korteweg-de Vries equation*, Appl. Math. Comput. **164**(2005), no. 3, 789-797.

[217] U. M. Ascher, R. I. McLachlan, *Multisymplectic box schemes and the Korteweg-de Vries equation*, Appl. Numer. Math. **48**(2004), no. 3-4, 255-269.

[218] S. Kutluay, A. R. Bahadir, A. Özde, *A small time solutions for the Korteweg-de Vries equation*, Appl. Math. Comput. **107**(2000), no. 2-3, 203-210.

[219] M. Idrees, S. Islam, S. I. A. Tirmizi, S. Haq, *Application of the optimal homotopy asymptotic method for the solution of the Korteweg-de Vries equation*, Math. Comput. Model. **55**(2012), no. 3-4, 1324-1333.

[220] N. Gücüyenen, G. Tanoğlu, *On the numerical solution of Korteweg-de Vries equation by the iterative splitting method*, Appl. Math. Comput. **218**(2011), no. 3, 777-782.

[221] J. Sarma, *Solitary wave solution of higher-order Korteweg-de Vries equation*, Chaos Solitons Fract. **39**(2009), no. 1, 277-281.

[222] M. Dehghan, A. Shokri, *A numerical method for KdV equation using collocation and radial basis functions*, Nonlinear Dyn. **50** (2007), no. 1, 111-120.

[223] B. K. Singh, G. Arora, P. Kumar, *A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme*, Ain Shams Eng. J. Available online 1 December 2016, ISSN 2090-4479, https://doi.org/10.1016/j.asej.2016.11.008.

[224] N. A. Kudryashov, *On wave structures described by the generalized Kuramoto-Sivashinsky equation*, Appl. Math. Lett. **49**(2015), 84-90.

[225] M. Chen, H. Hu, H. Zhu, *Consistent Riccati expansion and exact solutions of the Kuramoto-Sivashinsky equation*, Appl. Math. Lett. **49**(2015), 147-151.

[226] Y. Bozhkov, S. Dimas, *Group classification and conservation laws for a two-dimensional generalized Kuramoto-Sivashinsky equation*, Nonlinear Anal. Theory Methods Appl. **84**(2013), 117-135.

[227] L. Bo, Y. Jiang, *Large deviation for the nonlocal Kuramoto-Sivashinsky SPDE*, Nonlinear Anal. Theory Methods Appl. **82**(2013), 100-114.

[228] E. Dabboura, H. Sadat, C. Prax, *A moving least squares meshless method for solving the generalized Kuramoto-Sivashinsky equation*, Alexandria Eng. J. **55**(2016), no. 3, 2783-2787.

[229] M. Lakestani, M. Dehghan, *Numerical solutions of the generalized Kuramoto-Sivashinsky equation using B-spline functions*, Math. Comput. Model. **36**(2012), no. 2, 605-617.

[230] S. Haq, N. Bibi, S. I. A. Tirmizi, M. Usman, *Meshless method of lines for the numerical solution of generalized Kuramoto-Sivashinsky equation*, Appl. Math. Comput. **217**(2010), no. 6, 2404-2413.

[231] R. C. Mittal, G. Arora, *Quintic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation*, Commun. Nonlin. Sci. Numer. Simul. **15**(2010), no. 10, 2798-2808.

[232] L. Wazzan, *A modified tanh-coth method for solving the general Burgers-Fisher and the Kuramoto-Sivashinsky equations*, Commun. Nonlinear Sci. Numer. Simul. **14**(2009), no. 6, 2642-2652.

[233] H. Lai, C. Ma, *Lattice Boltzmann method for the generalized Kuramoto-Sivashinsky equation*, Physica A, **388**(2009), no. 8, 1405-1412.

[234] L. Cueto-Felgueroso, J. Peraire, *A time-adaptive finite volume method for the Cahn-Hilliard and Kuramoto-Sivashinsky equations*, J. Comput. Phy. **227**(2008), no. 24, 9985-10017.

[235] A. H. Khater, R. S. Temsah, *Numerical solutions of the generalized Kuramoto-Sivashinsky equation by Chebyshev spectral collocation methods*, Comput. Math. Appl. **56**(2008), no. 6, 1465-1472.

[236] D. Michelson, *Radial asymptotically periodic solutions of the Kuramoto-Sivashinsky equation*, Physica D: Nonlinear Phenomena, **237**(2008), no. 3, 351-358.

[237] Y. Cao, E. S. Titi, *Trivial stationary solutions to the Kuramoto-Sivashinsky and certain nonlinear elliptic equations*, J. Diff. Equ. **231**(2006), no. 2, 755-767.

[238] A. M. Wazwaz, *New solitary wave solutions to the Kuramoto-Sivashinsky and the Kawahara equations*, Appl. Math. and Comput. **182**(2006), no. 2, 1642-1650.

[239] Y. Xu, C. W. Shu, *Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations*, Comput. Methods Appl. Mech. Eng. **195**(2006), no. 25, 3430-3447.

[240] G. Akrivis, Y. S. Smyrlis, *Implici-explicit BDF methods for the Kuramoto-Sivashinsky equation*, Appl. Numer. Math. **51**(2004), no. 2-3, 151-169.

[241] B. Abdel-Hamid, *An exact solution to the Kuramoto-Sivashinsky equation*, Phy. Lett. A, **263**(1999), no. 4-6, 338-340.

[242] R. C. Mittal, S. Dahiya, *A quintic B-spline based differential quadrature method for numerical solution of Kuramoto-Sivashinsky equation*, Int. J. Nonlin. Sci. Num. **18**(2017), no. 2, 103-114.

[243] P. Coullet, C. Elphick, D. Repaux, *Nature of spatial chaos*, Phys. Rev. Lett. **58**(1987), 431-434.

[244] W. V. Saarloos, *Dynamical velocity selection: marginal stability*, Phys. Rev. Lett. **58**(1987), 2571-2574.

[245] G. T. Dee, W. V. Saarloos, *Bistable systems with propagating fronts leading to pattern formation*, Phys. Rev. Lett. **60**(1988), 2641-2644.

[246] S. W. Van, *Front propagation into unstable states: marginal stability as a dynamical mechanism for velocity selection*, Phys. Rev. Lett. A **37**(1988), 211-229.

[247] W. Zimmerman, *Propagating fronts near a Lifschitz point*, Phys. Rev. Lett. **66**(1991), 15-46.

[248] R. M. Hornreich, M. Luban, S. Shtrikman, *Critical behaviour at the onset of k-space instability on the k line*, Phys. Rev. Lett. **35**(1975), 1678-1681.

[249] V. Rottschafer, A. Doelman, *On the transition from the Ginzburg-Landau equation to the extended Fisher-Kolmogorov equation*, Physica D, **118**(1998), 261-292.

[250] L. A. Peletier, W. C. Troy, *A topological shooting method and the existence of kinks of the extended Fisher-Kolmogorov equation*, Topol. Methods Nonlinear Anal. **6**(1995), 331-355.

[251] L. A. Peletier, W. C. Troy, *Chaotic spatial patterns described by the extended Fisher-Kolmogorov equation*, J. Differ. Eqs. **129**(1996), 458-508.

[252] R. D. Benguria, M. C. Depassier, *On the transition from pulled to pushed monotonic fronts of the extended Fisher-Kolmogorov equation*, Physica A **356**(2005), 61-65.

[253] P. Danumjaya, A. K. Pani, *Finite element methods for the extended Fisher-Kolmogorov equation*, Research Report: IMGRR-2002-3, Industrial Mathematics Group, Department of Mathematics, IIT, Bombay, 2002.

[254] M. Aghamohamadi, J. Rashidinia, R. Ezzati, *Tension spline method for solution of non-linear Fisher's equation*, Appl. Math. Comput. **249**(2014), no. 15, 399-407.

[255] R. C. Mittal, S. Dahiya, *A study of quintic B-spline based differential quadrature method for a class of semi-linear Fisher-Kolmogorov equations*, Alexandria Eng. J. (2016), http://dx.doi.org/10.1016/j.aej.2016.06.019

[256] N. Khiari, K. Omrani, *Finite difference discretization of the extended Fisher-Kolmogorov equation in two dimensions*, Comput. Math. Appl. **62**(2011), 4151-4160.

Research publications/Conferences/Workshops attended

Research publications:

1. G. Arora, V. Joshi, **"A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes"**, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.06.013 ISSN: 2090-4479.

2. G. Arora, V. Joshi, **"A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions"**, Alexandria Eng. J. (2017), http://dx.doi.org/10.1016/j.aej.2017.02.017

3. G. Arora and V. Joshi, **"Comparison of Numerical Solution of 1D Hyperbolic Telegraph Equation using B-Spline and Trigonometric B-Spline by Differential Quadrature Method"**, Indian Journal of Science and Technology, Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106356, December 2016

4. G. Arora and V. Joshi, **"A hybrid approach for numerical solution of nonlinear Schrödinger equation in one and two dimensions"**, (communicated).

5. G. Arora and V. Joshi, **"Numerical solution of sine-Gordon equation in one and two dimensions using modified trigonometric differential quadrature method"**, (communicated).

6. G. Arora and V. Joshi, **"A computational approach using trigonometric differential quadrature method for numerical simulation of 2D Fisher's equation"**,(communicated).

7. G. Arora and V. Joshi, **"Numerical solutions of the generalized Burgers-Huxley equation using trigonometric differential quadrature method"**, (communicated).

8. G. Arora and V. Joshi, **"Numerical solution of generalized non-linear fourth order partial differential equation using quintic trigonometric differential quadrature method"**, (communicated).

9. G. Arora and V. Joshi, **"Numerical solution of two dimensional extended Fisher-Kolmogorov equation using quintic trigonometric differential quadrature method"**, (communicated).

Conferences:

1. Paper presented on "Travelling wave solutions of the generalized Burgers-Huxley equation", at **International conference on mathematical techniques in engineering application** at Graphic Era University, Dehradun, India. (April 29-30, 2016).

2. Paper presented on "Comparison of numerical solution of 1D hyperbolic telegraph equation using B-spline and trigonometric B-spline by differential quadrature method", at **Shannon 100-3rd International Conference on Computing Sciences**, in Lovely Professional University, Punjab, India. (April 8-9, 2016).

3. Paper presented on "Solution of 1D hyperbolic telegraphic equation by differential quadrature method", at **Ist International Conference on Modern Mathematical Methods and High Performance Computing in Science and Technology**, in Raj Kumar Goel Institute of Technology, Ghaziabad, India. (December 27-29, 2015).

Workshops:

1. "Advanced Level Training Programme on Differential equations", at Birla Institute of Technology and Science, Pilani-K. K. Birla Goa Campus, organized by Indian Institute of Technology, Bombay (May 25-June14, 2015).

2. "Advanced Summer School on Hyperbolic Conservation Laws", at TIFR-CAM, Bangalore (May 16-June 4, 2016).

The front page of published papers are as follows:

Indian Journal of Science and Technology, *Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106356, December 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Comparison of Numerical Solution of 1D Hyperbolic Telegraph Equation using B-Spline and Trigonometric B-Spline by Differential Quadrature Method

Geeta Arora and Varun Joshi*

Department of Mathematics, Lovely Professional University, Punjab – 144411, India;
geetadma@gmail.com, varunjoshi20@yahoo.com

Abstract

Objectives: This paper aims to compute the approximate solution of one dimensional (1D) hyperbolic telegraph equation with appropriate primary and limiting conditions. **Methods/Statistical Analysis**: To find the approximate solution, two different modified spline basis function are used with the differential quadrature method and splines are used to compute the weighting coefficients and thus the equation is transformed to a set of first order conventional differential equations which is further solved by the SSP-RK43 method. Three test problems of this equation are simulated to establish the precision and usefulness of the proposed scheme. **Findings:** The obtained numerical results are found to be good in terms of accuracy, efficiency and simplicity. To validate the computed results using proposed scheme, various comparisons at different time levels has been done in the form of L_2 and L_∞ errors. These errors are compared and enlisted in the form of tables with computed errors enlisted in literature. **Application/Improvements:** Being an important equation of nuclear material science, one dimensional (1D) hyperbolic telegraphs equation needs to take care in the sense of better numerical solution. In this context, a successful effort has been done in this research article by proposing a hybrid numerical scheme.

Keywords: 35K57, Differential Quadrature Method, Hyperbolic Telegraphs Equation, Mathematics Subject Classification (2010): 65M06, Modified Cubic B-Spline, Modified Trigonometric B-Spline, SSP-RK43

1. Introduction

The telegraph equation is utilized to outline the reaction diffusion in numerous branches of emerging sciences. This mathematical equation premised for crucial equations of nuclear material science. It is used to demonstrate the vibrations of structures, e.g. structures, shafts, and machines. It likewise emerges in the investigation of throb blood stream in supply routes, in the 1D irregular movement of bugs along a hedge[1,2] and also play a significant part in demonstrating of numerous appropriate problems like signal investigation[3], wave propagation[4], random walk theory[5] etc. This mathematical equation is regularly utilized as a part of signal investigation for transmission and proliferation of electrical signs[6] furthermore has applications in different fields[7]. This equation is given as:

$$u_{tt}(x,t)+2\alpha u_t(x,t)+\beta^2 u(x,t)=u_{xx}(x,t)+f(x,t) \quad x\in[a,b] \quad t\geq 0 \tag{1}$$

With initial or starting conditions (ICs)

$$u(x,t_0)=g_1(x)\ ,\ u_t(x,t_0)=g_2(x),\ \ x\in[a,b], \tag{2}$$

and with limiting conditions (BCs)

$$u(a,t)=\psi_0\ ,\ u(b,t)=\psi_1,\quad t>0 \tag{3}$$

where f, g_1, g_2, ψ_1, ψ_2 are known functions and $u_{tt}=\frac{\partial^2 u}{\partial t^2}, u_t=\frac{\partial u}{\partial t}, u_{xx}=\frac{\partial^2 u}{\partial x^2}$.

*Author for correspondence

ARTICLE IN PRESS

Ain Shams Engineering Journal (2016) **xxx**, xxx–xxx

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

ENGINEERING PHYSICS AND MATHEMATICS

A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes

Geeta Arora, Varun Joshi *

Department of Mathematics, Lovely Professional University, Punjab, India

Received 6 January 2016; revised 13 April 2016; accepted 21 June 2016

KEYWORDS

Differential equation;
Trigonometric B-spline;
Collocation method;
Fisher's equation;
Stability;
Convergence

Abstract In this article, trigonometric B-spline collocation method is used to compute the numerical solution of nonlinear Fisher's equation, in which the nonlinear term is locally linearized. This equation arises in many biological and chemical processes. The consistency of the proposed method is shown using the concept of stability and convergence and to validate the numerical scheme, the obtained theoretical results for convergence are given. We have simulated certain numerical examples of the Fisher's equation and compared their results with the exact solutions of the problems. The numerical results are found to be in good agreement with the exact solution. The accuracy and efficiency of the method are discussed by computing L_2 and L_∞ norm which are represented in the forms of tables and figures.

1. Introduction

It has been observed that nonlinear partial differential equations arise in modelling of various phenomena in engineering and sciences. One of the important equations is Fisher's equation which has well defined applications in various biological and chemical processes, and in engineering such as gene propagation [1,2], combustion [3], autocatalytic chemical reaction [4], and tissue engineering [5,6]. This one dimensional nonlinear parabolic partial differential equation proposed by Fisher [1], is a reaction diffusion type of equation that examines the wave proliferation of a beneficial quality gene in a population. It also describes the kinetic advancing rate of an advantageous gene to portray the propagation of viral mutant in an infinitely long habitat. So the study of this type of partial differential equation becomes a relevant area of research.

Genetic engineering in animals has been increased significantly in recent years and this made it a new area of research in which different techniques are being applied to change the genetic makeup of cells with the purpose of making an organism better in some way. By changing the genome, engineers are able to give desirable properties to various organisms. The most exciting potential applications of genetic modelling are concerned with animal breeding and treatment of their genetic disorders. Fisher's equation plays an important role in

* Corresponding author.
E-mail addresses: geetadma@gmail.com (G. Arora), varunjoshi20@yahoo.com (V. Joshi).

Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.asej.2016.06.013
2090-4479 © 2016 Faculty of Engineering, Ain Shams University. Production and hosting by Elsevier B.V.

Please cite this article in press as: Arora G, Joshi V, A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes, Ain Shams Eng J (2016), http://dx.doi.org/10.1016/j.asej.2016.06.013

ARTICLE IN PRESS

Alexandria Engineering Journal (2017) **xxx**, xxx–xxx

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions

Geeta Arora, Varun Joshi *

Department of Mathematics, Lovely Professional University, Punjab, India

Received 30 December 2016; revised 30 January 2017; accepted 13 February 2017

KEYWORDS

SSP-RK 43;
Burgers' equation;
Differential quadrature method;
Modified trigonometric B-spline

Abstract This article aims to obtain the numerical solution of nonlinear Burgers' equation in one and two dimensions using hybrid trigonometric differential quadrature method. Simulating five different test problems, the obtained results are compared with exact solution as well as the numerical solution obtained by other researchers. The obtained numerical solutions of this equation are found to have a good match with the exact solutions. To demonstrate the applicability and robustness of the proposed scheme the L_2 and L_∞ norms are computed and the results are presented both numerically and graphically. The stability of the proposed scheme using the matrix method is also discussed in this article.

1. Introduction

Relation between convection and diffusion or diffusion and reaction portrays the numerous physical phenomenons in physical sciences. From a physical perspective, the basic equations to portray a wide assortment of issues in physical, compound, natural, and designing sciences are the diffusion-reaction and the convection-diffusion process. These procedures are being modeled by many nonlinear partial differential equation given by numerous new experiences concerned with the association of nonlinearity with dispersion. One of the significant nonlinear equations is Burgers' equation which is very important in various fields of research. This well-known equation in one dimension can be expressed as follows:

$$u_t + \alpha u u_x = v u_{xx} \tag{1}$$

here kinematic consistency is represented by v. The two dimensional form of Burgers' equation is

$$u_t = -uu_x - uu_y + \frac{1}{R}(u_{xx} + u_{yy}) \tag{2}$$

where R is Reynolds number.

Burgers' equation is also called as viscid Burgers' equation and gives rise to a form known as inviscid Burgers' equation when the kinematic thickness is not considered. This equation acts as a good example of harmony among time development, nonlinearity, and dispersion. In nonlinear mathematical models this is one of the most easiest model or Partial differential equation (PDE) for waves with diffusive term in liquid elements. Burgers in 1948 initially builds up this equation

* Corresponding author.
E-mail addresses: geetadma@gmail.com (G. Arora), varunjoshi20@yahoo.com (V. Joshi).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

http://dx.doi.org/10.1016/j.aej.2017.02.017