Differential Quadrature Method for Non-linear Fractional Partial Differential Equations

Muhammad Umair
Regn.#00000172902

A thesis submitted in partial fulfillment of the requirements for the degree of **Master of Science** in **Mathematics**

Supervised by: Dr Umer Saeed

Department of Mathematics

School of Natural Sciences
National University of Sciences and Technology
H-12, Islamabad, Pakistan

2019

FORM TH-4

National University of Sciences & Technology

MS THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: Mr. Muhammad Umair, Regn No. 00000172902 Titled: Differential Quadrature Method for Non-Linear Fractional Partial Differential Equations be accepted in partial fulfillment of the requirements for the award of **MS** degree.

Examination Committee Members

1. Name: DR. MUHAMMAD ASIF FAROOQ Signature: ____

2. Name: DR. MUJEEB UR REHMAN Signature: ____

External Examiner: DR. NAZIR AHMAD MIR Signature: ____

Supervisor's Name DR. UMER SAEED Signature: ____

Head of Department

28/02/19
Date

COUNTERSINGED

Date: 28/02/2019

Dean/Principal

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by **Mr. Muhammad Umair,** (Registration No. **00000172902**), of **School of Natural Sciences** has been vetted by undersigned, found complete in all respects as per NUST statutes/regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/M.Phil degree. It is further certified that necessary amendments as pointed out by GEC members and external examiner of the scholar have also been incorporated in the said thesis.

Signature: ______________________

Name of Supervisor: Dr. Umer Saeed

Date: 28/02/19

Signature (HoD): ______________________

Date: 28/02/19

Signature (Dean/Principal): ______________________

Date: 28/02/19

Dedicated to

My beloved parents and respected teachers.

Acknowledgements

All praise to **Almighty Allah** and His beloved **Prophet Hazrat Muhammad (P.B.U.H.)**. I would like to express my sincere gratitude to my thesis advisor Dr. Umer Saeed, whose guidance, continuous encouragement and support right from the start to the end enabled me to successfully complete this work. His constructive remarks during the meetings we held together helped me in understanding the problems and the relevant concepts. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for this work. Besides my advisor, I would like to express my sincere appreciation to the members of the mathematics department of SNS, especially to Dr. Mujeeb-ur-Rehman and Dr. Muhammad Asif Farooq for their valuable guidance and helpful suggestions in every aspect from the very beginning onwards. Also I would like to thank my college life teachers Mr. Khurram Shehzad and Mr. Muhammad Aslam Shakir for their prayers and motivation. There are no words to thank my parents who supported me over the years and always inspired me. I am extremely grateful to my beloved sisters and elder brothers Saif-ur-Rehman and Shafqat Rasool for their love, prayers, support and sacrifices for educating and preparing me for my future. I expand my thanks to Dr. Mobina Hashmi who shared her residence with me during my stay in Islamabad, and extended brotherly love towards me. These acknowledgments would not be complete without mentioning my friends: Muhammad Aleem, Zeshan Zulifqar, Mohsin Nawaz, Zain-ul-Abdeen, Muhammad Asif Ijaz, Fazal Abbas, Amir Rahim and M. Huzaifa Yaseen. I would like to thank them for their support and encouragement. I am pleased to have such jolly fellows who have always brought smile to my face. Thank you.

Abstract

This thesis presents the modified cubic B-splines and polynomial based differential quadrature method. We extend the differential quadrature method (DQM) for solving time and space fractional non-linear partial differential equations on a semi infinite domain. The fractional derivative of Lagrange polynomial is the big hurdle in classical DQM. To overcome this problem, we represent the Lagrange polynomial in terms of shifted Legendre polynomial. We construct a transformation matrix, which transforms the Lagrange polynomial into shifted Legendre polynomial of arbitrary order. We obtain the new weighting coefficients matrices for space fractional derivatives by shifted Legendre polynomials and use these in conversion of a non-linear fractional partial differential equation into a system of fractional ordinary differential equations (ODEs). Adam Bashforth Moulton predictor-corrector approach is implemented to solve the reduced system of fractional order, α, ODEs. A very small number of grid points leads very accurate results while adopting the proposed scheme. Comparison analysis of the proposed method with some well known methods and with exact solutions are provided. Convergence analysis for the proposed method is also discussed. Many engineers can utilize the present method for solving their non-linear time and space fractional models.

Contents

1 Introduction **1**

1.1 Approximation of derivatives by DQM 3

1.2 Determination of weighting coefficients 4

2 B-Splines Differential Quadrature Method **10**

2.1 Splines 10

2.2 B-Splines 11

2.2.1 Properties of B-Splines functions 12

2.2.2 Types of B-splines functions 12

2.3 Modified cubic B-splines DQM 14

2.4 Determination of weighting coefficients using B-splines 15

2.5 The procedure of implementation of DQM 18

2.6 Application of the modified cubic B-splines DQM 19

2.6.1 Generalized Fitzhugh-Nagumo equations 21

2.6.2 Numerical solutions of Burgers' equation 24

2.6.3 Numerical solutions of non-linear generalized Burger-Fisher equation 27

3 A Modified Method for Solving Non-linear Time and Space Fractional Partial Differential Equations **31**

3.1 Preliminaries 32

3.2 Development of the proposed method 33
3.2.1 Transformation matrices 34
3.2.2 Weighting coefficient matrices for fractional derivatives 35
3.3 The procedure of implementation 37
3.4 Convergence analysis 38
3.5 Application 40
3.5.1 Fractional Burgers' equation 41
3.5.2 Fractional order Huxley equation 41
3.5.3 Generalized fractional Burger-Fisher equation 46

4 Conclusion **51**

Bibliography **58**

List of Figures

2.1 Comperison of numerical and exact solutions of equation (2.23) with absolute error L_∞ for $\Delta t = 0.05,\ \rho = 3/4,\ -2$ upto time t=5. 22

2.2 Comparison of the obtained results and exact solutions of problem (3.33) along with absolute error L_∞ for $\Delta t = 0.05,\ \rho = 1/2,\ 3/4$ upto time t=5. 25

3.1 Numerical results of equation (3.29) by proposed scheme at $x = 0.5$ and $\hat{n} = 5, 7,\ h = 0.01$ for different values of α and β. 42

3.2 Numerical results of Example (3.31) by proposed scheme at $x = 0.5$ and $\hat{n} = 5,\ h = 0.01$ for different values of α and β. 45

3.3 Numerical results of Example (3.33) by present method for different values of $\alpha,\ \beta_1$ and β_2 at $x = 0.5,\ h = 0.0025,\ \hat{n} = 5, 7$. 48

List of Tables

2.1 Values of $\Phi_i(w)$ at knot points 15

2.2 Values of $\Phi_i'(w)$ at knot points 15

2.3 Values of $\Phi_i''(w)$ at knot points 16

2.4 Comparison of numerical and exact solutions at $\rho = 0.75$, h=0.05 and $\hat{n} = 40$ for different values of x and t. 23

2.5 Comparison between numerical and exact solutions with $\rho = -2$, h=0.05 and $\hat{n} = 40$. 24

2.6 Max absolute error L_∞ of problem (2.25) at different time t with $\rho = 1/2,\ 3/4$ and $h = 0.05$. 26

2.7 Numerical results of equation (2.27) at different values of α, β and time t by present method. 27

2.8 Comparison of solutions by present method and exact solutions for $h = 0.002$ and $\hat{n} = 5$. 28

2.9 Numerical results of equation (2.29) for $\gamma = 1$ and $\delta = \rho$ at different times t. 29

2.10 Numerical results of equation (2.29) for $h = 0.005$, $\hat{n} = 7$, $\gamma = 2$ and $\delta = \rho = 1$. 30

3.1 Numerical results of problem (3.29) by proposed scheme at different values of α, β_1 and β_2, with $h = 0.002$, $\hat{n} = 5$. 43

3.2 Numerical results of equation (3.31) by proposed method at different values of α, and β, with $h = 0.005$, $\hat{n} = 7$. 44

3.3 Comparison between solutions of problem (3.31) by present method with exact solutions and Haar wavelet method at $\beta = 2$ and $h = 0.01,\ \hat{n} = 5$. 46

3.4 Numerical results of problem (3.33) at different values of $\alpha,\ \beta_1$ and β_2 with $h = 0.005,\ \hat{n} = 7, \gamma = 2$ and $\delta = \rho = 1$. 48

3.5 Comparison of the solutions obtained by present method, variational iteration method, differential transformation method and exact solutions for integer values of $\alpha,\ \beta_1,$ and β_2 with $h = 0.01,\ \hat{n} = 5, \gamma = 1$ and $\delta = \rho = 0.001$. 49

3.6 Comparison of the obtained results and the results by reduced differential transformation method at integer values of $\alpha,\ \beta_1$ and β_2 with $h = 0.01,\ \hat{n} = 5, \gamma = 2$ and $\delta = \rho = 0.001$. 49

3.7 Comparison between solutions by proposed method and Homotopy perturbation method (HPM) for different values of δ and ρ, at integer values of $\alpha,\ \beta_1$ and β_2 with $h = 0.01,\ \hat{n} = 5, \gamma = 2$. 50

Chapter 1

Introduction

Fractional calculus is the field of mathematics in which non-integer order derivatives and integrals are involved. Fractional order partial and ordinary differential equations are widely used in various fields of engineering and sciences. These equations are significantly served as a tool for the mathematical formulation of various scientific and engineering problems involving functions of several variables. The solutions of these differential equations are very applicable due to their practical significance and paramount role in the field of science. During some last decades a number of applications of fractional calculus have been appeared, specially in the biological disease problems and bioengineering [1, 2], chemical reactions, underwater flow problems in fluid dynamics [3], signal processing [4], viscoelasticity theory [5, 6], fractional quantum mechanics, robotics and electronics [7, 8, 9] etc. Several methods have been developed and many approaches are utilized to obtain the solutions of fractional differential equations. Some of these methods include the homotopy perturbation transform techique [10], the Haar wavelet picard method [11, 12], Adomian decomposition method (ADM) [13, 14, 15, 16] etc.

In this work, we describe a numerical method for both integer and fractional order partial differential equations (PDEs). There are many theoretical results available in the literature about the existence and uniqueness of the solutions of non-linear partial differential equations. Partial differential equations that model the daily life problems are mostly non-linear in nature. Due to the non-linear inherent nature of these problems, the analytic solutions of corresponding partial differential equations are not easy to find. There are many challenges regarding to these equations which are either very tough to solve explicitly or have no solution at all. There are limited class of simple specific non-linear differential equations that can be solved analytically. In some cases, the existence of the analytic solutions of partial differential equations can be confirmed but they are not known or can not be found in closed form. Sometimes, the solutions are known in complex closed form or in the form of infinite series along with complicated coefficients involving complex integrals which are very difficult to evaluate such as Fourier series, Laplace transform etc. Therefore, the methods which are used to find the analytic form of solutions are very complicated and costly in the sense of computations and efforts. To overcome these hurdles, numerous numerical techniques are developed and implemented to figure out the solutions of one and higher dimensional PDEs. In numerical schemes, the basic idea is to discretize the whole domain into finite discrete points called the grid points, mesh points, or nodes. Then we approximate the solution at these nodal points and on the boundary points of the given domain. The competence of a numerical technique bank on a number of aspects such as ease of implementation of the algorithm, efficiency and accuracy of the results, computational labor, convergence, stability and the versatility of the method to handle the variety of problems etc. Numerical techniques like spectral method [17], finite difference [18], finite element [19], finite volume [20] and Adomian decomposition [21] methods have attained a lot of attention.

Finite difference and finite element are predominant techniques in solving PDEs. In finite difference method, the derivative of an unknown function is approximated by sums and differences of the function values at some nodal points. The given PDE gets reduce to system of algebraic equations. Suitable differencing schemes need to be chosen for different problems. The step size in nodes determines the accuracy of the technique. In case of one dimensional problem, the technique is quite efficient but its formulation for more than one dimensional problems is a laborious task. More research needs to be carried out for the development of efficient schemes for solving higher dimensional PDEs owing to their vast applicability and significance.

In recent few years, differential quadrature method (DQM) has become progressively popular numerical scheme to solve a verity of linear and non-linear PDEs . The main idea about DQM was initially originated by R.E Bellman [22] in 1972 and later on further modifications for this technique were carried out by Quan and Chang [23]. Bellman [22] et al. followed the idea of conventional integral quadrature and introduced the idea of DQM. It is a high order numerical discretization scheme in comparison of lower order methods like finite difference and finite element methods, because DQM needs a considerably less nodal points. Consequently, differential quadrature method is very economical in terms of storage space requirement and CPU time. Basically, differential quadrature method is identical to pseudo spectral (collocation) techniques. But, it directly calculates the function values at nodes rather than calculating the spectral variables. In differential quadrature method, the spatial derivative of an unknown function is approximated in a particular direction using the weighted sum of all function values at nodes. The most significant step while employing DQM is the determination of weighting coefficients. This method requires considerably less formulation and programming efforts and is also capable to handle a variety of different types of boundary conditions. It comprises of very well refined mathematical techniques. The involved mathematical techniques are explicit and easy to grasp. We can successfully utilize the differential quadrature method for a variety of problems especially the non-linear ones occurring in the fields of engineering and applied science. For detailed study, we refer the reader to [22].

1.1 Approximation of derivatives by DQM

In this section, we describe the DQM for one dimensional partial differential equations. Bellman [22] et al. initially considered a one dimensional problem. Assuming an unknown function $u(x,t)$ which is sufficiently smooth over the given entire domain. The partial derivative of u at a particular point of the domain x_q with respect to x can be taken as a linear sum of all the function values along the mesh line [24]. Discretizing the given domain $[a,b]$ in $\hat{n}$ grid points i.e $(\hat{n}-1)$ sub-intervals such that

$a = x_1, x_2, \cdots, x_{\hat{n}} = b$, the first and second order partial derivatives of u at a particular grid point x_q can be approximated as

$$\begin{aligned} u_x(x_q, t) &= \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell}^{(1)} u(x_\ell, t), \\ u_{xx}(x_q, t) &= \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell}^{(2)} u(x_\ell, t), \end{aligned} \tag{1.1}$$

for $q = 1, 2, \cdots, \hat{n}$, where $\hat{\Omega}_{q\ell}^{(1)}$ and $\hat{\Omega}_{q\ell}^{(2)}$ for $q, \ell = 1, 2, \cdots, \hat{n}$ represent the weights corresponding to the first and second order derivatives and $u(x_\ell, t)$ denotes the function values at ℓ^{th} node x_ℓ. The foremost task which is of paramount, is the determination of the weights. Once these weights are obtained, we can approximate the derivatives at all discrete points. DQ method is drafted in such a way that the given PDE gets reduce to a system of first order ODEs. The resulting system of ODEs is then solved by implementing some suitable numerical techniques such as strong stability-preserving Runge-Kutta method (SSP-RK43), Adam Bashforth Moulton predictor-corrector method, Runge Kutta method of order 4 (RK-4) etc. In the present work, we have utilized both the RK-4 and Adam-Bashforth predictor-corrector approaches to solve the reduced system of differential equations.

1.2 Determination of weighting coefficients

Numerous approaches are available in the literature given by the researchers to compute these weights, which are based on the different selection of test functions. These approaches include Bellman's approach [24], Chang Shu's approach [22], Quan & Chang approach [23] etc. These test functions include the Lagrange interpolation cosine functions, Lagrange interpolation polynomials, Legendre polynomials etc. In recent few years, some researchers have used B-splines [25, 26, 27, 28], Sinc functions [23, 29] and the radial basis functions [30], as test functions for the determination of weights. Bellman [24] with his associates suggested two approaches to calculate the weights for first order derivatives. In his first approach, he solved an algebraic system by using following test functions as trial functions

$$g_\kappa = x^\kappa, \qquad \kappa = 0, 1, 2 \cdots, \hat{n} - 1, \tag{1.2}$$

these are actually $\hat{n}$ test functions. From equation (1.1) the weights $\hat{\Omega}^{(1)}_{q\ell}$, q and ℓ are taken from 1 to $\hat{n}$. So, there are $\hat{n} \times \hat{n}$ number of weights. When we apply $\hat{n}$ nodes on $\hat{n}$ test functions, we get $\hat{n} \times \hat{n}$ algebraic equations for $\hat{\Omega}^{(1)}_{q\ell}$ as

$$\begin{cases} \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell} = 0, \\ \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell}\ x_\ell = 1, \\ \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell}.x^p_\ell = p.x^{p-1}_\ell, \qquad p = 2, 3, \cdots, \hat{n} - 1, \quad q = 1, 2, \cdots \hat{n}. \end{cases} \tag{1.3}$$

The matrix obtained from system (1.3) is of Vandermonde form i.e each row with terms of geometric progression, so the system (1.3) has unique solution. For large value of $\hat{n}$, the obtained matrix becomes ill-conditioned and it becomes very tedious task to compute the weights. Therefore, for this approach the nodes $\hat{n}$ can't be taken large and is usually restricted to be less than 13 [22].
In the other approach, the following functions are used as test functions

$$g_\kappa(x) = \frac{H_{\hat{n}}(x)}{(x - x_\kappa)H^{(1)}_{\hat{n}}(x_\kappa)}, \qquad \kappa = 1, 2, \cdots, \hat{n}, \tag{1.4}$$

where $H_{\hat{n}}(x)$ and $H^{(1)}_{\hat{n}}(x_\kappa)$ denotes the $\hat{n}^{th}$ degree Legendre polynomials and its first order derivative, respectively. In this approach, the roots of shifted Legendre polynomials are taken as the coordinates of the mesh points and the following algebraic formulations are obtained to compute $\hat{\Omega}_{q\ell}$ as

$$\begin{aligned} \hat{\Omega}_{q\ell} &= \frac{H_{\hat{n}}(x_q)}{(x_q - x_\ell)H^{(1)}_{\hat{n}}(x_\ell)}, \qquad \text{for } q \neq \ell, \\ \hat{\Omega}_{qq} &= \frac{1 - 2x_q}{2x_q(x_q - 1)}, \qquad \text{for } q = \ell. \end{aligned} \tag{1.5}$$

From equation (1.5), one can easily compute the weights. Bellman's second approach is not as economical as first approach, because we can not choose the coordinates of mesh points arbitrarily. This approach looks like a special case. Various efforts have been made to overcome these drawbacks. Quan and Chang [23] have suggested an approach in which they used Lagrange interpolated polynomials as trial functions to compute the weights $\hat{\Omega}_{q\ell}$. The Lagrange interpolated polynomials are given as

$$g_\kappa(x) = \frac{N(x)}{(x - x_\kappa)N^{(1)}(x_\kappa)}, \qquad \kappa = 1, 2, \cdots, \hat{n}, \tag{1.6}$$

where

$$N(x) = \prod_{p=1}^{\hat{n}} (x - x_p),$$

and

$$N^{(1)}(x_q) = \prod_{\kappa=1,\kappa\neq q}^{\hat{n}} (x_q - x_\kappa).$$

By using equation (1.6), the explicit formulations for weights are

$$\begin{aligned} \hat{\Omega}_{q\ell} &= \frac{1}{(x_\ell - x_p)} \prod_{p=1,p\neq q}^{\hat{n}} \frac{(x_q - x_p)}{(x_\ell - x_p)}, && \text{for } q \neq \ell. \\ \hat{\Omega}_{qq} &= \sum_{p=1,p\neq q}^{\hat{n}} \frac{1}{(x_q - x_p)}, && \text{for } q = \ell. \end{aligned} \tag{1.7}$$

Equation (1.7) overcomes the disadvantage of restricting the choice and number of nodes. Another breakthrough in the determination of weights was achieved by Shu [22]. He concluded that all the available schemes to compute the weights can be generalized under the analysis of a linear vector space and the analysis of higher order polynomial approximation. From the properties of linear vector space, if one basis satisfies a linear operator then all the other basis do the same i.e each base will produce the same weighting coefficients. He observed that the following relation holds

$$\sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell}^{(1)} = 0, \quad \text{or} \quad \hat{\Omega}_{qq}^{(1)} = -\sum_{\ell=1,\ell\neq q}^{\hat{n}} \hat{\Omega}_{q\ell}.$$

Shu [22] has also proposed a general formulation to find the weights corresponding to second and higher order derivatives. These can be obtained by using the following explicit relations as

$$
\begin{aligned}
\hat{\Omega}^{(\gamma)}_{q\ell} &= \gamma \left[\hat{\Omega}^{(1)}_{q\ell} \hat{\Omega}^{(\gamma-1)}_{qq} - \frac{\hat{\Omega}^{(\gamma-1)}_{q\ell}}{(x_q - x_\ell)} \right], \qquad && \text{for } q \neq \ell, \\
\hat{\Omega}^{(\gamma)}_{qq} &= - \sum_{\ell=1, \ell \neq q}^{\hat{n}} \hat{\Omega}^{(\gamma)}_{q\ell}, && \text{for } q = \ell,
\end{aligned}
\tag{1.8}
$$

where $q, \ell = 1, 2, \cdots, \hat{n}$, $\gamma = 2, 3, \cdots, \hat{n}$. Also $\hat{\Omega}^{(\gamma-1)}_{q\ell}$ and $\hat{\Omega}^{(\gamma)}_{q\ell}$ are the weights corresponding to $(\hat{\Omega} - 1)^{th}$ and $\hat{\Omega}^{th}$ order partial derivatives.

Multiplication of matrices approach

Another simple and easy approach available in literature for the determination of weighting coefficients, named as multiplications of matrices approach.

By the differential operator, we have

$$
\frac{\partial^2 g}{\partial x^2} = \frac{\partial}{\partial x} \left[\frac{\partial g}{\partial x} \right]. \tag{1.9}
$$

For our convenience, we use the simple notation $\frac{\partial^2 g}{\partial x^2}$ as $g^{''}_x$ and $\frac{\partial g}{\partial x}$ as $g^{'}_x$. When we apply the differential quadrature approximation on left side of equation (1.9), we have

$$
g^{''}_x(x_q) = \sum_{\ell=1}^{n} \hat{\Omega}^{(2)}_{q\ell} g(x_\ell), \qquad q = 1, 2, 3, \cdots, \hat{n} \tag{1.10}
$$

Again by differential quadrature approximation, the right side of equation (1.9) is

$$
g^{''}_x(x_q) = \sum_{p=1}^{\hat{n}} \hat{\Omega}^{(1)}_{qp} \, g^{'}_x(x_p), \qquad q = 1, 2, 3, \cdots, \hat{n}. \tag{1.11}
$$

$$
\begin{aligned}
&= \sum_{p=1}^{\hat{n}} \hat{\Omega}^{(1)}_{qp} \sum_{\ell=1}^{\hat{n}} \hat{\Omega}^{(1)}_{p\ell} \, g(x_\ell), \\
&= \sum_{\ell=1}^{\hat{n}} \left[\sum_{p=1}^{\hat{n}} \hat{\Omega}^{(1)}_{qp} \hat{\Omega}^{(1)}_{p\ell} \right] g(x_\ell).
\end{aligned}
\tag{1.12}
$$

By the comparison of equation (1.10) and (1.12), we have

$$\hat{\Omega}^{(2)}_{q\ell} = \sum_{\ell=1}^{\hat{n}} \hat{\Omega}^{(1)}_{qp} \hat{\Omega}^{(1)}_{p\ell}. \tag{1.13}$$

If we define two matrices as

$$[\Omega^{(1)}] = \begin{bmatrix} \hat{\Omega}^{(1)}_{11} & \hat{\Omega}^{(1)}_{12} & \cdots & \hat{\Omega}^{(1)}_{1\hat{n}} \\ \hat{\Omega}^{(1)}_{21} & \hat{\Omega}^{(1)}_{22} & \cdots & \hat{\Omega}^{(1)}_{2\hat{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Omega}^{(1)}_{\hat{n}1} & \hat{\Omega}^{(1)}_{\hat{n}2} & \cdots & \hat{\Omega}^{(1)}_{\hat{n}\hat{n}} \end{bmatrix}, \quad [\Omega^{(2)}] = \begin{bmatrix} \hat{\Omega}^{(2)}_{11} & \hat{\Omega}^{(2)}_{12} & \cdots & \hat{\Omega}^{(2)}_{1\hat{n}} \\ \hat{\Omega}^{(2)}_{21} & \hat{\Omega}^{(2)}_{22} & \cdots & \hat{\Omega}^{(2)}_{2\hat{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Omega}^{(2)}_{\hat{n}1} & \hat{\Omega}^{(2)}_{\hat{n}2} & \cdots & \hat{\Omega}^{(2)}_{\hat{n}\hat{n}} \end{bmatrix},$$

then from equation (1.13), we have

$$\left[\Omega^{(2)}\right] = \left[\Omega^{(1)}\right]\left[\Omega^{(1)}\right]. \tag{1.14}$$

Equation (1.14) shows that the weights for second order partial derivative can be calculated by the matrices product of weights of first order derivatives. In the same way, for the $p^{(th)}$ order derivative of $g(x)$, we can write

$$\frac{\partial^p g}{\partial x^p} = \frac{\partial}{\partial x}\left[\frac{\partial^{(p-1)} g}{\partial x^{(p-1)}}\right] \tag{1.15}$$

Let $\left[\Omega^{(p)}\right]$ and $\left[\Omega^{(p-1)}\right]$ are matrices of weighting coefficients for p^{th} and $(p-1)^{th}$ order derivatives respectively, we have the following recurrence relation from the application of differential quadrature approximation to equation (1.15) as

$$\left[\Omega^{(p)}\right] = \left[\Omega^{(1)}\right]\left[\Omega^{(p-1)}\right], \qquad p = 2, 3, \cdots, \hat{n} - 1. \tag{1.16}$$

Although the relation (1.16) looks simple, but there are more arithmetic operations involved as compared to the expression for weighting coefficients in Shu's recurrence formula. We can see that for the computation of each weighting coefficients by relation (1.16), there are $\hat{n}$ multiplications and $(\hat{n} - 1)$ additions, i.e total $(2\hat{n} - 1)$ arithmetic operations. While Shu's recurrence formula involves only two multiplications, one subtraction and one division, i.e total 4 arithmetic operations for the computation of each off-diagonal coefficient, whereas the computation of every diagonal coefficient from Shu's recurrence formula, there are $(\hat{n}-2)$ subtractions. Therefore, the total arithmetic operations involved in Shu's recurrence formula are substantially less than that of equation (1.16). So, when we increase the number of nodes the computation of weights through Shu's recurrence formula ought to be inexpensive and more accurate because of smaller rounding off errors as compared to equation (1.16). From the practical application it is concluded that this may effect the accuracy of the numerical solutions. In this work, first we used the modified cubic B-splines functions as trial functions to compute the weights for DQ method. We employed the modified cubic B-splines DQM to solve the integer order non-linear partial differential equations. We considered Fitz Nagumo, generalized Burger-Fisher and Burgers' equations as test problems for application purpose. We also proposed a method based on differential quadrature method for both time and space fractional linear or non-linear PDEs. The brief details about this work is discussed in next coming chapters. In Chapter 2, the brief discussion is given about the computation of weighting coefficients through B-splines and the modified cubic B-splines differential quadrature method along with its applications. Chapter 3 is related to the development of differential quadrature method for non-linear space and time fractional PDEs. Finally in Chapter 4, some concluding remarks about the work are given in details.

Chapter 2

B-Splines Differential Quadrature Method

2.1 Splines

Spline is a function which is formulated piecewise from a polynomial function. To approximate the solutions of PDEs, the piecewise polynomial functions have become essential tools. Piecewise polynomial approximations served highly accurate approximations. But, while employing piecewise polynomials the most important requirement is the smoothness of approximating curve at the nodes. Furthermore, for smooth graph of interpolant not only the continuity but also continuous differentiability of the polynomial is required on the given interval. Spline functions are the piecewise polynomials having high degree of smoothness. B-splines are basis of vector space comprises of spline functions and have minimal support.

In this chapter, we employed differential quadrature method using third degree B-splines as basis functions. To demonstrate the significant importance of B-splines as basis functions, we have explained B-spline functions of different degrees with their characteristic properties. Some times ago, days before the innovation of computer modeling, spline curves were used in shipbuilding industries by naval architects. In early 1890's the word spline was meant by a stretchable or adjustable ruler. In mathematics, spline refers to a numeric function which is piecewise defined by a polynomial and having a high degree of smoothness at the joining points. On interpolation with a spline yields similar results to interpolation with high degree polynomials while avoiding instabilities due to Runge's phenomenon.

2.2 B-Splines

The idea of B-spline was first introduced by Schoenberg [31] in 1946, who employed analytic functions while facing the problem of approximations of equidistance data. The basis splines written as B-splines are the *smooth piecewise polynomial functions.* B-splines are the spline functions with minimal support having particular degree, smoothness and domain partition. Schumacker [32] gave the first definition of B-splines by employing the idea of divided difference, after this Carl de Bore [33] independently derived a recursive relation to determine B-spline functions by applying Leibniz theorem. Consider an interval $[a, b]$ and suppose $Q = \{w_\lambda\}_{\lambda=1}^{\infty}$ be a sequence of non-decreasing real numbers such that $a = x_0$ and $b = x_{\hat{n}}$. These $w_\lambda{}'s$ are known as knots. The sequence Q named as knot sequence is the defining feature of the basis functions. The semi open interval $[w_\lambda, w_{\lambda+1})$ is named as the λ^{th} knot span. The knot sequence Q is supposed to be uniform or non-uniform depending upon the knot vectors, whether they are equally spaced or not. A p^{th} degree B-spline function covers $(p + 1)$ knot vectors or p intervals. B-spline functions defined over uniform knot sequence are said to be uniform B-spline functions. The zeroth degree B-spline for a non-decreasing knot sequence $\{w_\lambda\}_{\lambda=1}^{\infty}$ is defined as [33]

$$\bar{B}_{\lambda,0} = \begin{cases} 1, & if \;\; w_\lambda < w < w_{\lambda+1} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

We can see that the zeroth degree B-spline is just a step function. From Carl de Boor [33] recursive formula, the λ^{th} B-spline of p^{th} degree can be computed by [33]

$$\begin{aligned} \bar{B}_{\lambda,p} &= W_{\lambda,p}\ \bar{B}_{\lambda,p-1} + (1 - W_{\lambda+1,p})\ \bar{B}_{\lambda+1,p-1}, \\ W_{\lambda,p} &= \frac{w - w_\lambda}{w_{\lambda+p} - w_\lambda}, \end{aligned} \tag{2.2}$$

where w is a parameter variable. The relation (2.2) shows that we can stably evaluated the higher degree B-spline functions as a linear combinations of lower degree B-splines.

2.2.1 Properties of B-Splines functions

We selected B-splines as the basis functions. Some significant properties of B-splines are given [32, 33, 34] as

1. $\bar{B}_{\lambda,p}(w)$ is a degree p polynomial in w.
2. $\forall\ \lambda, p$ and w, $\bar{B}_{\lambda,p}(w)$ is non-negative.
3. For $p \geqslant 0$, $\bar{B}_{\lambda,p}(w)$ is non-zero polynomial on semi open interval $[w_\lambda, w_{\lambda+p+1})$.
4. $\bar{B}_{\lambda,p}(w) = 0$ if w does not belong to open interval $(w_\lambda, w_{\lambda+p+1})$,
where $\lambda = 0, 1, 2, 3, \cdots, \hat{n} - 1$.
5. For $p > 0$, $\bar{B}_{\lambda,p}(w)$ is a linear combination of two basis functions having degree $(p-1)$.
6. The relative spacing between the knots is due to the factor that influence the shape of the basis function $\bar{B}_{\lambda,p}(w)$. There is no effect on the shape of basis function and B-spline curves by translating or scaling.
7. On combining of all non-zero p^{th} degree B-splines results in unity i.e $\sum\limits_{-\infty}^{+\infty} \bar{B}_{\lambda,p}(w) = 1$.
This could be easily prove by using recurrence formula and definition of $\bar{B}_{q,0}(w)$.
8. For a given knot sequence, B-splines are linearly independent if B-splines of degree p are not vanishing over a particular knot sequence.
9. For $(\hat{n}+1)$ control points, B-splines $\bar{B}_{\lambda,p}(w),\ \lambda = 0, 1, 2, 3, \cdots, \hat{n}$ form a basis of all splines of degree p.
10. $\bar{B}_{\lambda,p}(w) = \bar{B}_{\lambda+1,p}(w+\Delta w) = \bar{B}_{\lambda+1,p}(w+2\Delta w)$ and so on, where Δw is the distance between two consecutive nodal points, i.e every function possess the same shape.

For further details about B-spline functions and their properties we refer the reader Schumaker [32], Carl D. Boor [33] and P. M Prenter [34].

2.2.2 Types of B-splines functions

We start by considering a uniform knot sequence with step size $\Delta w = h$. First degree B-splines are termed as linear B-splines. The explicit form of first degree B-splines is given by

$$\bar{B}_{\lambda,1}(w) = \begin{cases} \frac{(w-w_\lambda)}{(w_{\lambda+1}-w_\lambda)}, & w \in [w_\lambda, w_{\lambda+1}), \\ \frac{(w_{\lambda+2}-w)}{(w_{\lambda+2}-w_{\lambda+1})}, & w \in [w_{\lambda+1}, w_{\lambda+2}), \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

B-splines of degree one resemble with a Tent or Hat function, and are non-zero for consecutive two intervals.

Quadratic B-splines

B-splines of degree two are termed as quadratic B-splines and can be calculated by linear B-spline in Carl D. Boor [33] recursive formula. Explicitly, the quadratic B-splines are written as

$$\bar{B}_{\lambda,2}(w) = \frac{1}{2h} \begin{cases} \frac{(w-w_\lambda)^2}{h}, & w \in [w_\lambda, w_{\lambda+1}), \\ (w-w_\lambda)(w_{\lambda+2}-w) + (w_{\lambda+3}-w)(w-w_{\lambda+1}), & w \in [w_{\lambda+1}, w_{\lambda+2}), \\ \frac{(w_{\lambda+3}-w)^2}{h}, & w \in [w_{\lambda+2}, w_{\lambda+3}), \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

Cubic B-splines

Third degree B-splines are termed as cubic B-splines and are defined at the knots as [25]

$$\bar{B}_{\lambda,3}(w) = \frac{1}{h^3} \begin{cases} (w-w_{\lambda-2})^3, & w \in [w_{\lambda-2}, w_{\lambda-1}), \\ (w-w_{\lambda-2})^3 - 4(w-w_{\lambda-1})^3, & w \in [w_{\lambda-1}, w_\lambda), \\ (w_{\lambda+2}-w)^3 - 4(w_{\lambda+1}-w)^3, & w \in [w_\lambda, w_{\lambda+1}), \\ (w_{\lambda+2}-w)^3, & w \in [w_{\lambda+1}, w_{\lambda+2}), \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Quartic and Quintic B-splines

B-splines of degree four and degree five are termed as quartic B-splines and quintic B-splines functions [35] respectively. They are written in explicit form as [36]

$$\bar{B}_{\lambda,4} = \frac{1}{h^4}\begin{cases} (w-w_{\lambda-2})^4, & w\in[w_{\lambda-2},w_{\lambda-1}),\\ (w-w_{\lambda-2})^4-5(w-w_{\lambda-1})^4, & w\in[w_{\lambda-1},w_{\lambda}),\\ (w-w_{\lambda-2})^4-5(w-w_{\lambda-1})^4+10(w-w_{\lambda})^4, & w\in[w_{\lambda},w_{\lambda+1}),\\ (w_{\lambda+3}-w)^4-5(w_{\lambda+2}-w)^4, & w\in[w_{\lambda+1},w_{\lambda+2}),\\ (w_{\lambda+3}-w)^4, & w\in[w_{\lambda+2},w_{\lambda+3}),\\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

$$\bar{B}_{\lambda,5} = \frac{1}{h^5}\begin{cases} (w-w_{\lambda-3})^5, & w\in[w_{\lambda-3},w_{\lambda-2}),\\ (w-w_{\lambda-3})^5-6(w-w_{\lambda-1})^5, & w\in[w_{\lambda-2},w_{\lambda-1}),\\ (w-w_{\lambda-3})^5-6(w-w_{\lambda-2})^5+15(w-w_{\lambda})^5, & w\in[w_{\lambda-1},w_{\lambda}),\\ (w_{\lambda+3}-w)^5-6(w_{\lambda+2}-w)^5+15(w_{\lambda+1}-w)^5, & w\in[w_{\lambda},w_{\lambda+1}),\\ (w_{\lambda+3}-w)^5-6(w_{\lambda+2}-w)^5, & w\in[w_{\lambda+1},w_{\lambda+2}),\\ (w_{\lambda+3}-w)^5, & w\in[w_{\lambda+2},w_{\lambda+3}),\\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

2.3 Modified cubic B-splines DQM

The cubic B-splines basis functions are modified into another set of basis functions to accommodate the boundary conditions effectively. This modification is done in such a way that the basis functions are exactly equal the number of points in whole domain. The modified cubic B-splines basis functions are given as [26]

$$\begin{aligned} \Phi_1(w) &= \bar{B}_1(w)+2\bar{B}_0(w),\\ \Phi_2(w) &= \bar{B}_2(w)-\bar{B}_0(w),\\ \Phi_\kappa(w) &= \bar{B}_\kappa(w), \quad \kappa=3,4,\cdots,\hat{n}-2,\\ \Phi_{\hat{n}-1}(w) &= \bar{B}_{\hat{n}-1}(w)-\bar{B}_{\hat{n}+1}(w),\\ \Phi_{\hat{n}}(w) &= \bar{B}_{\hat{n}}(w)+2\bar{B}_{\hat{n}+1}(w), \end{aligned} \tag{2.8}$$

where $\Phi_1, \Phi_2, \cdots, \Phi_{\hat{n}}$ form a basis over interval $[a, b]$ and $\bar{B}_r's$, $r = 0, 1, 2, \cdots, \hat{n}+1$, are cubic B-splines functions. $\Phi_\ell(w)$, $\ell = 1, 2, \cdots, \hat{n}$, and its evaluated derivatives at nodal points are given Table (2.1), (2.2) and (2.3).

Table 2.1: Values of $\Phi_i(w)$ at knot points

w	Φ_1	Φ_2	Φ_3	$\cdots$	$\Phi_{\hat{n}-2}$	$\Phi_{\hat{n}-1}$	$\Phi_{\hat{n}}$
w_1	6	1					
w_2	0	4	1				
w_3		1	4	1			
$\vdots$			$\cdots$	$\cdots$	$\cdots$		
$w_{\hat{n}-2}$				1	4	1	
$w_{\hat{n}-1}$					1	4	0
$w_{\hat{n}}$						1	6

Table 2.2: Values of $\Phi'_i(w)$ at knot points

w	Φ'_1	Φ'_2	Φ'_3	$\cdots$	$\Phi'_{\hat{n}-2}$	$\Phi'_{\hat{n}-1}$	$\Phi'_{\hat{n}}$
w_1	$-6/h$	$6/h$	0				
w_2	$-3/h$	0	$3/h$				
w_3		$-3/h$	0	$3/h$			
$\vdots$			$\cdots$	$\cdots$	$\cdots$		
$w_{\hat{n}-2}$				$-3/h$	0	$3/h$	
$w_{\hat{n}-1}$					$-3/h$	0	$3/h$
$w_{\hat{n}}$						$-6/h$	$6/h$

2.4 Determination of weighting coefficients using B-splines

Consider the interval $[a, b]$ be the domain under consideration. Dividing it into a grid of uniform length $h = x_{q+1} - x_q$, where x_q and x_{q+1} are two consecutive grid points in the mesh sequence, for $q = 1, 2, \cdots, \hat{n}$. From differential quadrature method, the m^{th} order partial derivative approximation of a function u with respect to x at a particular

Table 2.3: Values of $\Phi_i''(w)$ at knot points

w	Φ_1''	Φ_2''	Φ_3''	$\cdots$	$\Phi_{\hat{n}-2}''$	$\Phi_{\hat{n}-1}''$	$\Phi_{\hat{n}}''$
w_1	0	0	0				
w_2	$6/h^2$	$-12/h^2$	$6/h$				
w_3		$6/h^2$	$-12/h^2$	$6/h^2$			
$\vdots$			$\cdots$	$\cdots$	$\cdots$		
$w_{\hat{n}-2}$				$6/h^2$	$-12/h^2$	$6/h^2$	
$w_{\hat{n}-1}$					$6/h^2$	$-12/h^2$	$6/h^2$
$w_{\hat{n}}$					0	0	0

point x_q is given as [22]

$$u_x^{(m)}(x_q, t) = \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{q\ell}^{(m)} u(x_\ell, t), \qquad q = 1, 2, \cdots, \hat{n}. \tag{2.9}$$

The weights corresponding to the first order derivatives are calculated using modified cubic B-spline basis functions $\Phi_p(w)$ as

$$\Phi_p^{'}(w_\lambda) = \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{\lambda\ell}^{(1)} \Phi(w_\ell), \qquad \lambda, p = 1, 2, \cdots, \hat{n}. \tag{2.10}$$

For $w_\lambda = w_1$, on the boundary point w_1, equation (2.10) becomes

$$\Phi_p{'}(w_1) = \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{1\ell}^{(1)} \Phi(w_\ell), \qquad p = 1, 2, \cdots \hat{n}. \tag{2.11}$$

From equation (2.11), we get the following tridiagonal system as

$$\begin{bmatrix} 6 & 1 & & & & & \\ 0 & 4 & 1 & & & & \\ & ... & ... & ... & & & \\ & & 1 & 4 & 1 & & \\ & & & 1 & 4 & 0 & \\ & & & & & 1 & 6 \\ & & & & & & \end{bmatrix} \begin{bmatrix} \hat{\Omega}_{11}^{(1)} \\ \hat{\Omega}_{12}^{(1)} \\ \hat{\Omega}_{13}^{(1)} \\ \vdots \\ \hat{\Omega}_{1\hat{n}-1}^{(1)} \\ \hat{\Omega}_{1\hat{n}}^{(1)} \end{bmatrix} = \begin{bmatrix} -6/h \\ 6/h \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.12}$$

The solution of above system provides us the required weighting coefficients $\begin{bmatrix} \hat{\Omega}_{11}^{(1)} & \hat{\Omega}_{12}^{(1)} & \cdots & \hat{\Omega}_{1\hat{n}-1}^{(1)} & \hat{\Omega}_{1\hat{n}}^{(1)} \end{bmatrix}^T$, the above tridiagonal system (2.12) can be solved easily by applying the "Thomas Algorithm " [37] in MATLAB. Similarly, we can also find the weighting coefficients for second knot point $w_\lambda = w_2$, the approximation is given as

$$\Phi_p'(w_2) = \sum_{\ell=1}^{\hat{n}} \hat{\Omega}_{2\ell}^{(1)} \Phi(w_\ell), \qquad p = 1, 2, \cdots \hat{n}. \tag{2.13}$$

This will give us the following tridiagonal system as

$$\begin{bmatrix} 6 & 1 & & & & \\ 0 & 4 & 1 & & & \\ & \cdots & \cdots & \cdots & & \\ & & 1 & 4 & 1 & \\ & & & 1 & 4 & 0 \\ & & & & 1 & 6 \\ & & & & & \end{bmatrix} \begin{bmatrix} \hat{\Omega}_{21}^{(1)} \\ \hat{\Omega}_{22}^{(1)} \\ \hat{\Omega}_{23}^{(1)} \\ \vdots \\ \hat{\Omega}_{2\hat{n}-1}^{(1)} \\ \hat{\Omega}_{2\hat{n}}^{(1)} \end{bmatrix} = \begin{bmatrix} -3/h \\ 0 \\ 3/h \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{2.14}$$

By solving the above tridiagonal system (2.14) for $\hat{\Omega}_{2\hat{n}}^{(1)}$, we get the second column of weighting coefficients matrix. The weighting coefficients $\hat{\Omega}_{\lambda\ell}^{(1)}$ for $\lambda = 3, 4, ..., (\hat{n} - 1)$ can be computed in same way. On boundary $w = w_{\hat{n}}$, we have

$$\begin{bmatrix} 6 & 1 & & & & \\ 0 & 4 & 1 & & & \\ & \cdots & \cdots & \cdots & & \\ & & 1 & 4 & 1 & \\ & & & 1 & 4 & 0 \\ & & & & 1 & 6 \\ & & & & & \end{bmatrix} \begin{bmatrix} \hat{\Omega}_{\hat{n}1}^{(1)} \\ \hat{\Omega}_{\hat{n}2}^{(1)} \\ \hat{\Omega}_{\hat{n}3}^{(1)} \\ \vdots \\ \hat{\Omega}_{\hat{n}\hat{n}-1}^{(1)} \\ \hat{\Omega}_{\hat{n}\hat{n}}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -6/h \\ 6/h \end{bmatrix}. \tag{2.15}$$

Hence, by following above procedure for $\lambda = 1, 2, \cdots, \hat{n}$, we can find the weighting coefficients for each w_λ. Further, Shu's recurrence formula [22] or multiplication of matrices approach [22] given in Chapter 1 can be used to determine the weighting coefficients for second and higher order derivatives.

In particular, for $\hat{n} = 7$ we have computed the following matrices of weights, $\hat{\Omega}_{7\times 7}$ for first and second derivative approximations as

$$\hat{\Omega}^{(1)}_{7\times 7} = \begin{bmatrix} -7.6076 & 9.6461 & -2.5846 & 0.6923 & -0.1846 & 0.0461 & -0.0076 \\ -2.7846 & -1.2923 & 5.1692 & -1.3846 & 0.3692 & -0.0923 & 0.0153 \\ 0.7461 & -4.4769 & -0.0923 & 4.8461 & -1.2923 & 0.3230 & -0.0538 \\ -0.2000 & 1.2000 & -4.8000 & 0 & 4.8000 & -1.2000 & 0.2000 \\ 0.0538 & -0.3230 & 1.2923 & -4.8461 & 0.0923 & 4.4769 & -0.7461 \\ -0.0153 & 0.0923 & -0.3692 & 1.3846 & -5.1692 & 1.2923 & 2.7846 \\ 0.0076 & -0.0461 & 0.1846 & -0.6923 & 2.5846 & -9.6461 & 7.6076 \end{bmatrix},$$

$$\hat{\Omega}^{(2)}_{7\times 7} = \begin{bmatrix} 28.9384 & -73.3846 & 66.1846 & -30.1846 & 11.3538 & -3.5999 & 0.6923 \\ 28.9384 & -50.1230 & 7.1999 & 22.9846 & -12.7384 & 4.7076 & -0.9692 \\ 5.6769 & 19.6615 & -50.1230 & 13.0153 & 19.6615 & -10.2461 & 2.3538 \\ -5.1230 & 16.3384 & 13.8461 & -50.1230 & 13.8461 & 16.3384 & -5.1230 \\ 2.3538 & -10.2461 & 19.6615 & 13.0153 & -50.1230 & 19.6615 & 5.6769 \\ -0.9692 & 4.7076 & -12.7384 & 22.9846 & 7.2000 & -50.1230 & 28.9384 \\ 0.6923 & -3.5999 & 11.3538 & -30.1846 & 66.1846 & -73.3846 & 28.9384 \end{bmatrix}.$$

2.5 The procedure of implementation of DQM

In this section, the implementation procedure for linear and non-linear partial differential equations (PDEs) is described. We consider the general form of second order non-linear partial differential equation as

$$\begin{aligned} &u_t(x,t) = F\left(x,\ t,\ u^{\gamma}(x,t),\ u_x(x,t),\ u_{xx}(x,t)\right), \quad (x,t) \in \varsigma \times [0, T^*], \\ &u(x,0) = \psi(x), \quad x \in [a,b], \end{aligned} \tag{2.16}$$

where $\varsigma = [a,b]$ and $\gamma \in \mathbb{Z}$. The first and second order spatial derivative approximations at a point x_q by differential quadrature method is given by

$$\begin{aligned} u_x(x_q,t) &= \sum_{\ell=1}^{\hat{n}} \hat{\Omega}^{(1)}_{q\ell} u(x_\ell,t), \\ u_{xx}(x_q,t) &= \sum_{\ell=1}^{\hat{n}} \hat{\Omega}^{(2)}_{q\ell} u(x_\ell,t), \qquad q = 1,2,\cdots,\hat{n}, \end{aligned} \tag{2.17}$$

where $\hat{n}$ is the total number of grid points. We calculate the weights $\hat{\Omega}_{q\ell}$ by using modified cubic B-splines as discussed in previous section. First we use the differential quadrature approximations (2.17) of spatial derivatives in equation (2.16) to obtain the system of non-linear ordinary differential equations as

$$\frac{du(x_q,t)}{dt} \cong F\left(x_q,\ t,\ u^{\gamma}(x_q,t), \sum_{\ell=1}^{\hat{n}} \hat{\Omega}^{(1)}_{q\ell} u(x_\ell,t), \sum_{\ell=1}^{\hat{n}} \hat{\Omega}^{(2)}_{q\ell} u(x_\ell,t)\right), \quad (x,t) \in \varsigma \times [0,T^*], \tag{2.18}$$

$$u(x_q,0) = \psi(x_q), \qquad q = 1,2,\cdots,\hat{n},$$

where $\varsigma = [a,b]$ and $\gamma \in \mathbb{Z}$. Equation (2.18) is the system of $\hat{n}$ non-linear ordinary differential equations with $\hat{n}$ number of unknowns. The system (2.18) can be solved by using any suitable numerical method. We utilize the Runge Kutta (RK-4) method of order 4 to solve this system. The description of method is given below.

Dividing $[0,T^*]$ into $\hat{m}$ sub-intervals, the step size $h = \frac{T^*}{\hat{m}}$ and $t_k = kh$, $k = 0,1,2,\cdots,\hat{m}$. The procedure of discretization by RK-method for system (2.18) is described as follows [38]

$$u(x_q,t_{k+1}) = u(x_q,t_k) + \frac{1}{6}\left(K_1^q + 2K_2^q + 2K_3^q + K_4^q\right), \tag{2.19}$$

where

$$\begin{aligned}
K_1^q =& h g_q\left(t_k, \{u_i(t_k)\}_{i=1}^{\hat{n}}\right),\\
K_2^q =& h g_q\left(t_k + \frac{h}{2}, \left\{u_i(t_k) + \frac{K_1^i}{2}\right\}_{i=1}^{\hat{n}}\right),\\
K_3^q =& h g_q\left(t_k + \frac{h}{2}, \left\{u_i(t_k) + \frac{K_2^i}{2}\right\}_{i=1}^{\hat{n}}\right),\\
K_4^q =& h g_q\left(t_k + h, \{u_i(t_k) + K_3^i\}_{i=1}^{\hat{n}}\right),
\end{aligned}$$

with the initial conditions

$u_q(t_0) = \psi(x_q), \quad q = 1,2,\ \cdots\ ,\hat{n}.$

2.6 Application of the modified cubic B-splines DQM

For the sake of application of the modified cubic B-splines differential quadrature method, we consider some well known non-linear PDEs such as Fisher type reaction diffusion equation, Fitzhugh-Nagumo equation with time dependent coefficients, parabolic equation and Burgers' equations . These PDEs model the scientific and engineering problems. For example, Fitzhugh-Nagumo equation has many applications in autocatalytic chemical reactions, propagation of flame, neurophysiology, nuclear reactor theory and logistic population growth [39]. The formation structure of Burgers' equation is same as the 1-D Navier-Stoke's equations excluding the stress term. Therefore, Burgers' equation supports us in modeling the various fluid mechanics problems as well as in general transport and wave mechanics. Various numerical techniques have been used to solve these models. We solved these partial differential equations by using modified cubic B-splines DQM and the obtained results along with maximum absolute errors L_∞ are displayed with the help of tables and figures. The numerical results are compared with exact solutions to demonstrate the accuracy and efficiency of proposed scheme.

We discuss a family of non-linear partial differential equations [40]

$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2} + g(u), \qquad x \in (-\infty, \infty), t \geq 0. \tag{2.20}$$

Equation (2.20) is one dimensional Fisher type reaction diffusion equation ($FRDE$), with $\beta > 0$ is a diffusion coefficient and $g(u)$ is non-linear reaction term. For $\beta = 1$, we solve the equation (2.20) with different non-linear reaction terms. For example, setting $g(u) = -u(1-u)(\gamma - u)$, equation (2.20) becomes Fitzhugh-Nagumo equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - u(1-u)(\gamma - u), \tag{2.21}$$

where $0 < \gamma < 1$. In [27] R. Jiwari et. al proposed the polynomial based DQM to determine the numerical solution of equation (2.21). S. Abbasbandy [41] find out the solition solution of equation (2.21) by using the Homotopy analysis method.

If we set $\gamma = -1$ then equation (2.21) will become the Newell Whitehead equation [42].

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(1-u^2). \tag{2.22}$$

Newell Whitehead equation describe the envelop of modulated roll solution in system along with too large unbounded space direction.

2.6.1 Generalized Fitzhugh-Nagumo equations

Example 1. *Consider the non-linear Fitzhugh-Nagumo equation [27]*

$$u_t = u_{xx} - u(1-u)(\rho - u), \qquad (x,t) \in \vartheta \times [0,T], \tag{2.23}$$

with the initial and the boundary conditions

$u(x,0) = \xi(x) := \frac{1+\tanh\left(x/2\sqrt{2}\right)}{2}, \quad x \in [a,b].$

$u(a,t) = \eta_1(t) := \frac{1+\tanh\left(\left(a - \frac{2\rho-1}{\sqrt{2}}t\right)/2\sqrt{2}\right)}{2},$

$u(b,t) = \eta_2(t) := \frac{1+\tanh\left(\left(b - \frac{2\rho-1}{\sqrt{2}}t\right)/2\sqrt{2}\right)}{2},$

where $\vartheta = [a,b]$ and ρ is constant. The exact solution of equation (2.23) is given in [27]. Applying modified cubic B-splines differential quadrature method on equation (2.23), we have the following first order non-linear system of ODEs as

$$\frac{du(x_q,t)}{dt} = \sum_{\ell=2}^{\hat{n}-1} \hat{\Omega}^{(2)}_{q\ell} u(x_\ell,t) - u(x_q,t)\left(1-u(x_q,t)\right)\left(\rho - u(x_q,t)\right) + \left(\hat{\Omega}^{(2)}_{q1}\eta_1 + \hat{\Omega}^{(2)}_{q\hat{n}}\eta_2\right), \tag{2.24}$$

$$u(x_q,0) = \xi(x_q), \qquad q = 2,3,\ \dots\ ,\hat{n}-1.$$

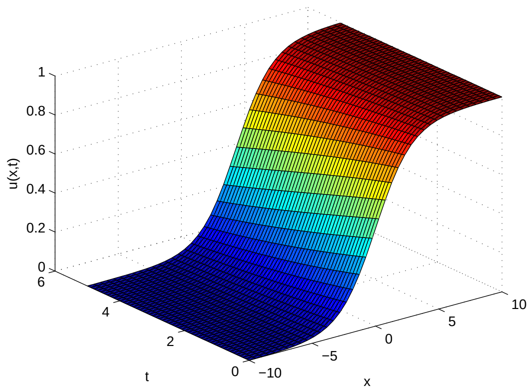
Equation (2.24) is the system of $\hat{n}-2$ equations with same number of unknowns. This system is solved by RK-4 method in MATLAB. Table (2.4) and (2.5) lists the numerical results and Figure (2.1) represents the comparison of solutions and maximum absolute error of equation (2.23) at $a = -b = -10$, $\rho = 3/4,\ -2$. From Table (2.4), (2.5) and Figure (2.1) we can see that the obtained results are very closed to the exact solutions that demonstrates the accuracy of the method.

Example 2. *Consider the non-linear Fitzhugh-Nagumo equation with time dependent coefficients [27]*

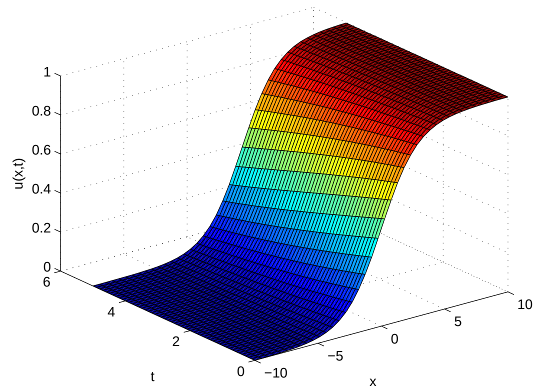
$$u_t = \cos(t)u_{xx} - \cos(t)u_x - 2\cos(t)\left[u(1-u)(\rho-u)\right], \quad (x,t) \in \vartheta \times [0,T], \tag{2.25}$$

with the initial and the boundary conditions

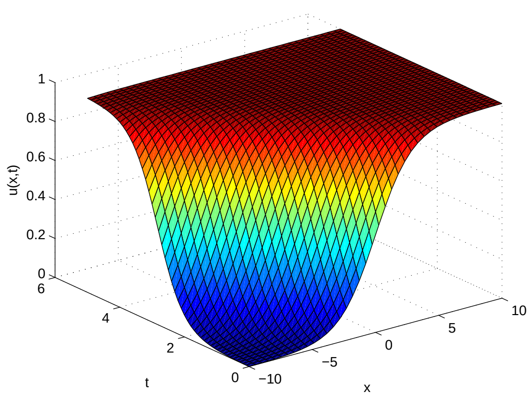
$u(x,0) = \zeta(x) := \frac{\rho + \rho\tanh\left(\frac{x}{2}\rho\right)}{2}, \quad x \in [a,b],$



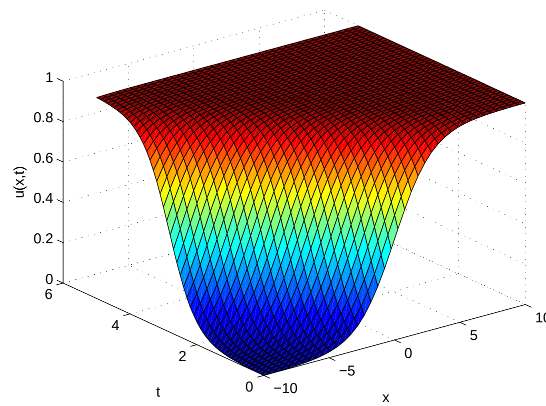
(a) Numerical solution when $\rho = 3/4$



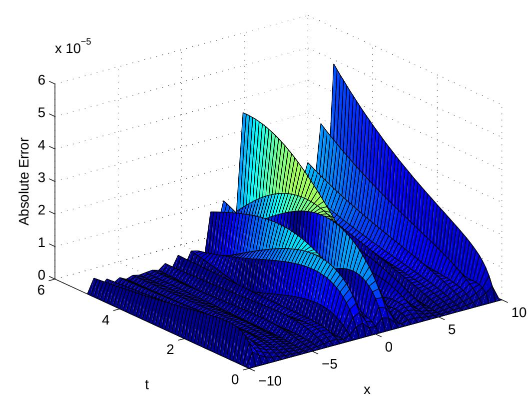
(b) Exact solution when $\rho = 3/4$



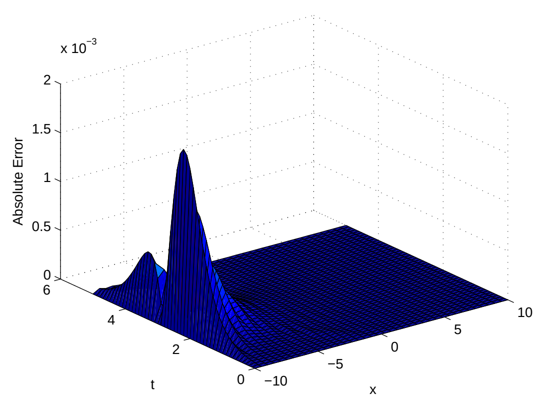
(c) Numerical solution when $\rho = -2$



(d) Exact solution when $\rho = -2$



(e) Absolute error L_∞ when $\rho = 3/4$



(f) Absolute error L_∞ when $\rho = -2$

Figure 2.1: Comperison of numerical and exact solutions of equation (2.23) with absolute error L_∞ for $\Delta t = 0.05,\ \rho = 3/4,\ -2$ upto time t=5.

Table 2.4: Comparison of numerical and exact solutions at $\rho = 0.75$, h=0.05 and $\hat{n} = 40$ for different values of x and t.

x	t	u_{DQM}	u_{Exact}	E_{DQM}
-8.0	0.2	0.0033	0.0033	$1.8914e^{-5}$
	0.4	0.0031	0.0031	$1.8831e^{-5}$
	0.6	0.0030	0.0029	$1.8262e^{-5}$
	0.8	0.0028	0.0028	$1.7476e^{-5}$
1.0	2.3	0.5327	0.5329	$2.7504e^{-5}$
	2.5	0.5203	0.5205	$1.7665e^{-4}$
	2.7	0.5079	0.5080	$7.7467e^{-5}$
	2.9	0.4955	0.4955	$2.2042e^{-5}$
5.0	1.5	0.9588	0.9593	$4.2554e^{-4}$
	2.0	0.9553	0.9541	$4.7304e^{-4}$
	3.0	0.9413	0.9418	$5.7775e^{-4}$
	5.0	0.9068	0.9076	$8.1806e^{-4}$

$u(a,t) = \sigma_1(t) := \frac{\rho+\rho\tanh\left(\frac{1}{2}(a-(3-\rho)\sin(t))\rho\right)}{2}$,
$u(b,t) = \sigma_2(t) := \frac{\rho+\rho\tanh\left(\frac{1}{2}(b-(3-\rho)\sin(t))\rho\right)}{2}$,
where $\vartheta = [a,b]$ and ρ is constant. The exact solution of equation (2.25) is given in [27]. Similarly, by following the procedure of implementation, we apply the modified cubic B-splines differential quadrature method on equation (2.25) to get the reduced system of non-linear ODEs as

$$\frac{du(x_q,t)}{dt} = \cos(t)\left(\sum_{\ell=2}^{\hat{n}-1}\hat{\Omega}^{(2)}_{q\ell}u(x_\ell,t) - \sum_{\ell=2}^{\hat{n}-1}\hat{\Omega}^{(1)}_{q\ell}u(x_\ell,t) - 2\left(u(x_q,t)\left(1-u(x_q,t)\right)\left(\rho-u(x_q,t)\right)\right)\right) + \cos(t)\left(\hat{\Omega}^{(2)}_{q1}\sigma_1(t) + \hat{\Omega}^{(2)}_{q\hat{n}}\sigma_2(t) - \hat{\Omega}^{(1)}_{q1}\sigma_1(t) - \hat{\Omega}^{(1)}_{q\hat{n}}\sigma_2(t)\right), \tag{2.26}$$

with initial conditions

$$u(x_q,0) = \zeta(x_q), \qquad q = 2,3,\ \ldots\ ,\hat{n}-1.$$

Table 2.5: Comparison between numerical and exact solutions with $\rho = -2$, h=0.05 and $\hat{n} = 40$.

x	t	u_{DQM}	u_{Exact}	E_{DQM}
-9.0	0.2	0.00284	0.00283	$1.3188e^{-5}$
	1.5	0.06849	0.06824	$2.4734e^{-4}$
	2.0	0.20406	0.20359	$4.6615e^{-4}$
	3.0	0.75629	0.75695	$6.5612e^{-4}$
1.0	2.0	0.99663	0.99668	$5.3528e^{-5}$
	2.5	0.99903	0.99904	$1.6339e^{-5}$
	3.0	0.99972	0.99972	$5.5109e^{-6}$
	4.0	0.99997	0.99997	$2.0748e^{-7}$
4.0	2.0	0.99959	0.99960	$5.3941e^{-6}$
	2.5	0.99988	0.99988	$1.8429e^{-6}$
	3.0	0.99996	0.99996	$8.9423e^{-7}$
	5.0	0.99999	0.99999	$8.0263e^{-8}$

We solved the system (2.26) by using Runge Kutta (RK-4) method and the computed numerical results with $a = -b = -10$, at $\rho = 3/4$ and $\rho = 1/2$ are shown in Table (2.6) and Figure (2.2). The maximum absolute error is also reported in Table (2.6) at different values of time t with $\rho = 3/4$. Figure (2.2) is interpreting the comparison between numerical and exact solutions with $\rho = 3/4$ and $\rho = 1/2$.

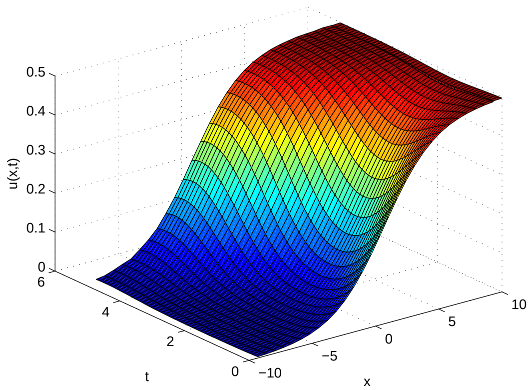
2.6.2 Numerical solutions of Burgers' equation

For $g(u) = -\alpha u_x$, equation (2.20) becomes

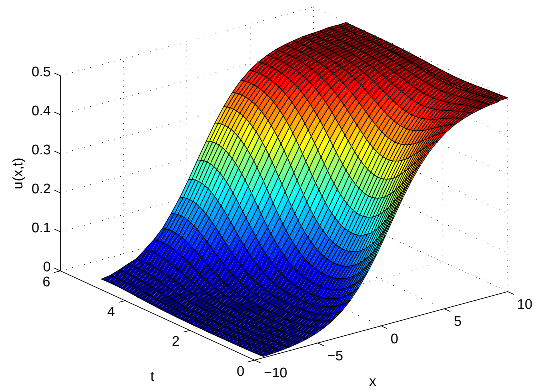
$$\frac{\partial u}{\partial t} = \beta \frac{\partial^2 u}{\partial x^2} - \alpha u \frac{\partial u}{\partial x}, \qquad (x,t) \in D \times [0,T], \tag{2.27}$$

where α and β are arbitrary constants and $D = \{x : 0 < x < 1\}$. Equation (2.27) is well known Burgers' equation [28].

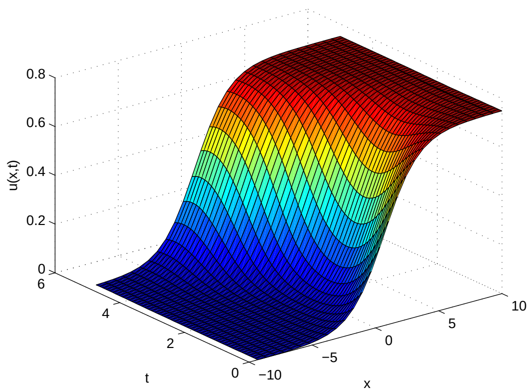
Example 3. *Consider Burgers' equation (2.27) with* $\alpha = \beta = 1$ *over the region* $[0, 1]$, *with the initial and boundary conditions*



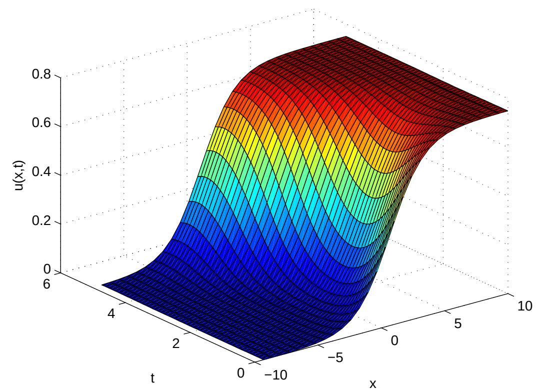
(a) Numerical solution when $\rho = 1/2$



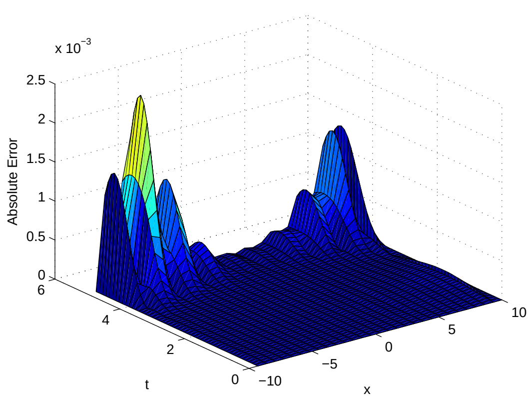
(b) Exact solution when $\rho = 1/2$



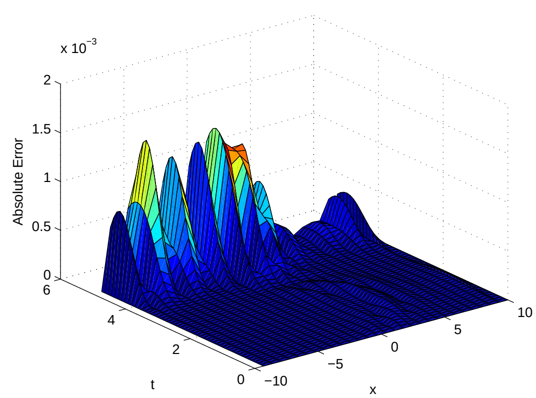
(c) Numerical solution when $\rho = 3/4$



(d) Exact solution when $\rho = 3/4$



(e) Absolute error L_∞ when $\rho = 1/2$



(f) Absolute error L_∞ when $\rho = 3/4$

Figure 2.2: Comparison of the obtained results and exact solutions of problem (3.33) along with absolute error L_∞ for $\Delta t = 0.05,\ \rho = 1/2,\ 3/4$ upto time t=5.

Table 2.6: Max absolute error L_∞ of problem (2.25) at different time t with $\rho = 1/2,\ 3/4$ and $h = 0.05$.

T	$\rho = 0.75$ L_∞	$\rho = 0.50$ L_∞
0.2	$3.0058e^{-5}$	$2.9193e^{-5}$
0.5	$5.4611e^{-5}$	$5.2827e^{-5}$
1.0	$7.8140e^{-5}$	$9.4866e^{-5}$
1.5	$8.1858e^{-5}$	$1.1720e^{-4}$
2.0	$8.2123e^{-5}$	$1.0419e^{-4}$
3.0	$2.2241e^{-5}$	$1.7875e^{-5}$
5.0	$1.1022e^{-3}$	$1.8567e^{-3}$

$$u(x,0) = h(x) := \frac{(c + 2\beta \tanh(x))}{\alpha},$$
$$u(0,t) = g_1(t) := \frac{(c + 2\beta \tanh(-ct))}{\alpha},$$
$$u(1,t) = g_2(t) := \frac{(c + 2\beta \tanh(1-ct))}{\alpha}.$$

The exact solution of equation (2.27) is given in [28]. By applying the MCB-DQ method on equation (2.27), we have following system of ODEs

$$\begin{aligned}\frac{du(x_q,t)}{dt} =& \beta\left(\sum_{\ell=2}^{\hat{n}} \hat{\Omega}^{(2)}_{q\ell} u(x_\ell,t) + \hat{\Omega}^{(2)}_{q1} g_1(t) + \hat{\Omega}^{(2)}_{q\hat{n}} g_2(t)\right) - \alpha u(x_q,t)\left(\sum_{\ell=2}^{\hat{n}} \hat{\Omega}^{(1)}_{ql} u(x_\ell,t)\right) \\ & - \alpha u(x_q,t)\left(\hat{\Omega}^{(1)}_{q1} g_1(t) + \hat{\Omega}^{(1)}_{q\hat{n}} g_2(t)\right),\end{aligned} \tag{2.28}$$

with initial conditions

$$u(x_q,0) = h(x_q), \qquad q = 2,3,\ \dots\ ,\hat{n}-1.$$

The system (2.28) is solved by RK-4 method and the obtained results with maximum absolute error L_∞ are listed in Table (2.7) at different values of α, β and time t, with $\hat{n} = 21$, $c = 0.1$ and $\Delta t = 0.01$.

Example 4. *In this example, problem (2.27) is solved with different initial and boundary conditions as*

Table 2.7: Numerical results of equation (2.27) at different values of α, β and time t by present method.

		$\|E(u)\|_\infty$		
α	β	$t = 0.1$	$t = .25$	$t = 0.5$
1	0.01	$3.20e^{-5}$	$7.77e^{-5}$	$1.53e^{-4}$
	0.001	$4.49e^{-7}$	$9.06e^{-7}$	$1.67e^{-6}$
	0.0001	$2.07e^{-8}$	$2.37e^{-8}$	$3.03e^{-8}$
	0.0001	$1.99e^{-9}$	$1.99e^{-9}$	$2.01e^{-9}$
0.1	0.01	$3.20e^{-4}$	$7.77e^{-4}$	$1.53e^{-4}$
	0.001	$4.49e^{-6}$	$9.06e^{-6}$	$1.67e^{-5}$
	0.0001	$2.07e^{-7}$	$2.37e^{-7}$	$3.03e^{-7}$
	0.0001	$1.99e^{-8}$	$1.99e^{-8}$	$2.01e^{-8}$

$$u(x,0) = 2x, \quad x \in [0,1],$$
$$u(0,t) = 0, \text{ and } u(1,t) = 2/1 + 2t, \quad t \in [0,T].$$

The exact solution of equation (2.27) is given in [43]. The obtained numerical results are displayed in Table (2.8) for different values of x and t with $\hat{n} = 5$ and $\Delta t = 0.002$. Table (2.8) lists the comparison of solutions of Example (4) along with the maximum absolute error L_∞.

2.6.3 Numerical solutions of non-linear generalized Burger-Fisher equation

When we set $g(u) = \delta u^\gamma u_x + \rho u(u^\gamma - 1)$ and $\beta = 1$, the equation (2.20) becomes the generalized non-linear Burger-Fisher equation [44] as

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + \delta u^\gamma \frac{\partial u}{\partial x} + \rho u(u^\gamma - 1) = 0, \quad (x,t) \in \vartheta \times [0, T^*]. \tag{2.29}$$

Table 2.8: Comparison of solutions by present method and exact solutions for $h = 0.002$ and $\hat{n} = 5$.

x	t	u_{DQM}	u_{Exact}	E_{DQM}
0.01	0.2	0.1428571	0.1428571	$5.2096e^{-11}$
	0.5	0.0999999	0.1000000	$1.5280e^{-11}$
	0.8	0.0769230	0.0769230	$6.8731e^{-12}$
0.3	0.3	0.3749999	0.3750000	$1.0266e^{-11}$
	0.7	0.2499999	0.2500000	$2.9664e^{-11}$
	0.9	0.2142857	0.2142857	$1.8838e^{-11}$
0.7	0.3	0.8750000	0.8750000	$2.5090e^{-10}$
	0.6	0.6363636	0.6363636	$8.9339e^{-11}$
	0.9	0.5000000	0.5000000	$4.0757e^{-11}$
0.9	0.5	0.90000000	0.9000000	$7.1191e^{-11}$
	0.7	0.7500000	0.7500000	$3.9697e^{-11}$
	0.9	0.6428571	0.6428571	$2.4301e^{-11}$

A significant amount of research work has been done for the analysis of Burgers' and Burger-Fisher equations. A number of techniques have been followed by different mathematicians and physicists to solve Burger-Fisher equations. D. Kaya and E. L Sayed [45] studied the explicit solutions of Burger-Fisher equation. To solve this equation Wazwaz [46] manifested the tanh method. Somalian [47] presented the variational iterative method to find the solutions of Burger-Fisher equation. We also have solved some examples of Burger-Fisher equation with different values of parameters by MCB-DQ method.

Example 5. *Consider the generalized form of Burger-Fisher equation (2.29), with the following initial and boundary conditions*

$$u(x,0)=h(x)=\left(\frac{1}{2}-\frac{1}{2}\tanh\left(\frac{\delta\gamma x}{2(1+\rho)}\right)\right)^{\gamma^{-1}},\quad 0\le x\le 1,$$

$$u(0,t)=\xi_1(t)=\left(\frac{1}{2}-\frac{1}{2}\tanh\left(\frac{\delta\gamma}{2(1+\rho)}\left[-\left(\frac{\delta^2+\rho(1+\gamma)^2}{\delta(1+\gamma)}\right)t\right]\right)\right)^{\gamma^{-1}},$$

$$u(1,t)=\xi_2(t)=\left(\frac{1}{2}-\frac{1}{2}\tanh\left(\frac{\delta\gamma}{2(1+\rho)}\left[1-\left(\frac{\delta^2+\rho(1+\gamma)^2}{\delta(1+\gamma)}\right)t\right]\right)\right)^{\gamma^{-1}}.$$

where α, δ and γ are the constants. The exact solution of equation (2.29) is given in [44]. By applying the MCB-DQ method, we have the following system of non-linear ODEs as

$$\begin{aligned}\frac{du(x_q,t)}{dt}=&\sum_{\ell=2}^{\hat{n}-1}\hat{\Omega}^{(2)}_{q\ell}u(x_\ell,t)+\delta u^{\gamma}(x_q,t)\left[\sum_{\ell=2}^{\hat{n}-1}\hat{\Omega}^{(1)}_{q\ell}u(x_\ell,t)+\hat{\Omega}^{(1)}_{q1}\xi_1(t)+\hat{\Omega}^{(1)}_{q\hat{n}}\xi_2(t)\right]+\\ &\delta u(x_q,t)\left(u^{\gamma}(x_q,t)-1\right)+\left(\hat{\Omega}^{(2)}_{q1}\xi_1(t)+\hat{\Omega}^{(2)}_{q\hat{n}}\xi_2(t)\right),\end{aligned} \tag{2.30}$$

$$u(x_q,0)=h(x_q),\qquad q=2,3,\ ...,\ \hat{n}-1.$$

The obtained system (2.30) is solved by RK-4 method and the computed results are displayed in Table (2.9) and (2.10) for different values of α, γ and δ.

Table 2.9: Numerical results of equation (2.29) for $\gamma = 1$ and $\delta = \rho$ at different times t.

t	$\|E(u)\|_\infty$				
	$\delta=\rho=0.001$	$\delta=\rho=0.01$	$\delta=\rho=0.1$	$\delta=\rho=0.5$	$\delta=\rho=1.0$
0.2	$3.6637e^{-14}$	$3.6422e^{-11}$	$3.6207e^{-8}$	$4.3946e^{-6}$	$3.3677e^{-14}$
0.4	$3.7747e^{-14}$	$3.8382e^{-11}$	$3.8860e^{-8}$	$5.0658e^{-6}$	$4.0978e^{-5}$
0.6	$4.9515e^{-14}$	$5.0171e^{-11}$	$5.0694e^{-8}$	$6.4112e^{-6}$	$4.6139e^{-5}$
0.8	$6.1395e^{-14}$	$6.2064e^{-11}$	$6.2578e^{-8}$	$7.5903e^{-6}$	$6.6005e^{-5}$
1.0	$7.3274e^{-14}$	$7.3971e^{-11}$	$7.4420e^{-11}$	$5.5740e^{-6}$	$4.1786e^{-5}$

Table 2.10: Numerical results of equation (2.29) for $h = 0.005$, $\hat{n} = 7$, $\gamma = 2$ and $\delta = \rho = 1$.

x	t	u_{DQM}	u_{Exact}	E_{DQM}
0.1	0.2	0.77019	0.77028	$9.0827e^{-5}$
	0.4	0.83338	0.83349	$1.0217e^{-4}$
	0.6	0.88317	0.88327	$9.6179e^{-5}$
	0.8	0.92023	0.92031	$8.0058e^{-5}$
0.4	0.1	0.69914	0.69920	$6.1122e^{-5}$
	0.3	0.77365	0.77374	$9.1177e^{-5}$
	0.6	0.86243	0.86253	$1.0084e^{-4}$
	0.9	0.92178	0.92186	$7.9749e^{-5}$
0.8	0.2	0.69118	0.69122	$3.9896e^{-5}$
	0.5	0.80028	0.80034	$6.5851e^{-5}$
	0.7	0.85749	0.85755	$6.7374e^{-5}$
	0.9	0.90135	0.90141	$5.9434e^{-5}$

Chapter 3

A Modified Method for Solving Non-linear Time and Space Fractional Partial Differential Equations

In this chapter, we extend the differential quadrature method (DQM) for solving time and space fractional non-linear PDEs on a semi infinite domain. Fractional order partial and ordinary differential equations are extensively used in various fields of engineering and sciences. During some last decades a number of applications of fractional calculus have been appeared, specially in the biological disease problems and bioengineering [1, 2], chemical reactions, underwater flow problems in fluid dynamics [3], signal processing [4], viscoelasticity theory [5, 6], fractional quantum mechanics, robotics and electronics [7, 8, 9] etc. Several methods have been developed and many approaches are utilized to obtain the solutions of these fractional differential equations.

In the present work, we have proposed a method based on differential quadrature method (DQM) for both time and space fractional PDEs. The Lagrange interpolation polynomials are used as test functions, but the fractional derivative of Lagrange polynomials is a big hurdle in classical DQM. To overcome this problem, we represent the Lagrange polynomials in terms of shifted Legendre polynomials by constructing a transformation matrix which transforms the Lagrange polynomials into shifted Legendre polynomials of arbitrary order. Transformation matrices are also used to represent the Caputo fractional derivative of Lagrange polynomials in terms of Caputo fractional derivative of shifted Legendre polynomials. We obtain the new weighting coefficients matrices for space fractional derivatives by shifted Legendre polynomials and use these in conversion of a non-linear fractional PDEs into a system of non-linear fractional ODEs. For the system of non linear time fractional ODEs, we utilize the Adam Bashforth Moulton predictor-corrector approach [48]. Comparison analysis of the proposed method with some well known methods is also provided. We have also discussed the convergence analysis for the proposed method. Many engineers can utilize the present technique for solving their non-linear time and space fractional models.

3.1 Preliminaries

Our main focus is to derive the weighting coefficients matrices for the approximations of fractional derivatives. Some basic definitions of fractional integrals and differential operators are used in the derivation of weighting coefficient matrices which are as follows [49]

Riemann Liouville differential and integral operator of order α

Let q be an integer such that $q-1<\alpha<q$ and $h \in AC^q[a_1,a_2]$. The left and right Riemann Liouville fractional order, α, derivative is

$$ {}_xD^{\alpha}_{a_1}h(x) = \frac{1}{\Gamma(q-\alpha)}\frac{d^q}{dx^q}\int\limits_{a_1}^{x}(x-\varsigma)^{q-\alpha-1}h(\varsigma)d\varsigma, \tag{3.1}$$

$$ {}_xD^{\alpha}_{a_2}h(x) = \frac{(-1)^q}{\Gamma(q-\alpha)}\frac{d^q}{dx^q}\int\limits_{x}^{a_2}(\varsigma-x)^{q-\alpha-1}h(\varsigma)d\varsigma, \tag{3.2}$$

where $a_1 \leq x \leq a_2$, $q=\lceil\alpha\rceil$ and $h \in AC^q[a_1,a_2]$. The left and right Riemann Liouville fractional integral operator are

$$ {}_xD^{-\alpha}_{a_1}h(x) = \frac{1}{\Gamma(\alpha)}\int\limits_{a_1}^{x}(x-\varsigma)^{\alpha-1}h(\varsigma)d\zeta, \tag{3.3}$$

$$ {}_xD^{-\alpha}_{a_2}hx) = \frac{1}{\Gamma(\alpha)}\int\limits_{x}^{a_2}(\varsigma-x)^{\alpha-1}h(\varsigma)d\varsigma, \tag{3.4}$$

where $a_1 \leq x \leq a_2$, $q = \lceil \alpha \rceil$ and $\alpha \in \mathbb{R}^+$.

Caputo differential operator of order α

The left and right Caputo fractional differential operator are

$$ {}_x^C D_{a_1}^{\alpha} h(x) = \frac{1}{\Gamma(q-\alpha)} \int_{a_1}^{x} (x-\varsigma)^{q-\alpha-1} h^{(q)}(\varsigma) d\varsigma, \tag{3.5} $$

$$ {}_x^C D_{a_2}^{\alpha} h(x) = \frac{(-1)^q}{\Gamma(q-\alpha)} \int_{x}^{a_2} (\varsigma-x)^{q-\alpha-1} h^{(q)}(\varsigma) d\varsigma, \tag{3.6} $$

where $a_1 \leq x \leq a_2$, $q = \lceil \alpha \rceil$ and $\alpha \in \mathbb{R}^+$.

3.2 Development of the proposed method

We propose a numerical scheme for solving non-linear fractional order PDEs. Consider the following general form of non-linear fractional PDE as

$$ \begin{aligned} & {}_tD^{\alpha} u(x,t) = F\left(x,\ t,\ u^{\gamma}(x,t),\ {}_xD^{\beta_1} u(x,t),\ {}_xD^{\beta_2} u(x,t)\right), \quad (x,t) \in \vartheta \times [0, T^*], \\ & u(x,0) = \psi(x), \end{aligned} \tag{3.7} $$

where $\vartheta = [a,b]$, $0 < \alpha, \beta_1 \leq 1$, $0 < \beta_2 \leq 2$ and $\gamma \in \mathbb{Z}$. Assuming a function u such that it satisfies the equation (3.7), and is sufficiently smooth that it allows us to write the following approximation

$$ {}_xD^{\beta} u(x_p, t) \cong \sum_{q=1}^{\hat{n}} \hat{\Omega}_{pq}^{(\beta)} u(x_q, t), \qquad p = 1, 2, 3, ..., \hat{n}, \tag{3.8} $$

where $\hat{\Omega}_{pq}^{(\beta)}$ are the weights for β order derivative approximation. Our main target is to compute these coefficients. First we construct the transformation matrices which transform the Lagrange polynomials into shifted Legendre polynomials and then with the help of these matrices we compute the weighting coefficients.

3.2.1 Transformation matrices

The Lagrange interpolation of a function u is given by

$$u(x,t) = \sum_{q=1}^{\hat{n}} u(x_q,t) l_q(x), \tag{3.9}$$

where

$$l_q(x) = \prod_{l=0, l\neq q}^{\hat{n}-1} \frac{x - x_l}{x_q - x_l}. \tag{3.10}$$

The fractional derivatives of $l_q(x)$ are very difficult to calculate, so we represent the Lagrange polynomials in terms of shifted Legendre polynomials. The transformation matrices are constructed for this purpose. The Lagrange polynomials $l_q(x)$ can be expressed by shifted Legendre polynomials as

$$l_q(x) = \sum_{p=0}^{\hat{n}-1} C_{qp} \hat{H}_p(x), \qquad q = 0, 1, 2, ..., \hat{n}-1, \tag{3.11}$$

where

$$\hat{H}_p(x) = \sum_{r=0}^{p} (-1)^{r+p} \frac{\Gamma(r+p+1)}{\Gamma(r-p+1)\ (p!)^2}\ x^r. \tag{3.12}$$

In vector form

$$\mathbf{L} = \mathbf{C}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}}\ \hat{\mathbf{H}},$$

where

$$\mathbf{L} = \left[\ l_0(x),\ \ l_1(x),\ \ l_2(x),\ \ \cdots\ \ l_{\hat{n}-2}(x),\ \ l_{\hat{n}-1}(x)\ \right]^T,$$

$$\hat{\mathbf{H}} = \left[\ \mho_0(x),\ \ \mho_1(x),\ \ \mho_2(x),\ \ \cdots\ \ \mho_{\hat{n}-2}(x),\ \ \mho_{\hat{n}-1}(x)\ \right]^T,$$

and

$$\mathbf{C}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}} = \begin{bmatrix} \varsigma_{0,0} & \varsigma_{0,1} & \varsigma_{0,2} & \cdots & \varsigma_{0,\hat{n}-2} & \varsigma_{0,\hat{n}-1} \\ \varsigma_{1,0} & \varsigma_{1,1} & \varsigma_{1,2} & \cdots & \varsigma_{1,\hat{n}-2} & \varsigma_{1,\hat{n}-1} \\ \varsigma_{2,0} & \varsigma_{2,1} & \varsigma_{2,2} & \cdots & \varsigma_{2,\hat{n}-2} & \varsigma_{2,\hat{n}-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \varsigma_{\hat{n}-2,0} & \varsigma_{\hat{n}-2,1} & \varsigma_{\hat{n}-2,2} & \cdots & \varsigma_{\hat{n}-2,\hat{n}-2} & \varsigma_{\hat{n}-2,\hat{n}-1} \\ \varsigma_{\hat{n}-1,0} & \varsigma_{\hat{n}-1,1} & \varsigma_{\hat{n}-1,2} & \cdots & \varsigma_{\hat{n}-1,\hat{n}-2} & \varsigma_{\hat{n}-1,\hat{n}-1} \end{bmatrix}.$$

For the present method, the collocation points are taken as $x_k = \dfrac{k-1}{\hat{n}-1}$, where $k = 1, 2, \ ..., \ \hat{n}$. Using these points in (3.11), we have

$$\mathbf{L}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}} = \mathbf{C}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}} \ \hat{\mathbf{H}}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}}, \tag{3.13}$$

where

$$\mathbf{L}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}} = \begin{bmatrix} l_0(x_1) & l_0(x_2) & l_0(x_3) & \cdots & l_0(x_{\hat{n}-1}) & l_0(x_{\hat{n}}) \\ l_1(x_1) & l_1(x_2) & l_1(x_3) & \cdots & l_1(x_{\hat{n}-1}) & l_1(x_{\hat{n}}) \\ l_2(x_1) & l_2(x_2) & l_2(x_3) & \cdots & l_2(x_{\hat{n}-1}) & l_2(x_{\hat{n}}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{\hat{n}-1}(x_1) & l_{\hat{n}-1}(x_2) & l_{\hat{n}-1}(x_3) & \cdots & l_{\hat{n}-1}(x_{\hat{n}-1}) & l_{\hat{n}-1}(x_{\hat{n}}) \end{bmatrix},$$

and

$$\hat{\mathbf{H}}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}} = \begin{bmatrix} \mho_0(x_1) & \mho_0(x_2) & \mho_0(x_3) & \cdots & \mho_0(x_{\hat{n}-1}) & \mho_0(x_{\hat{n}}) \\ \mho_1(x_1) & \mho_1(x_2) & \mho_1(x_3) & \cdots & \mho_1(x_{\hat{n}-1}) & \mho_1(x_{\hat{n}}) \\ \mho_2(x_1) & \mho_2(x_2) & \mho_2(x_3) & \cdots & \mho_2(x_{\hat{n}-1}) & \mho_2(x_{\hat{n}}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mho_{\hat{n}-2}(x_1) & \mho_{\hat{n}-2}(x_2) & \mho_{\hat{n}-2}(x_3) & \cdots & \mho_{\hat{n}-2}(x_{\hat{n}-1}) & \mho_{\hat{n}-2}(x_{\hat{n}}) \\ \mho_{\hat{n}-1}(x_1) & \mho_{\hat{n}-1}(x_2) & \mho_{\hat{n}-1}(x_3) & \cdots & \mho_{\hat{n}-1}(x_{\hat{n}-1}) & \mho_{\hat{n}-1}(x_{\hat{n}}) \end{bmatrix}.$$

We can calculate the matrices $\mathbf{L}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}}$ and $\hat{\mathbf{H}}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}}$ from equations (3.10) and (3.12), we get the transformation matrices $C_{\hat{n}\times\hat{n}}$ as $C_{\hat{n}\times\hat{n}} = L_{\hat{n}\times\hat{n}}(\hat{H}_{\hat{n}\times\hat{n}})^{-1}$. In particular for $\hat{n} = 7$ and $x_\ell = \dfrac{\ell-1}{\hat{n}-1}$, $\ell = 1, 2, \cdots, \ \hat{n}$, we have the following transformation matrix $\mathbf{C}$ as

$$\mathbf{C}_{\mathbf{7}\times\mathbf{7}} = \begin{bmatrix} 0.0488 & -0.1464 & 0.1940 & -0.2249 & 0.1870 & -0.1280 & 0.0701 \\ 0.2571 & -0.5142 & 0.5142 & 0 & 0.3506 & 0.5142 & -0.4207 \\ 0.0321 & -0.0321 & -0.8035 & 0.6750 & -0.2805 & -0.6428 & 1.0519 \\ 0.3238 & 0 & 0.1904 & 0 & 0.8883 & 0 & -1.4025 \\ 0.0321 & 0.0321 & -0.8035 & 0.6749 & -0.2805 & 0.6428 & 1.0519 \\ 0.2571 & 0.5142 & 0.5142 & 0 & -0.3506 & -0.5142 & -0.4207 \\ 0.0488 & 0.1464 & 0.1940 & 0.2250 & 0.1870 & 0.1285 & 0.0701 \end{bmatrix}.$$

3.2.2 Weighting coefficient matrices for fractional derivatives

Applying the Caputo fractional differential operator on equation (3.9), we have

$${}_xD_0^\beta u(x,t) = \sum_{q=1}^{\hat{n}} u(x_q,t) \ {}_xD_0^\beta l_q(x). \tag{3.14}$$

By using equation (3.11), we get

$$ {}_xD_0^{\beta} u(x,t) = \sum_{q=1}^{\hat{n}} u(x_q,t)\ {}_xD_0^{\beta}\left[\sum_{p=0}^{\hat{n}-1} C_{qp}\hat{H}_p(x)\right], \tag{3.15} $$

or

$$ {}_xD_0^{\beta}\ u(x,t) = \sum_{q=1}^{\hat{n}}\left[\sum_{p=0}^{\hat{n}-1} C_{qp}\ {}_xD_0^{\beta}\hat{H}_p(x)\right] u(x_q,t). \tag{3.16} $$

From equation (3.8) and (3.16), we can write the expression for weighting coefficients matrices $\hat{\boldsymbol{\Omega}}^{\beta}_{\hat{n}\times\hat{n}}$ as

$$ \hat{\Omega}^{\beta}_{qk} = \sum_{p=0}^{\hat{n}-1} C_{qp}\ {}_xD_0^{\beta}\hat{H}_p(x_k), \qquad k = 1,2,3,\cdots,\hat{n}, \tag{3.17} $$

where

$$ {}_xD_0^{\beta}\ \hat{H}_p(x) = \sum_{r=0}^{p}(-1)^{r+p}\frac{\Gamma(r+p+1)\ \Gamma(r+1)}{\Gamma(r-p+1)\ \Gamma(r+1-\beta)\ (p!)^2}\ x^{r-\beta}. $$

In particular, for $\hat{n}=7$ and the collocation points $x_\ell = \dfrac{\ell-1}{\hat{n}-1}$, $\ell = 1,2,\cdots,\ \hat{n}$, we have the following matrices of weighting coefficients for different values of β as

$$ \hat{\boldsymbol{\Omega}}^{0.87}_{7\times 7} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.6918 & -4.3451 & 10.6364 & -7.3527 & 3.7299 & -1.1277 & 0.1511 \\ -0.2749 & -2.4291 & -0.9675 & 5.1855 & -1.9772 & 0.5296 & -0.0663 \\ -0.3468 & 0.4298 & -3.7442 & 1.2917 & 2.8974 & -0.5947 & 0.0668 \\ -0.1453 & -0.6235 & 1.8742 & -6.1157 & 3.6478 & 1.4889 & -0.1263 \\ -0.2982 & 0.8976 & -3.2509 & 6.2756 & -10.5268 & 6.2977 & 0.6049 \\ 0.4597 & -4.3969 & 13.6196 & -24.4387 & 27.4568 & -23.2333 & 10.5327 \end{bmatrix}, $$

$$ \hat{\boldsymbol{\Omega}}^{1.785}_{7\times 7} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 33.7604 & -77.7742 & 58.8352 & -20.9247 & 7.8848 & -2.0275 & 0.2460 \\ 9.3751 & 3.7565 & -35.1861 & 19.9513 & 4.1619 & -2.4661 & 0.4074 \\ 6.6003 & -12.9015 & 38.4270 & -70.8470 & 45.9116 & -8.0845 & 0.8941 \\ 5.6870 & -14.7665 & 26.3135 & -9.6999 & -30.3347 & 24.2257 & -1.4251 \\ 2.5293 & 3.1640 & -27.7276 & 62.1810 & -51.8979 & -2.8330 & -14.5841 \\ 17.9737 & -110.2912 & 324.0292 & -550.2484 & 589.2497 & -360.2572 & 99.5442 \end{bmatrix}. $$

3.3 The procedure of implementation

In this section, we describe the implementation of method for non-linear space and time fractional partial differential equations. The approximation of space fractional derivative of u by DQ method is given as

$$ {}_xD^{\beta}u(x_p,t) \cong \sum_{q=1}^{\hat{n}} \hat{\Omega}_{pq}^{(\beta)} u(x_q,t), \qquad p=1,2,\cdots,\hat{n}, \tag{3.18}$$

where $\hat{n}$ are the collocation points taken as $x_\ell = \dfrac{\ell-1}{\hat{n}-1}$, $\ell = 1,2,\cdots,\hat{n}$, in the interval $[a,b]$. The weighting coefficients $\hat{\Omega}_{pq}^{(\beta)}$ can be obtained easily from equation (3.17). The first step of the proposed method is to use equation (3.18) in equation (3.7) to obtain the system of α^{th} order ODEs as

$$ {}_tD^{\alpha}u(x_p,t) \cong F\left(x_p,\ t,\ u^{\gamma}(x_p,t), \sum_{q=1}^{\hat{n}} \hat{\Omega}_{pq}^{(\beta_1)} u(x_q,t), \sum_{q=1}^{\hat{n}} \hat{\Omega}_{pq}^{(\beta_2)} u(x_q,t)\right), \quad (x,t) \in \vartheta \times [0,T^*], \tag{3.19}$$

$$ u(x_p,0) = \psi(x_p), \qquad p=1,2,\cdots,\hat{n},$$

where $\vartheta = [a,b]$, $0 < \alpha, \beta_1 \leq 1$, $1 < \beta_2 \leq 2$ and $\gamma \in \mathbb{Z}$. Equation (3.19) is the system of $\hat{n}$ number of non-linear fractional order, α, ODEs with $\hat{n}$ number of unknowns, $\{u(x_i,t)\}_{i=1}^{\hat{n}}$. The second step is to use the suitable method to solve fractional order, α, system (3.19). In the present work, we utilize both the Runge Kutta (RK-4) and Adam Bashforth Moulton predictor-corrector methods when $\alpha = 1$ and $0 < \alpha < 1$, respectively.

In case when $0 < \alpha < 1$, we will utilize the Adam Bashforth Moulton predictor-corrector method [50] for solving the system of fractional ODEs. This technique is followed by two basic steps, first is predictor in which a roughly approximation of the required quantity is predicted by an explicit method and then the predicted value is refined by the corrector step, which is an implicit method. Dividing $[0,T^*]$ into $\hat{m}$ sub-intervals, the step size $h = \dfrac{T^*}{\hat{m}}$ and $t_k = kh$, $k = 0,1,2,\cdots,\hat{m}$. The procedure of implementation of the Adam Bashforth Moulton predictor-corrector method for the system (3.19) is as follows

Predictor value

The predictor values for fractional system of ODEs (3.19) are

$$u^P(t_{n+1},x_j) = u_0^{(j)} + \frac{1}{\Gamma(\alpha)}\sum_{i=0}^{n} b_{i,n+1}F\left(x_j,\ t_i,\ u(x_j,t_i), \sum_{k=1}^{\hat{n}}\hat{\Omega}_{jk}^{(\beta_1)}u(t_i,x_j), \sum_{k=1}^{\hat{n}}\hat{\Omega}_{jk}^{(\beta_2)}u(t_i,x_j)\right),$$

where

$b_{i,n+1} = \frac{h^\alpha}{\alpha}\left((n-i+1)^\alpha - (n-i)^\alpha\right)$.

Corrector value

With the help of above predictor values, the corrector values are obtained as

$$\begin{aligned} &u(t_{n+1},x_j) \\ &= u_0^{(j)} + \frac{h^\alpha}{\Gamma(\alpha+2)}F\left(t_{n+1},x_j,u^P(t_{n+1},x_j), \sum_{k=1}^{\hat{n}}\hat{\Omega}_{jk}^{(\beta_1)}u^P(t_{n+1},x_k), \sum_{k=1}^{\hat{n}}\hat{\Omega}_{jk}^{(\beta_2)}u^P(t_{n+1},x_k)\right) \\ &\quad + \frac{h^\alpha}{\Gamma(\alpha+2)}\sum_{k=0}^{n} a_{k,n+1}F\left(t_{n+1},x_j,u^P(t_{n+1},x_j), \sum_{k=1}^{\hat{n}}\hat{\Omega}_{jk}^{(\beta_1)}u^P(t_{n+1},x_k), \sum_{k=1}^{\hat{n}}\hat{\Omega}_{jk}^{(\beta_2)}u^P(t_{n+1},x_k)\right), \end{aligned}$$

where

$$a_{k,n+1} = \begin{cases} n^{\alpha+1} + (n+1)^\alpha(\alpha - n), & if\ k=0, \\ (n-k)^{(\alpha+1)} + (n+2-k)^{(\alpha+1)} - 2(n+1-k)^{(\alpha+1)}, & if\ 1 \le k \le n. \end{cases}$$

From the mathematical analysis of this technique in [48], the order of convergence is $K = min(2, \alpha+1)$.

3.4 Convergence analysis

In this section, we explain the truncation error bound for any arbitrary fractional order derivative of an unknown function $g(x)$ by using present method.

Theorem 3.4.1. *[43] Consider $\{x_p\}_{p=1}^{\hat{n}}$ are distinct $\hat{n}$ nodes in an interval $[a,b]$ and suppose a one dimensional continuous function $g(x)$ such that $g \in C^{\hat{n}}[a,b]$. Then for* *the fractional derivative of* $g(x)$, ${}_xD^{\alpha}g(x)$, $i-1<\alpha<i$, $\;i=1,2$, *the truncation error* T_i *is given by*

$$|T_i| \le \frac{1}{(\hat{n}-2+i)!}e_i,$$

where $\hat{n}=1,2,3,\ \ldots,\ \hat{n}$ are the grid points and e_i are positive constants for $i=1,2$. Inequality shows that the absolute error approaches to zero when $\hat{n}\to\infty$.

Proof. The Lagrange interpolation of function g is given by

$$g(x)=\sum_{j=1}^{\hat{n}} l_i(x)g(x_j)+\frac{g^{\hat{n}}(\eta(x))}{\hat{n}!}\prod_{i=1}^{\hat{n}}(x-x_i), \tag{3.20}$$

where $\eta\in[a,b]$, and $l_j(x)=\prod\limits_{p=0,p\neq j}^{\hat{n}}\frac{(x-x_p)}{(x_j-x_p)}$.

Taking fractional derivative of (3.20) and using DQ method, we have

$$\frac{\partial^{\alpha}g(x_k)}{\partial x^{\alpha}}=\sum_{j=1}^{\hat{n}} a_{jk}^{(\alpha)}g(x_j)+T_i, \tag{3.21}$$

with $T_i=\frac{1}{\hat{n}!}\frac{\partial^{\alpha}}{\partial x^{\alpha}}\left(g^{\hat{n}}(\eta(x))q(x)\right)_{x=x_k}$, the truncation error and $q(x)=\prod\limits_{i=1}^{\hat{n}}(x-x_i)$. From Leibnitz formula [48], we have

$$ {}_xD_a^{\alpha}[gf](x)=\sum_{r=0}^{\lfloor\alpha\rfloor}\binom{\alpha}{r}\left(D_a^r g\right)(x)\left(D_a^{\alpha-r}f\right)(x)+\sum_{r=\lfloor\alpha\rfloor+1}^{\infty}\binom{\alpha}{r}\left(D_a^r g\right)(x)\left(I_a^{r-\alpha}f\right)(x). \tag{3.22}$$

In truncation error T_i for $i=1$, we get $0<\alpha\le 1$, and

$$T_1=\frac{1}{\hat{n}!}\left[\binom{\alpha}{0}q(x)|_{x=x_k}\left.\frac{\partial^{\alpha}g^{\hat{n}}(\eta(x))}{\partial x^{\alpha}}\right|_{x=x_k}+\sum_{r=1}^{\infty}\binom{\alpha}{r}\left.\frac{d^r}{dx^r}q(x)\right|_{x=x_k}J^{r-\alpha}g^{\hat{n}}(\eta(x))\big|_{x=x_k}\right]. \tag{3.23}$$

Also, for $i=2$, we have $1<\alpha\le 2$, and

$$\begin{aligned}T_2=&\frac{1}{\hat{n}!}\left[\binom{\alpha}{0}q(x)|_{x=x_k}\left.\frac{\partial^{\alpha}g^{\hat{n}}(\eta(x))}{\partial x^{\alpha}}\right|_{x=x_k}+\binom{\alpha}{1}\left.\frac{d}{dx}q(x)\right|_{x=x_k}\left.\frac{\partial^{\alpha-1}g^{\hat{n}}(\eta(x))}{\partial x^{\alpha-1}}\right|_{x=x_k}\right]\\&+\frac{1}{\hat{n}!}\sum_{r=2}^{\infty}\binom{\alpha}{r}\left.\frac{d^r}{dx^r}q(x)\right|_{x=x_k}J^{r-\alpha}g^{\hat{n}}(\eta(x))\big|_{x=x_k}.\end{aligned} \tag{3.24}$$

At grid points $q(x_i) = 0,\ i = 0, 1, 2,\ ...,\ \hat{n}$, the equation (3.23) and (3.24) will reduce to

$$T_1 = \frac{1}{\hat{n}!} \sum_{r=1}^{\hat{n}} \left(\begin{array}{c} \alpha \\ r \end{array} \right) \frac{d^r}{dx^r} q(x)\bigg|_{x=x_k} J^{r-\alpha} g^{\hat{n}}(\eta(x))\big|_{x=x_k}, \tag{3.25}$$

$$T_2 = \frac{1}{\hat{n}!} \left[\left(\begin{array}{c} \alpha \\ 1 \end{array} \right) \frac{d}{dx} q(x)\bigg|_{x=x_k} \frac{\partial^{\alpha-1} g^{\hat{n}}(\eta(x))}{\partial x^{\alpha-1}}\bigg|_{x=x_k} + \sum_{r=2}^{\infty} \left(\begin{array}{c} \alpha \\ r \end{array} \right) \frac{d^r}{dx^r} q(x)\bigg|_{x=x_k} J^{r-\alpha} g^{\hat{n}}(\eta(x))\big|_{x=x_k} \right]. \tag{3.26}$$

Since $g(x)$ is continuous on $[a, b]$ so bounded $i.e$ $\left|g^{\hat{n}}(x)\right| \leq H,\ H > 0$, and $\left| \frac{\partial^{\alpha-1} g^{\hat{n}}(\eta(x))}{\partial x^{\alpha-1}}\Big|_{x=x_k} \right| = K > 0$. Also, $q(x) = \prod_{i=1}^{\hat{n}} (x - x_i)$ is differentiable in [a,b], so $max \left| \frac{d^r}{dx^r} q(x)\big|_{x=x_k} \right| = M > 0,\ r, k =\ 1, 2,\ ...,\ \hat{n}$. Then the equation (3.25) and (3.26) will take the form as follows

$$T_1 \leq \frac{1}{(n-1)!} \sum_{r=1}^{\hat{n}} |{}^{\alpha}C_r| M \frac{H^{r-\alpha}}{\Gamma(r-\alpha+1)}, \tag{3.27}$$

$$T_2 \leq \frac{1}{n!} \left[|{}^{\alpha}C_1|\, MK + (\hat{n}-1) \sum_{r=2}^{\hat{n}} |{}^{\alpha}C_r| M \frac{H^{r-\alpha}}{\Gamma(r-\alpha+1)} \right]. \tag{3.28}$$

Since $\sum_{r=1}^{\hat{n}} |{}^{\alpha}C_r| M \frac{H^{r-\alpha}}{\Gamma(r-\alpha+1)} > 0,\ K > 0,\ M > 0$, let $\sum_{r=1}^{\hat{n}} |{}^{\alpha}C_r| \frac{H^{r-\alpha}}{\Gamma(r-\alpha+1)} M = e_1 > 0$, and $|{}^{\alpha}C_1|\, MK + (\hat{n}-1) \sum_{r=2}^{\hat{n}} |{}^{\alpha}C_r| M \frac{H^{r-\alpha}}{\Gamma(r-\alpha+1)} = e_2 > 0$.
Thus from (3.27) and (3.28), we have

$$|T_i| \leq \frac{1}{(\hat{n}-2+i)!} e_i.$$

This inequality shows that for the fractional derivative of $g(x)$, the truncation error is reciprocal to $\hat{n}$, as $\hat{n} \to \infty,\ |T_i| \to 0$. □

3.5 Application

For the sake of application of the proposed method we consider the fractional Huxley, fractional Burger and generalized fractional Burger-Fisher equations as test problems.

3.5.1 Fractional Burgers' equation

Consider the one dimensional non-linear space and time fractional Burgers' equation

$$\begin{aligned} {}_t^C D^{\alpha} u(x,t) &= {}_x^C D^{\beta_2}\, u(x,t) - u\ {}_x^C D^{\beta_1}\, u(x,t), \quad (x,t) \in \tau \times [0, T^*], \qquad (3.29) \\ u(x,0) &= \varphi(x) := 2x, \\ u(0,t) &= \varphi_1(t) := 0, \qquad u(1,t) = \varphi_2(t) := \frac{2}{1+2t}, \end{aligned}$$

where $\tau = (a,b)$, $0 < \alpha, \beta_1 \leq 1$, $0 < \beta_2 \leq 2$. Exact solution of equation (3.29) with $\alpha = \beta_1 = 1$ and $\beta_2 = 2$ is given in [43]. Applying the proposed method on (3.29), we get the following system of non-linear fractional ODEs in time as

$$\begin{aligned} {}_t^C D^{\alpha} u(x_q,t) = &\sum_{\ell=2}^{\hat{n}-1} \hat{\Omega}_{q\ell}^{(\beta_2)} u(x_\ell,t) - u(x_q,t) \left(\sum_{\ell=2}^{\hat{n}-1} \hat{\Omega}_{q\ell}^{(\beta_1)} u(x_\ell,t) + \hat{\Omega}_{q1}^{(\beta_1)} \varphi_1(t) + \hat{\Omega}_{q\hat{n}}^{(\beta_1)} \varphi_2(t) \right) \\ &+ \hat{\Omega}_{q1}^{(\beta_2)} \varphi_1(t) + \hat{\Omega}_{q\hat{n}}^{(\beta_2)} \varphi_2(t), \qquad q = 1, 2, \cdots, \hat{n}. \end{aligned} \qquad (3.30)$$

The obtained system (3.30) is solved by using Adam Bashforth Moulton predictor-corrector method in MATLAB, and the obtained results are displayed in Table (3.1) at different values of α, β_1 and β_2 which shows that the numerical solutions u_{MDQM} converges to the exact solutions u_{Exact} as α, β_1 and β_2 approaches to 1 and 2 respectively. We also plot the exact solutions and the obtained solutions by proposed method at $x = 0.5$ in Figure (3.1) with $\hat{n} = 5$ and $h = 0.01$ at different values of α, β_1 and β_2 and observe that the obtained results are in quite good agreement with exact solutions. We can also have more accurate results by increasing $\hat{n}$ and with smaller values of step size h as mentioned in above theorem.

3.5.2 Fractional order Huxley equation

Consider the following general form of non-linear fractional Huxley equation

$$\begin{aligned} {}_t D^{\alpha} u(x,t) &= {}_x D^{\beta} u(x,t) - u(u-1)(\gamma - u), \qquad (x,t) \in \vartheta \times [0, T^*], \qquad (3.31) \\ u(x,0) &:= \varphi(x) = \left(\frac{1 + \tanh(x/2\sqrt{2})}{2} \right), \quad 0 \leq x \leq 1, \\ u(0,t) &:= \mu_1(t) = \left(\frac{1 + \tanh(t/4)}{2} \right), \quad u(1,t) := \mu_2(t) = \frac{1}{2} \left(1 + \tanh \left\{ \frac{1}{2\sqrt{2}} - \frac{t}{4} \right\} \right), \end{aligned}$$

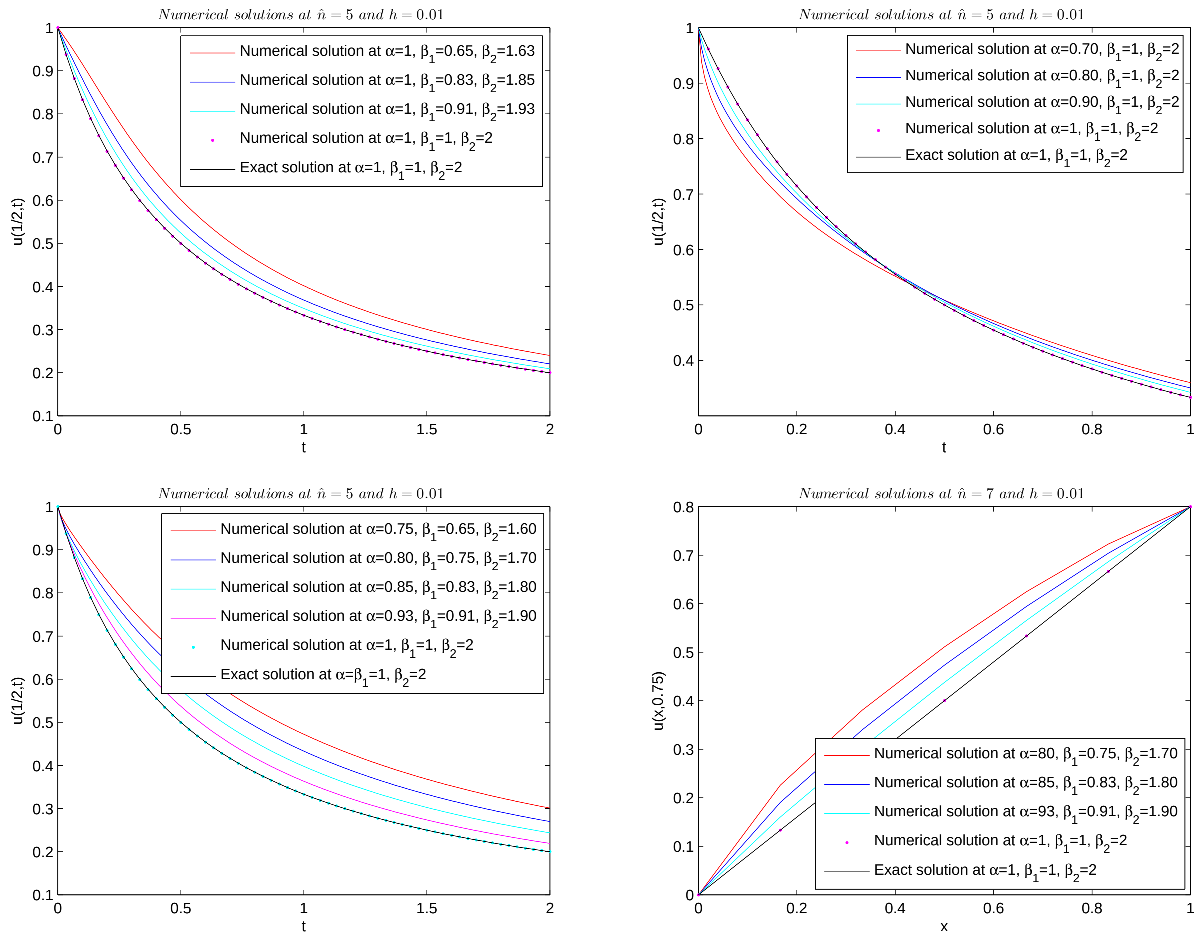


Figure 3.1: Numerical results of equation (3.29) by proposed scheme at $x = 0.5$ and $\hat{n} = 5, 7$, $h = 0.01$ for different values of α and β.

Table 3.1: Numerical results of problem (3.29) by proposed scheme at different values of α, β_1 and β_2, with $h = 0.002$, $\hat{n} = 5$.

x	t	$\alpha = 0.90,\ \beta_1 = 0.83,\ \beta_2 = 1.85$	$\alpha = 0.95,\ \beta_1 = 0.91,\ \beta_2 = 1.93$	$\alpha = \beta_1 = 1,\ \beta_2 = 2$		
		u_{MDQM}	u_{MDQM}	u_{MDQM}	u_{Exact}	E_{MDQM}
0.01	0.2	0.1625298	0.1517983	0.1428571	0.1428571	$5.2096e^{-11}$
	0.5	0.1205965	0.1093150	0.0999999	0.1000000	$1.5280e^{-11}$
	0.8	0.0948358	0.0848730	0.0769230	0.0769230	$6.8731e^{-12}$
0.3	0.3	0.4277532	0.3992321	0.3749999	0.3750000	$1.0266e^{-11}$
	0.7	0.2982079	0.2716698	0.2499999	0.2500000	$2.9664e^{-11}$
	0.9	0.2579223	0.2336828	0.2142857	0.2142857	$1.8838e^{-11}$
0.7	0.3	0.9126574	0.8927542	0.8750000	0.8750000	$2.5090e^{-10}$
	0.6	0.6773933	0.6555298	0.6363636	0.6363636	$8.9339e^{-11}$
	0.9	0.5365901	0.5167735	0.5000000	0.5000000	$4.0757e^{-11}$
0.9	0.5	0.9152454	0.9071864	0.90000000	0.9000000	$7.1191e^{-11}$
	0.7	0.7646254	0.7568029	0.7500000	0.7500000	$3.9697e^{-11}$
	0.9	0.6563434	0.6490557	0.6428571	0.6428571	$2.4301e^{-11}$

where $0 < \alpha \leq 1$, $1 < \beta \leq 2$, $\vartheta \in [0,1]$, $t \in [0, T^*]$, $T^* > 0$, and γ is constant. The exact solutions of (3.31), when $\alpha = 1$ and $\beta = 2$ is given in [51]. By using DQM approximation (3.18) in problem (3.31), we obtain the following non-linear system of fractional order ODEs as

$$ {}_tD^{\alpha}u(x_j,t) = \sum_{k=2}^{\hat{n}-1} \hat{\Omega}_{jk}^{(\beta)} u(x_k,t) + \hat{\Omega}_{j1}^{(\beta)} \mu_1(t) + \hat{\Omega}_{j\hat{n}}^{(\beta)} \mu_2(t) + u(x_j,t)\left(u(x_j,t) - 1\right)\left(\gamma - u(x_j,t)\right), \tag{3.32} $$

$$ u(x_j,0) = \varphi(x_j), \qquad j = 2,3,\cdots,\hat{n}-1. $$

We solved the problem (3.31) for $\gamma = 1$, and the results are shown with the help of Figures and Tables. Table (3.2) lists the obtained numerical results at different values of α and β, which demonstrate that the obtained results by proposed method are converging to the exact solutions when α and β approach to 1 and 2 respectively. Figure (3.2) is used to plot the exact solutions at $\alpha = 1$, $\beta = 2$ along with the obtained numerical solutions, at $x = 0.5$ and for different values of α and β. In Table (3.3), we compare the obtained results with the results by Haar wavelet method [52] and exact solutions. Where E_{MDQM} and E_{HW} represents the absolute error by modified differential quadrature and Haar wavelet methods respectively. From the numerical results displayed in Table (3.2) and (3.3), we can claim that our results are in quite

Table 3.2: Numerical results of equation (3.31) by proposed method at different values of α, and β, with $h = 0.005$, $\hat{n} = 7$.

x	t	$\alpha = 0.80,\ \beta = 1.81$	$\alpha = 0.90,\ \beta = 1.92$	$\alpha = 1,\ \beta = 2,$		
		u_{MDQM}	u_{MDQM}	u_{MDQM}	u_{Exact}	u_{Error}
0.1	0.2	0.49639	0.49405	0.49268	0.49267	$7.7390e^{-6}$
	0.5	0.46065	0.45740	0.45533	0.45529	$3.8524e^{-5}$
	0.7	0.43656	0.43298	0.43068	0.43062	$5.8182e^{-5}$
	0.9	0.41262	0.40881	0.40637	0.40629	$7.6717e^{-5}$
0.4	0.1	0.56161	0.55896	0.55789	0.55794	$4.9156e^{-5}$
	0.3	0.54114	0.53626	0.53313	0.53316	$2.9097e^{-5}$
	0.6	0.50678	0.50025	0.49571	0.49571	$1.9417e^{-6}$
	0.9	0.47106	0.46358	0.45834	0.45830	$3.2893e^{-5}$
0.8	0.2	0.61695	0.61517	0.61430	0.61436	$5.7539e^{-5}$
	0.4	0.59473	0.59213	0.59037	0.59041	$4.5690e^{-5}$
	0.6	0.57141	0.56827	0.56600	0.56603	$3.2970e^{-5}$
	0.8	0.54744	0.54391	0.54130	0.54132	$1.9607e^{-5}$

good agreement with the exact solutions and better than Haar wavelet method.

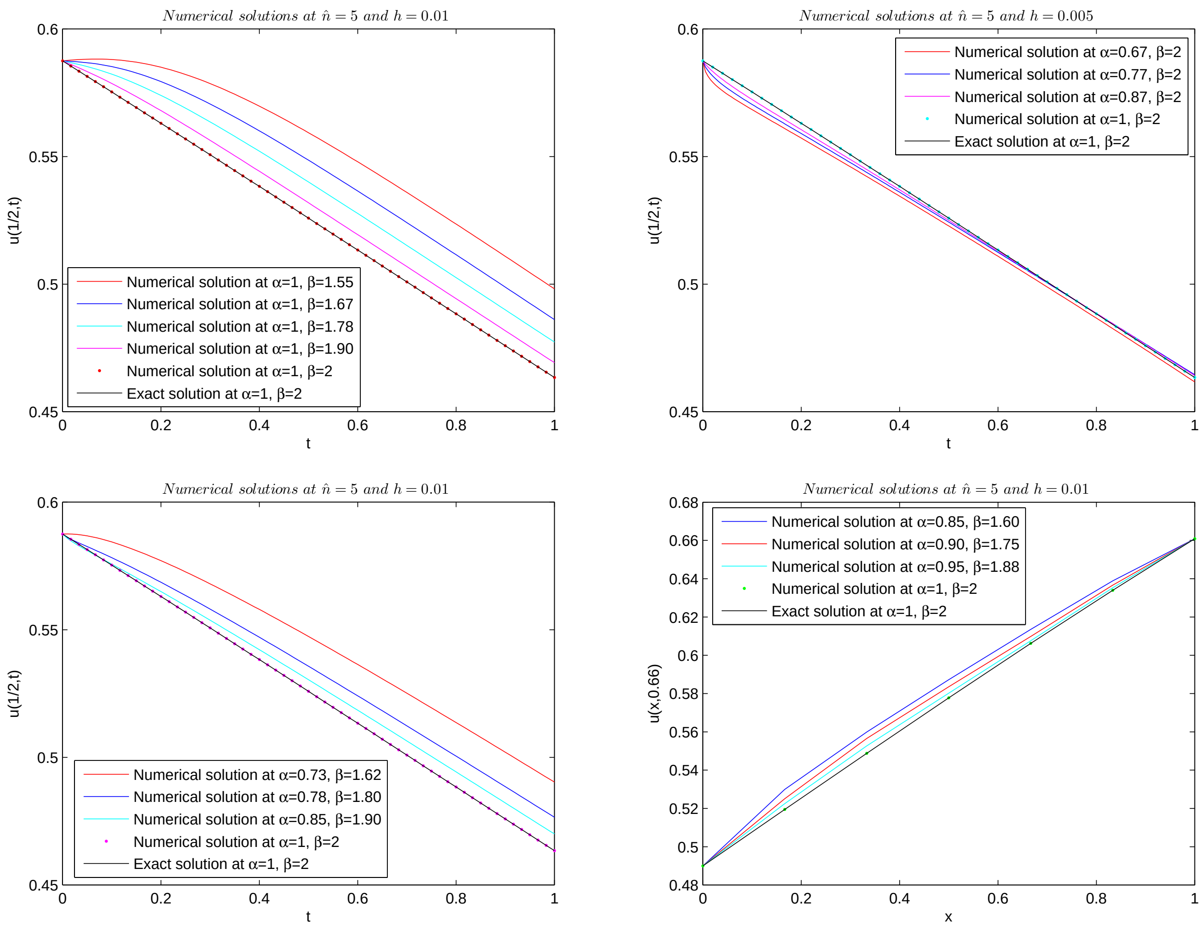


Figure 3.2: Numerical results of Example (3.31) by proposed scheme at $x = 0.5$ and $\hat{n} = 5$, $h = 0.01$ for different values of α and β.

Table 3.3: Comparison between solutions of problem (3.31) by present method with exact solutions and Haar wavelet method at $\beta = 2$ and $h = 0.01,\ \hat{n} = 5$.

t	x	u_{MDQM}	u_{Exact}	E_{MDQM}	E_{HW}
0.4	0.03125	0.455669	0.455641	$2.8118e^{-5}$	$9.5553e^{-5}$
	0.28125	0.499707	0.499718	$1.0838e^{-5}$	$3.7861e^{-3}$
	0.53125	0.543754	0.543800	$4.6144e^{-5}$	$1.2628e^{-2}$
	0.71875	0.576400	0.576453	$5.3785e^{-5}$	$2.2813e^{-2}$
	0.96875	0.618842	0.618930	$8.8005e^{-5}$	$4.9314e^{-2}$
0.6	0.03125	0.431016	0.430968	$4.8723e^{-5}$	$1.8745e^{-4}$
	0.28125	0.474750	0.474740	$1.0495e^{-5}$	$6.3583e^{-3}$
	0.53125	0.518878	0.518903	$2.5337e^{-5}$	$2.0833e^{-2}$
	0.71875	0.551837	0.551870	$3.3259e^{-5}$	$3.7170e^{-2}$
	0.96875	0.595011	0.595080	$6.9748e^{-5}$	$8.8748e^{-2}$
1.0	0.03125	0.382834	0.382747	$8.6744e^{-5}$	$4.2376e^{-4}$
	0.28125	0.425334	0.425282	$5.2244e^{-5}$	$1.0439e^{-2}$
	0.53125	0.468969	0.498952	$1.7370e^{-5}$	$3.2187e^{-2}$
	0.71875	0.502067	0.502058	$9.2660e^{-6}$	$5.6207e^{-2}$
	0.96875	0.546091	0.546120	$2.9612e^{-5}$	$1.7284e^{-1}$

3.5.3 Generalized fractional Burger-Fisher equation

Consider the generalized form of fractional Burger-Fisher equation [44]

$$\frac{\partial^{\alpha} u}{\partial t^{\alpha}} - \frac{\partial^{\beta_1} u}{\partial x^{\beta_1}} + \delta u^{\gamma} \frac{\partial^{\beta_2} u}{\partial x^{\beta_2}} + \rho u(u^{\gamma} - 1) = 0, \qquad (x,t) \in \vartheta \times [0, T^*], \tag{3.33}$$

subject to the initial and boundary conditions

$$u(x,0) = h(x) = \left(\frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\delta\gamma x}{2(1+\rho)}\right)\right)^{\gamma^{-1}} \quad 0 \le x \le 1,$$

$$u(0,t) = \xi_1(t) = \left(\frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\delta\gamma}{2(1+\rho)}\left[-\left(\frac{\delta^2 + \rho(1+\gamma)^2}{\delta(1+\gamma)}\right)t\right]\right)\right)^{\gamma^{-1}},$$

$$u(1,t) = \xi_2(t) = \left(\frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\delta\gamma}{2(1+\rho)}\left[1-\left(\frac{\delta^2 + \rho(1+\gamma)^2}{\delta(1+\gamma)}\right)t\right]\right)\right)^{\gamma^{-1}},$$

where $0 < \alpha, \beta_2 \le 1,\ 1 < \beta_1 \le 2,\ \vartheta = [0,1],\ t \in [0, T^*],\ \ T^* > 0$, and $\rho,\ \gamma,\ \delta$ are constants. The exact solution of problem (3.33), when $\alpha = \beta_2 = 1$ and $\beta_1 = 2$ is given in [44]. We solve problem (3.33) in similar way by using derivative approximations

(3.18), we have the following non-linear system of fractional order ODEs

$$
\begin{aligned}
{}_tD^{\alpha}u(x_j,t) = \sum_{k=2}^{\hat{n}-1} \hat{\Omega}_{jk}^{(\beta_1)} u(x_k,t) + \delta u^{\gamma}(x_j,t) \left[\sum_{k=2}^{\hat{n}-1} \hat{\Omega}_{jk}^{(\beta_2)} u(x_k,t) + \hat{\Omega}_{j1}^{(\beta_2)} \xi_1(t) + \hat{\Omega}_{j\hat{n}}^{(\beta_2)} \xi_2(t) \right] \\
+ \delta u(x_j,t)\left(u^{\gamma}(x_j,t)-1\right) + \left(\hat{\Omega}_{j1}^{(\beta_1)} \xi_1(t) + \hat{\Omega}_{j\hat{n}}^{(\beta_1)} \xi_2(t) \right),
\end{aligned}
\tag{3.34}
$$

with the initial conditions

$$
u(x_j,0) = h(x_j), \quad j = 2,3, \cdots, \hat{n}-1.
$$

The system (3.34) is solved by Adam Bashforth Moulton predictor-corrector method in MATLAB, and the obtained results are displayed below with the help of Figures and Tables. Table (3.4) lists the numerical solutions of equation (3.33) by proposed scheme at different values of α, β_1 and β_2. We also compared the obtained results with the exact solutions, the variational iteration method [53], the differential transformation method [53] and the homotopy perturbation method [44] as shown in Table (3.5), (3.6), and (3.7) by choosing different values of δ, ρ and γ. Where E_{MDQM}, E_{HPM}, E_{VIM} and E_{RDTM} in the following tables presents the absolute error by modified differential quadrature method, homotopy perturbation method, variation iteration method and reduced differential transformation method respectively. Figure (3.3) demonstrate the solutions of problem (3.33) by proposed scheme at different values of α, β_1 and β_2. It can be observed from Table (3.4) and Figure (3.3), that our results are converging to the exact solutions as α, β_1 and β_2 approach to the integer values.

Table 3.4: Numerical results of problem (3.33) at different values of $\alpha,\ \beta_1$ and β_2 with $h = 0.005,\ \hat{n} = 7,\ \gamma = 2$ and $\delta = \rho = 1$.

x	t	$\alpha = 0.71,\ \beta_2 = 0.76,\ \beta_1 = 1.73$	$\alpha = 0.85,\ \beta_2 = 0.83,\ \beta_1 = 1.88$	$\alpha = \beta_2 = 1,\ \beta_1 = 2,$		
		u_{MDQM}	u_{MDQM}	u_{MDQM}	u_{Exact}	E_{MDQM}
0.1	0.2	0.77306	0.77135	0.77019	0.77028	$9.0827e^{-5}$
	0.4	0.83231	0.83263	0.83338	0.83349	$1.0217e^{-4}$
	0.6	0.87956	0.88133	0.88317	0.88327	$9.6179e^{-5}$
	0.8	0.91518	0.91786	0.92023	0.92031	$8.0058e^{-5}$
0.4	0.1	0.71416	0.70671	0.69914	0.69920	$6.1122e^{-5}$
	0.3	0.77798	0.77542	0.77365	0.77374	$9.1177e^{-5}$
	0.6	0.85532	0.85842	0.86243	0.86253	$1.0084e^{-4}$
	0.9	0.90924	0.91550	0.92178	0.92186	$7.9749e^{-5}$
0.8	0.2	0.69863	0.69512	0.69118	0.69122	$3.9896e^{-5}$
	0.5	0.79849	0.79906	0.80028	0.80034	$6.5851e^{-5}$
	0.7	0.85179	0.85434	0.85749	0.85755	$6.7374e^{-5}$
	0.9	0.89323	0.89714	0.90135	0.90141	$5.9434e^{-5}$

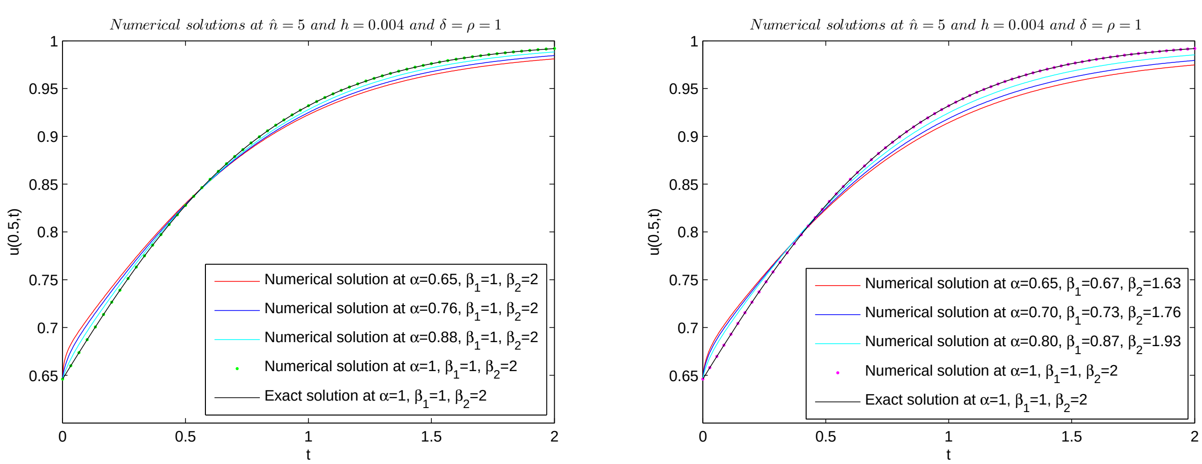


Figure 3.3: Numerical results of Example (3.33) by present method for different values of $\alpha,\ \beta_1$ and β_2 at $x = 0.5,\ h = 0.0025,\ \hat{n} = 5, 7$.

Table 3.5: Comparison of the solutions obtained by present method, variational iteration method, differential transformation method and exact solutions for integer values of $\alpha,\ \beta_1,$ and β_2 with $h = 0.01,\ \hat{n} = 5,\ \gamma = 1$ and $\delta = \rho = 0.001$.

x	t	u_{MDQM}	u_{Exact}	E_{MDQM}	E_{VIM}	E_{RDTM}
0.01	0.02	0.500003751	0.500003751	$7.7715e^{-16}$	$2.5031e^{-3}$	$0.4999e^{-5}$
	0.04	0.500008752	0.500008752	$1.1102e^{-16}$	$2.5081e^{-3}$	$0.4999e^{-5}$
	0.06	0.500013754	0.500013754	$6.6613e^{-16}$	$2.5131e^{-3}$	$1.4999e^{-5}$
	0.08	0.500018755	0.500018755	$1.3322e^{-15}$	$2.5181e^{-3}$	$1.9999e^{-5}$
0.04	0.02	0.500000001	0.500000001	$3.6637e^{-15}$	$9.9962e^{-3}$	$0.4997e^{-5}$
	0.04	0.500005002	0.500005002	$1.1102e^{-15}$	$1.0001e^{-2}$	$0.4997e^{-5}$
	0.06	0.500010004	0.500010004	$1.5543e^{-15}$	$1.0006e^{-2}$	$1.4997e^{-5}$
	0.08	0.500015004	0.500015005	$4.1078e^{-15}$	$1.0011e^{-2}$	$1.9997e^{-5}$
0.08	0.02	0.499995001	0.499995001	$7.3829e^{-15}$	$1.9989e^{-2}$	$0.4995e^{-5}$
	0.04	0.500000002	0.500000002	$3.2196e^{-15}$	$1.9984e^{-2}$	$0.9995e^{-5}$
	0.06	0.50005004	0.500005004	$9.9920e^{-16}$	$1.9989e^{-2}$	$1.4995e^{-5}$
	0.08	0.500010005	0.500010005	$5.3290e^{-15}$	$1.9994e^{-2}$	$1.9995e^{-5}$

Table 3.6: Comparison of the obtained results and the results by reduced differential transformation method at integer values of $\alpha,\ \beta_1$ and β_2 with $h = 0.01,\ \hat{n} = 5,\ \gamma = 2$ and $\delta = \rho = 0.001$.

x	t	u_{MDQM}	u_{Exact}	E_{MDQM}	E_{RDTM}
0.01	0.02	0.7071126744	0.7071126745	$2.3569e^{-11}$	$4.7133e^{-6}$
	0.04	0.7071197462	0.7071197462	$2.3571e^{-11}$	$9.4271e^{-6}$
	0.06	0.7071268179	0.7071268179	$2.3572e^{-11}$	$1.4142e^{-5}$
	0.08	0.7071338895	0.7071338895	$2.3573e^{-11}$	$1.8855e^{-5}$
0.04	0.02	0.7071091389	0.7071091389	$8.2493e^{-11}$	$4.7117e^{-6}$
	0.04	0.7071162107	0.7071162107	$8.2498e^{-11}$	$9.4260e^{-6}$
	0.06	0.7071232824	0.7071232825	$8.2502e^{-11}$	$1.4140e^{-5}$
	0.08	0.7071303540	0.7071303541	$8.2506e^{-11}$	$1.8854e^{-5}$
0.08	0.02	0.7071044248	0.7071044249	$1.3556e^{-10}$	$4.7104e^{-6}$
	0.04	0.7071114966	0.7071149678	$1.3356e^{-10}$	$9.4241e^{-6}$
	0.06	0.7071185684	0.7071185685	$1.3557e^{-10}$	$1.4138e^{-5}$
	0.08	0.7071256401	0.7071256402	$1.3358e^{-10}$	$1.8852e^{-5}$

Table 3.7: Comparison between solutions by proposed method and Homotopy perturbation method (HPM) for different values of δ and ρ, at integer values of $\alpha,\ \beta_1$ and β_2 with $h = 0.01,\ \hat{n} = 5,\ \gamma = 2$.

x	t	E_{MDQM} $\delta=\rho=0.01$	E_{HPM} $\delta=\rho=0.01$	E_{MDQM} $\delta=\rho=0.1$	E_{HPM} $\delta=\rho=0.1$	E_{MDQM} $\delta=\rho=0.5$	E_{HPM} $\delta=\rho=0.5$
0.1	0.2	$1.6259e^{-11}$	$6.2800e^{-11}$	$1.6727e^{-8}$	$4.3262e^{-8}$	$2.3450e^{-6}$	$1.8855e^{-5}$
	0.4	$3.7143e^{-11}$	$5.0800e^{-11}$	$3.8063e^{-8}$	$4.7133e^{-6}$	$5.2087e^{-6}$	$1.8855e^{-5}$
	0.6	$5.8025e^{-11}$	$1.6380e^{-10}$	$5.9367e^{-8}$	$4.7133e^{-6}$	$7.9431e^{-6}$	$1.8855e^{-5}$
	0.8	$7.8908e^{-11}$	$3.0250e^{-10}$	$8.0622e^{-8}$	$4.7133e^{-6}$	$1.0484e^{-6}$	$1.8855e^{-5}$
0.4	0.2	$5.3265e^{-13}$	$3.5000e^{-10}$	$6.3662e^{-11}$	$4.7133e^{-6}$	$2.5227e^{-6}$	$1.8855e^{-5}$
	0.4	$2.0351e^{-11}$	$6.0200e^{-10}$	$2.1286e^{-8}$	$4.7133e^{-6}$	$3.1657e^{-6}$	$1.8855e^{-5}$
	0.6	$4.1235e^{-11}$	$1.6560e^{-9}$	$4.2618e^{-8}$	$4.7133e^{-6}$	$5.9999e^{-6}$	$1.8855e^{-5}$
	0.8	$6.2119e^{-11}$	$2.7100e^{-9}$	$6.3914e^{-8}$	$4.7133e^{-6}$	$8.6860e^{-6}$	$1.8855e^{-5}$
0.8	0.2	$1.2698e^{-11}$	$6.6990e^{-9}$	$1.2385e^{-8}$	$4.7133e^{-6}$	$1.3719e^{-6}$	$1.8855e^{-5}$
	0.4	$1.2209e^{-12}$	$2.6960e^{-9}$	$1.8453e^{-8}$	$4.7133e^{-6}$	$5.7731e^{-7}$	$1.8855e^{-5}$
	0.6	$1.5140e^{-11}$	$1.3110e^{-9}$	$1.6074e^{-8}$	$4.7133e^{-6}$	$2.5120e^{-6}$	$1.8855e^{-5}$
	0.8	$2.9060e^{-11}$	$5.3200e^{-9}$	$3.0290e^{-8}$	$4.7133e^{-6}$	$4.3839e^{-6}$	$1.8855e^{-5}$

Chapter 4

Conclusion

This thesis revolves about the modified cubic B-splines and polynomial based differential quadrature methods for the numerical approximations of some very important non-linear integer and fractional order PDEs. Weights required for the approximations of the spatial derivatives, which are involved in given partial differential equations are determined by using cubic B-splines and shifted Legendre polynomials. We have employed the modified cubic B-splines DQM to solve the integer order non-linear partial differential equations. The differential quadrature method reduces the given non-linear partial differential equation into a system of non linear ODEs. The resulting system of non-linear ODEs is solved by RK-4 method in MATLAB. For the sake of application, we considered Fitz Nagumo, generalized Burger-Fisher and Burgers' equations. Comparison of the obtained results with exact values is also provided to show the efficiency and effectiveness of the method. We also presented an efficient numerical scheme for non-linear space and time fractional partial differential equations. The fractional derivatives of Lagrange polynomials $l_q(x)$ are represented in terms of fractional derivative of shifted Legendre polynomials by constructing transformation matrices $\mathbf{C}_{\hat{\mathbf{n}}\times\hat{\mathbf{n}}}$. We derived and constructed the matrices of weighting coefficients $\hat{\boldsymbol{\Omega}}_{\hat{n}\times\hat{n}}$ for Caputo fractional, β, order derivatives.

We successfully used these matrices to solve the time and space fractional non-linear generalized Burger-Fisher and Huxley equations. Adam Bashforth Moulton predictor-corrector approach is implemented to solve the reduced system of fractional order,

α, ODEs. We observed that the obtained results of time and space fractional non-linear problems converge to the exact solution of integer order problem, when fractional values of α, β_1 and β_2 approaches to the integer values as shown in Figures (3.1), (3.2), (3.3) and Tables (3.1), (3.2) and (3.4). The obtained results by present method are also compared with some earlier work present in literature. The reported results in Tables (3.5), (3.6) and (3.7) show that the results obtained by proposed technique are in quite good agreement with the exact solutions and are better than the reduce differential transform method, variation iteration method and Haar wavelet method. For the proposed method, we require a few number of grid points, thus less computational work and very low data complexity leads to minimizing errors. Different types of time and space fractional PDEs can be easily handled by the presented technique.

Bibliography

[1] Q. S Zeng, G. Y Cao and X. J Zhu. The effect of the fractional-order controller's orders variation. *In proceedings of the 1st International conference on machine Learning and Cybernetics*, **1**:367–372, 2002.

[2] R. L Magin and M. Ovadia. Modeling the cardiac tissue electrode interface using fractional. *Journal of Vibration and Control*, **14**:1431–144, 2008.

[3] R. Singh and S. Manoj. Fractional calculus applied in solving instability phenomena in fluid dynamics. *International journal of civil engineering and technology*, **6**:34–44, 2015.

[4] Z. Z Yang and J. L Zhou. An improved design for the IIR-type digital fractional order differential filter. *In proceedings of the international seminar on future Bio-Medical information engineering (FBIE 08)*, 473–476, 2008.

[5] C. A Bavastri, J. J De Espindola and E. M De Oliveira Lopes. Design of optimum systems of viscoelastic vibration absorbers for a given material based on the fractional calculus model. *Journal of Vibration and Control*, **14**:1607–1630, 2008.

[6] N. Heymans. Dynamic measurements in long-memory materials fractional calculus evaluation of approach to steady state. *Journal of Vibration and Control*, **14**:1587–1596, 2008.

[7] B. T Krishna and K. V. V. S Reddy. Active and passive realization of fractance device of order 1/2. *Active and Passive Electronic Components*, 2008.

[8] Y. Pu, X. Yuan, K. Liao, J. Zhou, N. Zhang, X. Pu and Y. Zeng. A recursive two-circuits series analog fractance circuit for any order fractional calculus. *International Society for Optics and Photonics*, **6027**:509–519, 2006.

[9] J. A. T Machado, M. F Lima and M. Cris. Experimental signal analysis of robot impacts in a fractional calculus perspective. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **11**:1079–1085, 2007.

[10] G. Aguilar et al. Homotopy perturbation transform method for nonlinear differential equations involving to fractional operator with exponential kernel. *Advances in Difference Equations*, 2017.

[11] U. Saeed and M. U Rehman. Haar wavelet picard method for fractional nonlinear partial differential equations. *Applied Mathematics and Computation*, 310–322, 2015.

[12] I. Celik. Haar wavelet method for solving generalized burgers-huxley equation. *Arab Journal of Mathematical Sciences*, **18**:25–37, 2012.

[13] I. Hashim and M. S. M Noorani. Solving the generalized burgers-huxley equation using the Adomian decomposition method. *Applied Mathematics and Computation*, **43**:1404–1411, 2006.

[14] N. A. H Ismail, K. Raslan, and A. A. A Rabboh. Adomian decomposition method for burger's-huxley and burger's-fisher equations. *Applied Mathematics and Computation*, 291–301, 2004.

[15] I. Hashim and B. Batiha. A note on the adomian decomposition method for the generalized huxley equation. *Applied Mathematics and Computation*, **181**:1439–1445, 2006.

[16] S. L Lili, L. Xiaoyan and J. Wei. Adomians method applied to solve ordinary and partial fractional differential equations. *Journal of Shanghai Jiaotong University (Science)*, **22**:371–376, 2017.

[17] C. Canuto, Q. Hussaini, Z. Alfio, M. Youssuff and A. Thomas. Spectral methods, 2006.

[18] J. W Thomas. *Numerical partial differential equations: finite difference methods*, **22**, 2013.

[19] O. C Zienkiewicz and L. Robert. *The finite element method*, volume **36**, 1977.

[20] Z. J Wang. Spectral (finite) volume method for conservation laws on unstructured grids, Basic formulation. *Journal of computational physics*, **178**:210–251, 2002.

[21] G. Adomian. A review of the decomposition method and some recent results for nonlinear equations. *Mathematical and Computer Modeling*, **13**:17–43, 1990.

[22] C. Shu. Differential Quadrature and Its Application in Engineering. *Springer-Verlag London Berlin Heidelberg*, 2000.

[23] C. T Chang and J. R Quan. New insights in solving distributed system equations by the quadrature method-II. numerical experiments. *Elsevier*, **13**:1017–1024, 1989.

[24] R. Bellman and B. G Kashef. Differential quadrature: A technique for the rapid solution of nonlinear partial differential equations. *Journal of Computational Physics*, **10**:40–52, 1972.

[25] R. C Mittal. Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. *Applied Mathematics and Computation*, **218**:7839–7855, 2012.

[26] G. Arora. Numerical solution of Burgers' equation with modified cubic b-spline differential quadrature method. *Applied Mathematics and Computation*, **224**:166–177, 2013.

[27] R. Jiwari. Polynomial differential quadrature method for numerical solutions of the generalized fitzhugh nagumo equation with time dependent coefficients. *Ain Shams Engineering*, **5**:1343–1350, 2014.

[28] R. C Mittal and R. Jiwari. A differential quadrature method for numerical solutions of Burgers' type equations. *International Journal of Numerical Methods for Heat and Fluid Flow*, **22**:880–895, 2014.

[29] E. N Aksan and T. Ozis. A finite element approach for solution of Burgers' equation. *Applied Mathematics and Computation*, **139**:417–428, 2003.

[30] Y. L Wu and C. Shu. Integrated radial basis functions-based differential quadrature method and its performance. *International Journal for numerical methods in fluids*, **53**:969–984, 2007.

[31] I. J Schoenberg. On cubic Spline Interpolation at Equidistant nodes. *Mathematics research center, university of wisconsin*, 1971.

[32] L. L Schumaker. Spline functions basic theory. *Cambridge University Press*, 2007.

[33] C. de Boor. A practical guide to Splines. *Springer*, 2001.

[34] P. M Prenter. Splines and Variational methods. *Willey*, 1989.

[35] A. Korkmaz. Quartic B-spline differential quadrature method. *International journal of nonlinear science*, **11**:403–411, 2011.

[36] B. Saka and D. Idris. Galerkin method for the numerical solution of the RLW equation using quintic B-splines. *Journal of Computational and Applied Mathematics*, **190**:532–547, 2006.

[37] C. Pozrikidis. Numerical computation in science and engineering. *Oxford university press New York*, **6**, 1998.

[38] R. L Burden and J. D Faires. Numerical analysis. *Cengage Learning*, **9**, 2010.

[39] A. H Bhrawy. A Jacobi–Gauss–Lobatto collocation method for solving generalized Fitzhugh-Nagumo equation with time-dependent coefficients. *Elsevier*, **222**:255–264, 2013.

[40] G. Arora and B. K Singh. A numerical scheme to solve fisher-type reaction-diffusion equations. *Mathematics in Engineering, Science and Aerospace*, **5**:153–164, 2014.

[41] S. Abbasbandy. Soliton solutions for the fitzhugh-nagumo equation with the homotopy analysis method. *Elsevier*, **32**:2706–2714, 2008.

[42] R. Ezzati and K. Shakibi. Using Adomian's decomposition and multiquadric quasi-interpolation methods for solving newell-whitehead equation. *Elsevier*, **3**:1043–1048, 2011.

[43] U. Saeed, M. U Rehman and Q. Din. Differential quadrature method for nonlinear fractional partial differential equations. *Engineering Computations*, **35**:2349–2366, 2018.

[44] M. M Rashidi and D. D Ganji. Explicit analytical solutions of the generalized burger and burger-fisher equations by homotopy perturbation method. *Wiley Inter Science*, 409–417, 2008.

[45] D. Kaya and S. M El-Sayed. A numerical simulation and explicit solutions of the generalized burgers-fisher equation. *Applied Mathematics and Computation*, **19**:847–898, 2004.

[46] A. M Wazwaz. The tanh method for generalized forms of nonlinear heat conduction and $Burger-Fisher$ equations. *Applied Mathematics and Computation*, **152**:403–416, 2004.

[47] A. A Soliman. A numerical simulation and explicit solutions of KDV-Burgers' and lax's seventh-order KDV equations. *Chaos Solitons Fractals*, **29**:294–302, 2006.

[48] K. Diethelm, A. D Freed and J. F Neville. Detailed error analysis for a fractional adams method. *Numerical Algorithms*, **36**:31–52, 2004.

[49] A. G Anastassiou. On right fractional calculus. *Chaos, Solitons and Fractals*, 365–376, 2009.

[50] K. Diethelm, J. F Neville and A. D Freed. A predictor-corrector approach for the numerical solution of fractional differential equations, 2001.

[51] X. W Zhou. Exp-function method for solving huxley equation. *Hindawi Publishing Corporation*, 2008.

[52] S. S Ray and A. K Gupta. On the solution of Burgers–Huxley and Huxley equation using wavelet collocation method. *CMES*, **6**:409–424, 2013.

[53] A. Kurnaz and D. Kocacoban. A better approximation to the solution of Burger-Fisher equation. *Proceedings of the World Congress on Engineering*, **1**, 2011.