NUMERICAL SOLUTION OF NONLINEAR EVOLUTION EQUATIONS USING B-SPLINE BASIS FUNCTION IN DIFFERENTIAL QUADRATURE METHOD

A Thesis

Submitted in partial fulfilment of the requirements for

the award of the degree of

DOCTOR OF PHILOSOPHY

IN

MATHEMATICS

By

Shubham Mishra

Registration Number: 12009881

Supervised By

Dr. Geeta Arora (18820)

Professor

Department of Mathematics

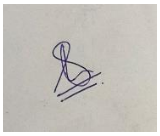
LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

2023

DECLARATION

I, hereby declared that the presented work in the thesis entitled **“NUMERICAL SOLUTION OF NONLINEAR EVOLUTION EQUATIONS USING B-SPLINE BASIS FUNCTION IN DIFFERENTIAL QUADRATURE METHOD”** in fulfilment of degree of **Doctor of Philosophy (Ph.D.)** is outcome of research work carried out by me under the supervision Dr. Geeta Arora, working as Professor, in the School of Chemical Engineering and Physical Sciences of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of others investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.



(Signature of Scholar)

Name of the scholar: Shubham Mishra

Registration No.: 12009881

Department/school: Mathematics/School of Chemical Engineering and Physical Sciences

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled **"NUMERICAL SOLUTION OF NONLINEAR EVOLUTION EQUATIONS USING B-SPLINE BASIS FUNCTION IN DIFFERENTIAL QUADRATURE METHOD"** submitted in fulfilment of the requirement for the reward of degree of **Doctor of Philosophy (Ph.D.)** in the Department of Chemical and Physical Sciences is a research work carried out by Shubham Mishra, Reg. No 12009881, is bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

Geeta Arora

(Signature of Supervisor)

Name of supervisor: Dr. Geeta Arora

Designation: Professor

Department/School: Mathematics/ School of Chemical Engineering and Physical Sciences

University: Lovely Professional University Jalandhar.

ABSTRACT

Nonlinear evolutionary equations (NLEEs) are critical tools for describing nonlinear phenomena in various scientific fields, such as biology, physics, pattern formation, solitons, ecology, heat transfer, chemistry and nonlinear dispersion. Solitary waves, which are solutions to NLEEs, maintain their shape while moving at a constant speed, and have received significant attention from the scientific community. This research provides, a comprehensive understanding of the origins and development of solitons and their behavior under different NLEEs. The versatility of solitons is explored, including their applications in plasma physics, nonlinear optics, epidemiology, and nonlinear Bose-Einstein condensation. NLEEs are mathematical models used to describe the time evolution of various physical, biological, and social systems. It is kind of partial differential equations (PDEs) that describes how a system changes over time based on the values of its variables and the relationship between them. They are often used to study the dynamics of system with multiple interacting components, where the behavior of the system as a whole cannot be understood by looking at each component separately. When these equations are mathematically modeled, it is a strong possibility that it is usually difficult to solve these equations analytically. Therefore, it would be challenging for researchers to identify analytical or exact solutions to these differential equations. Advanced numerical approaches are shown to be the most efficient in these situations for providing an accurate numerical solution to these differential equations.

Partial differential equations (PDEs) have been used extensively to model complex phenomena like fluid flows, elastic elasticity, and gene mutation. The complicacy of these equations is that, these are complex and difficult to derive their analytical solutions. To address these difficult problems, advanced numerical techniques including differential quadrature, collocation, and finite difference have been developed. The differential quadrature method (DQM), one of these methods, has been shown to be extremely effective in getting numerical solutions to various kinds of PDEs. DQM, which employs the linear summation of a function's values at certain discrete grid points over the issue domain, can be thought of as an approximation to a function's derivative. A set of basis functions are used to determine the proper weighing factors in the DQM approximation. The weighing coefficients can be calculated using a variety of basis functions.

In this research work, three different techniques, such as Standard B-spline basis function with DQM, Crank-Nickolson with DQM using standard B-spline basis function, and particle swarm optimisation approach with differential quadrature method using exponential B-spline basis function, have been developed. In this study, modified cubic B-spline basis functions and exponential B-spline basis functions are used to obtain the weighing coefficient for DQM to improve the accuracy of Crank-Nickolson with the differential quadrature method and particle swarm optimisation with the differential quadrature method. These developed techniques have been implemented on non-linear PDEs to obtain accurate solutions.

The obtained results evaluate the effectiveness of the DQM and highlight the capability of our method. This finding could be useful in solving a wide range of differential equations encountered in various fields of research.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who have supported and guided me throughout the journey of completing this thesis.

First and foremost, I am profoundly thankful to my Supervisor, Dr. Geeta Arora for their invaluable guidance, encouragement, and unwavering support. Their expertise and insights have been instrumental in shaping this work.

I extend my sincere appreciation to the Head of School Dr. Kailash Juglan and Head of Department of Mathematics Dr. Kulwinder Singh, Dr. Deepak Kumar, Dr. Pankaj Kumar and Dr. Gurpreet Singh Bhatia of my thesis, for their valuable feedback and constructive criticism. Your input has immensely contributed to the refinement of this research.

I am indebted to the faculty Dr. Rekha Sharma, Dr. Preeti Kalra, Dr. Ratesh Kumar, Dr. Sachin Kaushal, Dr. Anjani Shukla and Dr. Pankaj Panday of Lovely Professional University for providing a conducive academic environment and access to resources essential for my research.

Iam profoundly grateful to my parents Mr. Samod Prakash Mishra and Mrs. Mamta Mishra, and beloved brother Mr. Shivansh Mishra, Vinesh, Saurav Mishra, Abhay Bajpai and Shubham Thakur whose unwavering support, sacrifices, and endless encouragement have been my guiding light throughout this academic journey. Your belief in my abilities has been a constant motivation, and I owe all my achievements to your love and guidance.

To my dear friends Jaya Gupta, Humaira Aassad, Gurpreet Kaur, Navneet Kaur, Samridhi Sud, Divya, Amandeep Kaur and Sapana Yadav and your friendship has been a source of joy, laughter, and strength. Your understanding during my busiest and most stressful times did not go unnoticed, and your encouragement kept me going. Each one of you played a unique role in making this journey memorable, and I cherish the moments we've shared.

Thank you, Mom and Dad, for being my pillars of strength, and thank you, my dear friends, for being my source of joy. Your presence in my life is my most cherished treasure.

Last but not least, I want to express my gratitude to all the participants and individuals who contributed to my research. Your willingness to share your knowledge and experiences has been invaluable.

With heartfelt gratitude,

Shubham Mishra

CONTENTS

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Boundary points	BP
Boundary value problem	BVP
Bose Eistein condensation	BEC
Crank Nicolson	CN
Differential Equations	DE’s
Differential Quadrature Method	DQM
Exponential B-spline	EBS
Exponential modified cubic B-spline	EMCBS
Fishers Equation	FE
Fishers Reaction Diffusion Equations	FRDE
Finite difference method	FDM
Finite volume method	FVM
Initial value problem	IBP
Korteweg-de Vries equation	KdV
Kadomtsev-Petviashvili	KP
Linear equations	LE
Modified cubic basis function	MCBF
Nonlinear Evolutionary Equations	NLEEs
Newmann’s condition	NCs
Ordinary Differential Equations	ODEs
Partial Differential Equations	PDEs
Particle Swarm Optimization	PSO
Radial Basis Function	RBF
Reaction diffusion equation	RDE
Sine Gordon	SG

CHAPTER-1

A COMPREHENSIVE INTRODUCTION TO NLEES

1. INTRODUCTION

Differential equations are powerful mathematical tools that relate a function to its derivatives, accounting for the behavior of the function with respect to other independent variables like time, position, or temperature. The process of solving a DEs involves determining a function that meets an equation's conditions while satisfying any given initial or boundary constraints, making it a crucial process for understanding and predicting the behavior of complex systems. In the engineering and science domains, different mathematical models are expressed using various ODEs and PDEs. These differential equations are essential tools to define other phenomena in the science area, for instance, electromagnetic fields, quantum mechanics, fluid flow diffusion processes, the physical laws of structural mechanics, and many more. Because of the importance of these differential equations in diverse branches of engineering and sciences and due to the vast applications of such equations in distinct fields related to science and engineering, a variety of ODEs and PDEs have been solved by researchers. For the purpose of getting solutions to such differential equations, different analytical and numerical methods have been in progress. When a mathematical model of such equations is framed, there are chances that, most of the time, it is bothersome to solve such equations explicitly or analytically. Therefore, fetching analytical or exact solutions to such differential equations is quite challenging for researchers. This problem indicates a need to find a solution to these differential equations by implementing some different techniques; in such situations, advanced numerical methods emerge as the most powerful tool to obtain an accurate numerical solution to these differential equations.

In the absence of an analytical solution, an extensive variety of phenomena must be evaluated numerically. Due to the intricacy of the equations, solving nonlinear PDEs can be difficult. The significance and simplicity of numerical approaches are emphasized at that time. To determine the outcomes of such complex-natured PDEs that are closest to the true value, many numerical approaches have been developed. The solutions to these kinds of PDEs are frequently approximated using numerical techniques like the "finite difference method," "finite element method," and "differential quadrature method," among others. These techniques involve discretizing the problem's domain and employing iterative algorithms to find an approximation that solves the equation while allowing for a certain amount of error.

Numerical analysis is a branch of mathematics that focuses on developing efficient numerical methods to address such difficult mathematical issues. The present work enhanced numerical solutions of nonlinear PDEs using DQM with B-spline basis functions, the Crank Nickolson method, and the particle swarm optimization approach. Initial research on the differential quadrature method focused on the latter and computed the weighing coefficients using a large number of B-spline basis functions.

The goal is to create a numerical approach for solving nonlinear PDEs. The different equations, including the KdV, Burgers, and Fisher’s equations, are essential for structuring a variety of mathematical models, which increases their significance in scientific study. Researchers can better comprehend these equations behavior and prospective applications by devising a numerical method to solve them, which will ultimately help in the creation of innovative answers to diverse scientific and engineering problems.

1.1 NON-LINEAR EVOLUTION EQUATION

Nonlinear evolution equations (NLEEs) are partial differential equations that dynamically, or in terms of both space and time, and through nonlinear systems, explain nonlinear science. Nonlinear evolutionary equations are called "nonlinear" because they involve nonlinear functions of the variables, which can lead to complex and often unpredictable behaviour of the system. They are also called "evolutionary" because they describe how the system evolves, taking into account the effects of various factors that influence the system's behaviour. NLEEs dynamically describe nonlinear sciences, the space and time two-dimensional system through the nonlinear systems. NLEEs are a class of nonlinear PDEs that have solitons as the solution and a number of other significant characteristics. Nonlinear dispersion, pattern formation, solitons, and other nonlinear processes in physics, chemistry, biology, and ecology are all described by these equations. As a result, it is characterized as a one-of-a-kind instrument for understanding scientific and technological phenomena.

1.1.1 Different types of solutions of nonlinear evolutionary equations

NLEEs have an extensive range of applications in chemistry, biology, physics, and engineering. Here are some applications of these equations based on their different types of solutions:

1.1.1.1 Periodic solutions: A periodic solution is one that repeats itself after a certain amount of time. Examples of equations with periodic solutions include the Kuramoto-Sivashinsky, Ginzburg-Landau, and Swift-Hohenberg equations. They are commonly used to model natural phenomena like flame fronts, chemical reactions, and turbulence in fluid dynamics. Identifying and analysing periodic solutions is essential to understanding the behavior of complex systems in nature.

1.1.1.2 Travelling wave solutions: Travelling wave solutions are types of soliton solutions that maintain their shape and speed while propagating through space. Equations such as KdV, and SG have travelling wave solutions. They are made to simulate how light and waves move through optical fibers and through water, respectively. Travelling wave solutions also explain the spread of epidemics and nerve impulses.

1.1.1.3 Shock waves: A shock wave is a sudden change in the amplitude or speed of a wave. Equations such as the Burgers equation and the Riemann problem have shock wave solutions. The Burgers equation, which has shock wave solutions, is used to model traffic flow and the behavior of fluids under high pressure. In addition, shock wave solutions are also used to explain the behavior of supernovae and other explosive events.

1.1.1.4 Multi soliton solutions: Multi solitons are solutions consisting of multiple solitons that maintain their shape and speed over long distances. Equations with multi soliton solutions include the KdV, and SG equations. Multi soliton solutions have applications in the study of magnetic flux tube dynamics, Bose-Einstein condensate behavior, and quantum fluid dynamics.

In summary, NLEEs can have a variety of solution types, including periodic solutions, travelling wave solutions, shock waves, soliton solutions, chaotic solutions, and multi-soliton solutions. The specific type of solution that arises depends on the physical system being modelled and the properties of the equation governing that system.

1.1.2 Types of nonlinear Evolutionary Equations

There are many examples of nonlinear evolutionary equations that are used to model a extensive range of biological, physical, and social phenomena. Here are a few examples:

1.1.2.1 The Fisher-Kolmogorov equation: This is a nonlinear PDE that is used to model the spread of a population over time. It is often used to study the dynamics of biological populations, such as the spread of a disease or the growth of a population.

$$u_t = vu_{xx} + \rho f(u), t \geq 0, x \in (-\infty, \infty) \qquad (1.1)$$

Where u is the velocity, x represents the coordinates in space, t denotes the time, v is a constant that denotes the coefficient of diffusion, f depicts nonlinear reaction term and ρ is a term for reaction factor.

1.1.2.2 The Korteweg–de Vries equation: This is a nonlinear PDEs that describes the propagation of waves in certain types of media, such as shallow water waves. It is used in diverse applications, including plasma physics, fluid dynamics, and optics.

$$\frac{\partial u}{\partial t}(x,t) + \varepsilon u(x,t)\frac{\partial u}{\partial x}(x,t) + \mu\frac{\partial^3 u}{\partial x^3}(x,t) = 0, \qquad a \leq x \leq b, t > 0 \qquad (1.2)$$

When a, b stands for the range being considered, u is the velocity, x stands for the coordinates in space, t is for time, and ε and μ are positive parameters.

1.1.2.3 The Schrodinger equation: This is a nonlinear PDE that describes the behavior of quantum mechanical systems. It has applications in disciplines including chemistry, materials science, and electronics and is used to study the behavior of particles at the atomic and subatomic level.

$$u_{tt} - u_{xx} + \sin(u) = 0, -\infty < x < \infty, \quad 0 \leq t < \infty \qquad (1.3)$$

where u is the velocity, t represents the time and x denotes the space coordinate in the direction of propagation.

1.1.2.4 Burgers Equation: Burger's equation is an important nonlinear equation that is relevant in a variety of disciplines of study. It is one of the simplest models, or PDEs, for waves with diffusive terms in liquid components in nonlinear mathematical models. Burgers created this equation in 1948 to link the two inverse effects of convection and dispersion in order to obtain insight into the study of turbulence. On the other hand, turbulence is more perplexing because it is both 3D and clearly irregular. Burgers one-dimensional equation can be found in many physical problems, including models of traffic flow and 1D turbulence, as well as waves in fluid-filled elastic tubes and shock and sound waves in viscous media.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2} \tag{1.4}$$

Where u is the velocity, x represents the coordinates in space, t denotes the time, and v depicts the kinetic viscosity or the diffusion coefficient.

1.1.3 Importance of the Non-linear Evolution Equation

Analytical solutions to nonlinear evolution equations (NLEEs) are essential in nonlinear physical research because they may accurately explain a variety of real phenomena, including vibrations, solitons, and finite-speed propagation. Exact solutions may be used to explain the key characteristics in diverse scientific, technological, and engineering applications, and they can also be used to develop and solve NLEEs when evaluating computer algebra software packages. Many chemistry, physics, and biological equations have empirical parameters or empirical functions, which is crucial. Exact solutions help researchers plan and execute experiments in order to ascertain these parameters or functions by simulating natural situations. However, not all interest equations can be solved. As a result, familiarity with all conventional and newly created methods for solving these models, as well as the development of new approaches, has become increasingly crucial. As a result, there has been a lot of research into developing techniques to solve not just NLEEs but also other forms of ODEs and PDEs.

The study of these non-linear evolution equations (NLEEs) also plays an important role in various scientific applications, like **biophysics** [1], The DNA lattice is used as an example in the study of soliton. A protein experiences structural changes that lead to intracellular communication as it gets closer to soliton. The transmission of solitons on the DNA lattice is shown in Feynman diagrams, which represent the persistence of cellular life [2]. Additionally, solitary waves are used to look at a number of biophysical processes. A single wave also appears in the investigation of the DNA molecule's nonlinear dynamics [3]. A field theory both classical and quantum field theory rely heavily on solitons and their relatives, such as instantons [4]. Field theory is made simple to understand by topological solitons [5], such as monopoles, kinks, vortices, and skyrmions.

Plasma Physics: The research on solitary waves is linked to the idea of plasma physics, which involves a high number of charged particles. The KdV equation, for example, defines the local ion density, which reflects the shift in charge from neutral. The KP equation, variations of the KdV equation, and the KP equation are additional equations used to investigate plasma physics and explain solitons and solitary wave solutions. Furthermore, soliton in plasma is investigated in a variety of settings, including the interaction of soliton in collision less plasma [6], soliton stability in hydrodynamics and plasma [7], and ion-acoustic solitons in plasma [8].

Fluid dynamics: Fluid dynamics problems are studied using solitary waves. Solitary waves may be found in deep water, as demonstrated by Vladimir Zakharov's work [9]. In order to examine these waves, it is necessary to develop the nonlinear Schrodinger equation (NLSE). Numerous fluid dynamics models, such as the theory of non-propagating surface waves, dispersive shock waves, and tidal bores, have created solitary wave solutions. Solenoids [10], small amplitude gravity capillary waves as envelope soliton solutions [11], and the soliton-mean field theory to describe the propagation of solitons in macroscopic hydrodynamic flow are all examples of solenoids.

Optical fiber: Light propagates as solitons in optical fibers [12]. Solitons, or data packets, are used for the optical cable transmission of the data. Solitons enable a high-speed, high-bandwidth network since they move at the speed of light. High-speed optical fibre communications can benefit from optical soliton due to this characteristic [13]. Furthermore, it has applications in a number of disciplines connected to fiber optics, such as soliton photonic switches, which exploit the mechanism of spatial soliton position shift following collision for optical switching. Optical logic gates may also be designed using soliton trapping in optical fiber [14].

Josephson Junctions: The Josephson junction [15] is a nonlinear oscillator made up of two weakly coupled superconductors divided by a small nonconductive layer that allows electrons to pass through. The propagation of electromagnetic waves between two superconductors may be analyzed as solitary waves. Mechanical circuits, such as SQUIDs (superconducting quantum interference devices), are made with these junctions. The Bose Einstein condensate method (BEC) was first demonstrated by Bose and Einstein in 1924. At very low temperatures, a small fraction of the particles in a diluted base gas can exist in the same quantum state, known as BEC. The macroscopic dynamics of BEC near absolute zero are simulated using the Gross-Pitaevskii equation [16, 17]. BECs were accomplished experimentally in 1995 [18, 19] by cooling atoms of diluted alkali vapours to incredibly low temperatures on the order of fractions of kelvin.

1.2 SOLITONS

In the shallow water of the Great Britain Canal in 1834, a Scottish naval engineer named John Scott Russell [1] observed a "Great wave of translation." Solitons are a type of non-dispersive long wave that travels in packet form at a constant velocity. They are also known as permanent-shaped shallow water waves. When a soliton collides with another soliton, its form remains constant, which is a unique characteristic. Solitons have attracted the attention of physicists, mathematicians, and engineers owing to their tenacity and practical utility. Soliton is a solution for nonlinear PDEs. The types of solitons are Kink solitons, breather solitons, gap solitons, envelope solitons, and solitary waves with discontinuous derivatives.

This research explores better numerical approximations of non-linear PDEs by implementing DQM based on different kinds of test functions, including B-spline basis functions. This research work intends to fetch numerical approximations of non-linear PDEs using different test functions, including a class of B-spline basis functions. For instance, the main core of this research work is finding a numerical approximation of the KdV equation, Burgers equation and Fishers equation regarding one and two dimensions. The importance of these equations increases as they serve the purpose of framing different mathematical models and have meaningfulness in other branches of science and engineering. Literature is explored in-depth to gain knowledge regarding the equations mentioned above. A brief literature review of such equations is discussed ahead.

1.3 DIFFERENTIAL QUADRATURE METHOD

To get approximation findings of nonlinear PDEs, several numerical approaches has been developed during the last few decades, for example “finite difference method”, “finite element method”, “finite volume method”, “collocation method”, etc. However, one of the most advanced and useful numerical approaches for obtaining the results of various nonlinear PDEs is DQM.

DQM is a well-known numerical technique, is employed to solve partial differential equations (PDEs). Bellman and Casti [20, 21] presented this technique in the 1970s. This technique underwent revision in the 1980s [22] and shown to be a useful numerical approach to issues in the physical and engineering sciences [23]. Due to its properties of quick convergence, high accuracy, and computational power, it is currently a well-known numerical approach. Professor Chang Shu [24] has produced a book on DQM and its use in general engineering up to the year 1999. The weighting coefficient formulation enhanced by

Quan and Chang [25] which is the most important component of DQM. It is effectively applied to estimate the weighting coefficients using a variety of basis functions, including B-spline functions [26], Lagrange interpolation polynomials, Fourier expansion-based functions, polynomial-based functions [27], radial basis functions [28], trigonometric B-spline functions [29], exponential B-spline functions [30], hyperbolic B-spline functions, sinc function etc.

A number of basis functions have been used in literature with DQM. The B-spline DQM is a significant numerical approach to obtain solutions for PDEs. There has been a lot of work described in the literature for obtaining numerical approximations of nonlinear PDEs using various types, orders, and degrees of B-spline DQM, some of which are presented here. Bashan et al. presented modified cubic B-spline [38], modified quintic B-spline [39], and Crank-Nicolson quintic B-spline [40] with DQM. Korkmaz and Dag presented cubic B-Spline[32], quartic B-spline [41] and sinc functions [42] with DQM. Tamsir et al. [43, 44] presented exponential modified cubic B-spline DQM. Arora et al. [36], [45], [46] presented modified trigonometric B-spline DQM. Shukla et al. presented cubic B-splines [47], exponential modified cubic B-spline [48] DQM. Kapoor et al. presented modified uniform algebraic hyperbolic (UAH) tension B-spline [49], modified quartic hyperbolic B-spline [50] and Barycentric Lagrange interpolation basis [51] with DQM. Kumar et al. [52] used radial basis functions with DQM.

The differential quadrature method (DQM) is a well-known numerical methodology [31] that evolved from the Gauss quadrature numerical integration approach, which approximates a defined integral using a weighted sum of integrand values at a set of nodes in the form of:

$$\int_a^b f(x)dx \approx \sum_{j=1}^{N} w_j f(x_i)$$

Nodes are $x_i; i = 1,2,\dots,N$ and weight coefficients are w_j. Bellman et al. [21] expanded Gauss quadrature to determine the derivatives of all kinds of orders of differentiable functions, which they named differential quadrature. DQM allows a space derivative to be expressed as a weighted sum of the values of the functions at the knots (similar to nodes in Gaussian quadrature) across the domain.

The fact that a function $u(x,t)$ may be approximated using interpolation at N discrete grid points is well-known. The function $u(x,t)$ can be defined in the form:

$$u(x,t) = \sum_{j=1}^{N} c_j u(x_i,t), 1 \le i \le N$$

$u(x_i,t)$ is the value of the function $u(x,t)$ at any knot point x_i in the domain $[a,b]$, where c_j is the basis function. Since of DQM, the first derivative of the same function may be expressed as,

$$\frac{\partial u(x_i,t)}{\partial x} \approx \sum_{j=1}^{N} a_{ij} u(x_i,t), 1 \le i \le N$$

Similar to the first derivative, the second derivative can be computed as

$$\frac{\partial^2 u(x_i,t)}{\partial x^2} \approx \sum_{j=1}^{N} b_{ij} u(x_i,t), 1 \le i \le N$$

where a_{ij} and b_{ij} are weight coefficients to be calculated using an appropriate technique. Several partial and ordinary differential equations have been solved using DQM, with weight coefficients computed using Lagrange polynomials [32], cubic B-splines [33], modified cubic B-splines [34, 35], trigonometric B-splines [36], quintic B-splines [37], and so on.

1.4 B-SPLINE BASIS FUNCTION

Mathematical models of most of the sciences and engineering problems and phenomena are expressed in terms of DEs, either ordinary or partial, and as a system of initial and boundary value problems, either linear or nonlinear. These equations have been intensively investigated in the literature as per their applicability in engineering and distinct areas such as quantum mechanics, electromagnetic fields, fluid flow diffusion, etc. Since finding an analytical solution to these equations is difficult, advanced numerical methods must be utilized. There are a variety of numerical techniques used for solving PDEs, including the collocation method, DQM, and FEM. When it comes to numerical aspects, in the subject of approximation theory, For the purpose of addressing BVP and PDEs, the B-spline basis function plays a key role. In the realm of mathematics, Schoenberg [53] developed the B-spline ('B' stands for Basis) in 1946, characterizing a uniform piecewise polynomial approximation. The term "B-spline" is an abbreviation for "basis spline." A least supported spline function in terms of degree, smoothness, and domain partition is B-Spline.

Consider the problem of finding a polynomial that passes through the points whose function values are given. For only two points, there exists a linear polynomial, but if the number of points doubles, a cubic polynomial can be fitted, and so on. Thus, with an increase in the number of data points, the degree of polynomial also increases. Thus, with a large number of points, a higher degree of polynomial emerges, which is difficult to work with. This situation can be tackled with the use of a piecewise polynomial. A piecewise polynomial is a polynomial that approximates the function over some part of the domain. This approximation allows us to construct an exact approximation, but since the portion of approximated polynomials is not smooth, the obtained function is not smooth at the point where the two piecewise polynomials are joined. To resolve this problem, splines are used.

The knot x_i sequence is used in order to define the basic function. Let X be a set of consisting of $N + 1$ non-decreasing real numbers $x_0 \leq x_1 \leq x_2 \leq \ldots \leq x_{n-1} \leq x_n$. The set X signifies the working region of the domain to define the B-spline basis, the i^{th} knot span is defined by the half interval $[x_i, x_{i+1}]$ and the knot series represents the working region of the real number line. The knot vectors or the knot series are called as ***uniform*** if the knots are uniformly distributed (i.e., $x_{i+1} - x_i$ is a constant for $0 \leq i \leq n$); otherwise, it is said to be ***non-uniform***. Having degree r, Each B-spline function of covers $r + 1$ knots or r intervals. Schumaker [54] expressed the B-spline basis functions based on the concept of divided differences. Cox [55] and Boor[56] separately discovered a recurrence relation for calculating B-spline basis functions, in the 1970s. The following discussed formula for the p^{th} B-spline basis function having r^{th} degree proposed by Boor in a recursive way by utilising Leibnitz' theorem as follows:

$$B_{p,r}(x) = V_{p,r}B_{p,r-1}(x) + \left(1 - V_{p+1,r}\right)B_{p+1,r-1}(x), for\ r \geq 1 \qquad (1.5)$$

Here, $V_{p,r} = \left(\frac{x-x_r}{x_{p+r}-x_r}\right)$

Cox de-Boor recursion formula expressed as in above form. Here $B_{p,r}(x)$ define a p^{th} B-spline basis function having degree r^{th}. By using this formula, it is shown that B-spline basis functions of any degree can be defined as a linear combination of basis functions of lower degrees.

Zero-degree B-spline is whose value is 1 in a half open internal and otherwise it is zero defined as a linear combination of basis functions of lower degrees.

By putting $r = 1$ in the recursive formula using zero-degree B-spline formula. **B-spline** having **First degree** also called as **linear B-spline**.

The value of the function can be obtained for linear B-spline can be represented as:

$$B_{i,1}(x) = \begin{cases} \frac{x - x_i}{x_{i+1} - x_i} & x \in [x_i, x_{i+1}] \\ \frac{x_{i+2} - x}{x_{i+2} - x_{i+1}} & x \in [x_{i+1}, x_{i+2}] \\ 0 & otherwise \end{cases} \qquad (1.6)$$

Using the value of ***linear B-spline*** and for $r = 2$ in the recursive formula, B-spline basis function of **degree second** given as:

$$B_{i,2}(x) = \begin{cases} \frac{(x-x_i)^2}{(x_{i+2}-x_i)(x_{i+1}-x_i)} & x \in [x_i, x_{i+1}] \\ \frac{(x-x_i)(x_{i+2}-x)}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} + \frac{(x_{i+3}-x)(x-x_{i+1})}{(x_{i+3}-x_{i+1})(x_{i+2}-x_{i+1})} & x \in [x_{i+1}, x_{i+2}] \\ \frac{(x_{i+3}-x)^2}{(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})} & x \in [x_{i+2}, x_{i+3}] \\ 0 & otherwise \end{cases} \quad (1.7)$$

The ***cubic B-spline basis function*** is third degree B-spline is given by formula

$$B_{i,3}(x) = \frac{1}{h^3}\begin{cases} (x-x_{i-2})^3 & x \in [x_{i-2}, x_{i-1}] \\ (x-x_{i-2})^3 - 4(x-x_{i-1})^3 & x \in [x_{i-1}, x_i] \\ (x_{i+2}-x)^3 - 4(x_{i+1}-x)^3 & x \in [x_i, x_{i+1}] \\ (x_{i+2}-x)^3 & x \in [x_{i+1}, x_{i+2}] \\ 0 & otherwise \end{cases} \quad (1.8)$$

From the definition given by (4), at the nodal points the values of $B_i(x)$ can be obtained, on differentiating with respect to x first and second derivative values of $B_i(x)$ can be obtained. Table 1.1 provides the value, $u = \frac{3}{h}$, at the nodal points the values of $B_i(x)$ and its derivatives.

Table 1.1: At the nodal points, value of $B_i(x)$for cubic B-spline and its derivatives.

x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$B_i(x)$	0	1	4	1	0
$B_i'(x)$	0	u	0	$-u$	0
$B_i''(x)$	0	$\frac{2}{3}u^2$	$-\frac{4}{3}u^2$	$\frac{2}{3}u^2$	0

The **fourth degree** B-spline basis function known as ***quartic B-spline*** is given by

$$B_{i,4}(x) = \frac{1}{h^4}\begin{cases} (x-x_{i-2})^4 & x \in [x_{i-2}, x_{i-1}] \\ (x-x_{i-2})^4 - 5(x-x_{i-1})^4 & x \in [x_{i-1}, x_i] \\ (x-x_{i-2})^4 - 5(x-x_{i-1})^4 + 10(x-x_i)^4 & x \in [x_i, x_{i+1}] \\ (x_{i+3}-x)^4 - 5(x_{i+2}-x)^4 & x \in [x_{i+1}, x_{i+2}] \\ (x_{i+3}-x)^4 & x \in [x_{i+2}, x_{i+3}] \\ 0 & otherwise \end{cases} \quad (1.9)$$

This basis function is non-zero on five knot spans. From the definition given by (5), the values of $B_i(x)$at the nodal points can be obtained. On differentiating with respect to x, Its three derivative values can be obtained in an identical approach. Using the value $v = \frac{4}{h}$, at the nodal points, the value of $B_i(x)$ and its derivatives may be tabulated as in Table 1.2.

Table 1.2: At the nodal points, value of $B_i(x)$ for quartic B-spline and its derivatives.

x	x_{i-3}	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}	x_{i+3}
$B_i(x)$	0	1	11	11	1	1	0
$B_i'(x)$	0	v	$3v$	$-3v$	v	$-3v$	0
$B_i''(x)$	0	$\frac{3}{4}v^2$	$-\frac{3}{4}v^2$	$-\frac{3}{4}v^2$	$\frac{3}{4}v^2$	$\frac{3}{4}v^2$	0
$B_i'''(x)$	0	$\frac{3}{8}v^3$	$-\frac{9}{8}v^3$	$\frac{9}{8}v^3$	$\frac{3}{8}v^3$	$-\frac{3}{8}v^3$	0

Extending the definition, the quintic B-spline basis function can be defined as follows:

$$B_{i,5}(x) = \frac{1}{h^5}\begin{cases} (x-x_{i-3})^5 & x_i \in [x_{i-3}, x_{i-2}] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 & x_i \in [x_{i-2}, x_{i-1}] \\ (x-x_{i-3})^5 - 6(x-x_{i-2})^5 + 15(x-x_{i-1})^5 & x_i \in [x_{i-1}, x_i] \\ (x_{i+3}-x)^5 - 6(x_{i+2}-x)^5 + 15(x_{i+1}-x)^5 & x_i \in [x_i, x_{i+1}] \\ (x_{i+3}-x)^5 - 6(x_{i+2}-x)^5 & x_i \in [x_{i+1}, x_{i+2}] \\ (x_{i+3}-x)^5 & x_i \in [x_{i+2}, x_{i+3}] \\ 0 & otherwise \end{cases} \quad (1.10)$$

The definition provides the values of $B_{i,5}(x)$ at different values of nodes. In an identical approach, the value of its four derivatives can be obtained on differentiating with respect to x. Table 1.3 provides the value of derivatives considering $\omega = \frac{5}{h}$.

Table 1.3: At the nodal points, value of $B_i(x)$for quintic B-spline and its derivatives.

x	x_{i-3}	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}	x_{i+3}
$B_i(x)$	0	1	26	66	26	1	0
$B_i'(x)$	0	ω	10ω	0	-10ω	$-\omega$	0
$B_i''(x)$	0	$\frac{4}{5}\omega^2$	$\frac{8}{5}\omega^2$	$-\frac{24}{5}\omega^2$	$\frac{8}{5}\omega^2$	$\frac{4}{5}\omega^2$	0
$B_i'''(x)$	0	$\frac{12}{25}\omega^3$	$-\frac{24}{25}\omega^3$	0	$\frac{24}{25}\omega^3$	$-\frac{12}{25}\omega^3$	0
$B_i^{iv}(x)$	0	$\frac{24}{125}\omega^4$	$-\frac{96}{125}\omega^4$	$\frac{144}{125}\omega^4$	$-\frac{96}{125}\omega^4$	$\frac{24}{125}\omega^4$	0

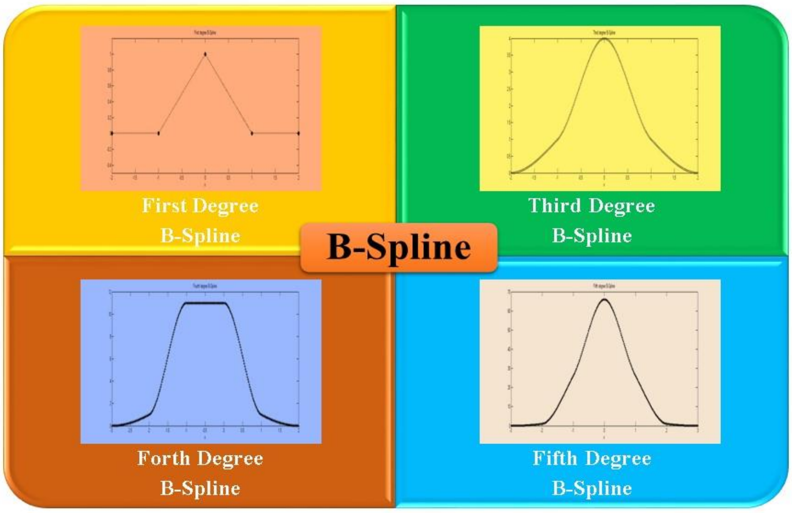


Figure 1.1 Different kinds of B-spline basis function.

The degree of B-spline can be extended as sixth degree, seventh degree, eighth degree, ninth degree and so on. Use of B-spline up to tenth degree exist in literature. Other than the standard B-spline there are different types of B-splines which can be used according to the considered function such as trigonometric B-spline, exponential B-spline, and so on.

1.5 TRIGONOMETRIC B-SPLINE

A trigonometric B-spline $T_i(x)$, is characterized as a spline function that has minimum supports for a given degree, smoothness, and space partition. Having degree r, Each trigonometric B-spline function covers $r+1$ knots or r intervals and is defined as follows [46], [57]:

$$T_i^r(x) = \frac{\sin\frac{x-x_i}{2}}{\sin\frac{x_{i+r-1}-x_i}{2}} T_i^{r-1}(x) + \frac{\sin\frac{x_{i+r}-x}{2}}{\sin\frac{x_{i+r}-x_{i+1}}{2}} T_{i+1}^{r-1}$$

for $r = 2,3,4, \ldots .$

This recurrence relation demonstrates that the trigonometric B-spline basis elements of a higher degree can be steadily assessed using of basis of lower degree. $T_i(x)$ is a piecewise trigonometric function with geometric properties like non-negativity, partition of unity and C^∞ continuity.

1.5.1 Three degree or Cubic Trigonometric B-Spline

The cubic trigonometric B-spline basis function $T_i(x)$, for $i = -1,0,\ldots,N+1$ can be expressed in the following form [58] :

$$T_i^3(x) = \frac{1}{x}\begin{cases} p^3(x_i), & x \in [x_i, x_{i+1}] \\ p(x_i)\big(p(x_i)q(x_{i+2}) + q(x_{i+3})p(x_{i+1})\big) + q(x_{i+4})p^2(x_{i+1}), & x \in [x_{i+1}, x_{i+2}] \\ q(x_{i+4})\big(p(x_{i+1})q(x_{i+3}) + q(x_{i+4})p(x_{i+2})\big) + p(x_i)q^2(x_{i+3}), & x \in [x_{i+2}, x_{i+3}] \\ q^3(x_{i+4}), & x \in [x_{i+3}, x_{i+4}] \\ 0, & otherwise \end{cases} \quad (1.11)$$

Where $p(x_i) = \sin\left(\frac{x-x_i}{2}\right), q(x_i) = \sin\left(\frac{x-x_i}{2}\right), \omega = \sin\left(\frac{h}{2}\right)\sin(h)\sin\left(\frac{3h}{2}\right)$ and $h = \frac{b-a}{n}$.

1.6 EXPONENTIAL B-SPLINE:

For a uniform mesh Γ with the knots x_i defined on $[a, b]$ if $B_i(x)$ be the B-splines at the points of Γ together with knots x_i, $i = -3, -2, -1, N+1, N+2, N+3$ outside the interval $[a, b]$ and a finite support on each of the four consecutive intervals $[x_i + rh, x_i + (r+1)h]_{r=-3}^{0}$, $i = 0, \dots, N+2$. The function formula can be expressed as [59, 60]:

$$B_i(x) = \begin{cases} b_2\left[(x_{i-2} - x) - \frac{1}{p}\Big(\sinh\big(p(x_{i-2} - x)\big)\Big)\right] & x \in [x_{i-2}, x_{i-1}] \\ a_1 + b_1(x_i - x) + c_1 e^{p(x_i - x)} + d_1 e^{-p(x_i - x)} & x \in [x_{i-1}, x_i] \\ a_1 + b_1(x - x_i) + c_1 e^{p(x - x_i)} + d_1 e^{-p(x - x_i)} & x \in [x_i, x_{i+1}] \\ b_2\left[(x - x_{i+2}) - \frac{1}{p}\Big(\sinh\big(p(x - x_{i+2})\big)\Big)\right] & x \in [x_{i+1}, x_{i+2}] \\ 0 & \text{otherwise.} \end{cases} \quad (1.12)$$

here,

$$p = \max_{0 \le i \le N} p_i, s = \sinh(ph), c = \cosh(ph)$$
$$b_2 = \frac{p}{2(phc - s)}, a_1 = \frac{phc}{phc - s}, b_1 = \frac{p}{2}\left[\frac{c(c-1) + s^2}{(phc - s)(1 - c)}\right],$$
$$c_1 = \frac{1}{4}\left[\frac{e^{-ph}(1-c) + s\big(e^{-ph} - 1\big)}{(phc - s)(1 - c)}\right], d_1 = \frac{1}{4}\left[\frac{e^{ph}(c-1) + s\big(e^{ph} - 1\big)}{(phc - s)(1 - c)}\right]$$

$B_i(x)$ is twice continuously differentiable. From the Table 1.4, the values of $B_i(x), B_i'(x)$ and $B_i''(x)$ at the knots x 's are obtained.

x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$B_i(x)$	0	$\frac{s - ph}{2(ph - s)}$	1	$\frac{s - ph}{2(phc - s)}$	0

$B'_i(x)$	0	$\frac{p(1-c)}{2(ph-s)}$	0	$\frac{p(c-1)}{2(phc-s)}$	0
$B''_i(x)$	0	$\frac{p^2 s}{2(phc-s)}$	$\frac{-p^2 s}{phc-s}$	$\frac{p^2 s}{2(phc-s)}$	0

Table 1.4: Exponential B-spline values

1.6.1 Exponential Cubic B-Spline:

For the uniform mesh $a = x_0 < x_1 < \cdots < x_n = b$, the exponential cubic B-spline, $B_i(x)$, is presented as a piecewise polynomial function as given below:

$$B_i(x) = \begin{cases} b_2\left((x_{i-2} - x) - \frac{1}{p}\left(\sinh\left(p(x_{i-2} - x)\right)\right)\right) & [x_{i-2}, x_{i-1}] \\ a_1 + b_1(x_i - x) + c_1 \exp\left(p(x_i - x)\right) + d_1 \exp\left(p(x_i - x)\right) & [x_{i-1}, x_i] \\ a_1 + b_1(x - x_i) + c_1 \exp\left(p(x - x_i)\right) + d_1 \exp\left(-p(x - x_i)\right) & [x_i, x_{i+1}] \\ b_2\left((\mathrm{x} - x_{i+2}) - \frac{1}{p}\left(\sinh\left(p(x - x_{i+2})\right)\right)\right) & [x_{i+1}, x_{i+2}] \\ 0 & \text{otherwise} \end{cases} \tag{1.13}$$

here,

$$a_1 = \frac{phc}{phc - s}, b_1 = \frac{p}{2}\left(\frac{c(c-1) + s^2}{(phc - s)(1 - c)}\right), b_2 = \frac{p}{2(phc - s)},$$

$$c_1 = \frac{1}{4}\left(\frac{\exp\,(-ph)(1 - c) + s(\exp\,(-ph) - 1)}{(phc - s)(1 - c)}\right),$$

$$d_1 = \frac{1}{4}\left(\frac{\exp\,(ph)(c - 1) + s(\exp\,(ph) - 1)}{(phc - s)(1 - c)}\right).$$

and $s = \sinh\,(ph), c = \cosh\,(ph)$ and p is a free parameter.

$\{B_{-1}(x), B_0(x), \cdots, B_{i+1}(x)\}$ is a basis defined over the interval $[a, b]$. Each basis function $B_i(x)$ is continuously differentiable twice. The values of $B_i(x), B'_i(x)$ and $B''_i(x)$ at the knots x_i can be computed from Eq. (1.13) and are documented in Table 1.5.

Table 1.5: Values of $B_i(x)$ and its first and second derivatives at the knot points

x	x_{i-2}	x_{i-1}	x	x_{i+1}	x_{i+2}
$B_i(\mathrm{w})$	0	$\frac{s-ph}{2(phc-s)}$	1	$\frac{s-ph}{2(phc-s)}$	0

$B_i'(\mathrm{w})$	0	$\frac{p(1-c)}{2(phc-s)}$	0	$\frac{p(c-1)}{2(phc-s)}$	0
$B_i''(\mathrm{w})$	0	$\frac{p^2 s}{2(phc-s)}$	$\frac{-p^2 s}{phc-s}$	$\frac{p^2 s}{2(phc-s)}$	0

that should be determined in computations.

1.7 DQM WITH B-SPLINE BASIS FUNCTIONS

The DQM is a numerical technique for solving nonlinear PDEs using B-spline functions as the basis functions. The algorithm for the DQM with B-spline basis functions is as follows:

Step 1 Discretize the domain: Divide the domain of the nonlinear PDEs into a set of discrete points. The points should be evenly spaced to simplify the calculations.

Step 2 Define the B-spline basis functions: A suitable B-spline basis functions are defined as piecewise polynomial functions that satisfy certain continuity conditions. These functions are used to approximate the unknown function and its derivatives at the discrete points.

Step 3 Approximation: Approximate the unknown function and its derivatives at each of the discrete points using a weighted sum of the function values at these points, where the weights are determined by the B-spline basis functions.

Step 4 Derivative evaluation: Evaluate the derivatives of the unknown function using the approximation obtained in step 3.

Step 5 System of equations: Transform the PDE into a system of algebraic equations by applying the approximation and derivative evaluation.

Step 6 Solve the system of equations: Solve the system of algebraic equations to obtain the values of the unknown function at the discrete points.

Step 7 Calculating the errors: Interpolate the solution obtained in step 6 to obtain the values of L_2 and L_∞ errors.

1.8 CONVERGENCE ANALYSIS

To analyze the convergence of the scheme, let $\overline{U}$ be the modified cubic B-spline interpolant of $u \in C^6[x_L, x_R]$. It satisfies the following interpolation condition [122]:

$$\overline{U}(x_i) = u(x_i), i = 1,2, \dots \dots., n$$

and end conditions given as:

$$\overline{U}''(x_L) = u''(x_L) - \frac{h^2}{12}u^4(x_L), \overline{U}''(x_R) = u''(x_R) - \frac{h^2}{12}u^4(x_R),$$

Thus

$$\overline{U}'(x_i) = u'(x_i) + o(h^4), 0 \le i \le n \tag{1.14}$$

$$\overline{U}''(x_i) = u''(x_i) - \frac{h^2}{12}u^4(x_i) + 0(h^2), 0 \le i \le n \tag{1.15}$$

1.8.1 Theorem: Let $u \in C^6[x_L, x_R]$ The r^{th} derivatives of u $(r = 1,2)$ is approximated as reported in (1.1) have the following error bounds

$$u'(x_j) = U'(x_j) + o(h^4), \tag{1.16}$$

and

$$u''(x_j) = U''(x_j) + o(h^2), \tag{1.17}$$

Proof. Let $\overline{U}$ be the modified cubic B-spline interpolant of $u \in C^6[x_L, x_R]$ that satisfies the interpolation and end conditions as discussed above. The triangle inequality yields:

$$|u_i' - U_i'| \le \left|u_i' - \overline{U}_i'\right| + \left|\overline{U}_i' - U_i'\right| \tag{1.18}$$

On utilizing (1.7) and (1.8), we get

$$|u_i' - U_i'| \le \left|u_i' - \overline{U}_i'\right| + O\left({h_y}^4\right), \tag{1.19}$$

Now,

$$\overline{U}_i' - U_i' = \sum_{k=1}^{n} \delta_k (\varphi_k')_i - \sum_{j=1}^{n} P_{ij}^{(1)} u(x_j, t)$$

$$= \sum_{k=1}^{n} \delta_k \sum_{j=1}^{n} P_{ij}^{(1)} (\varphi_k')_i - \sum_{j=1}^{n} P_{ij}^{(1)} u(x_j, \tau)$$

$$= \sum_{j=1}^{n} P_{ij}^{(1)} \left(\sum_{k=1}^{n} \delta_k (\varphi_k')_i - u(x_j, \tau) \right) \tag{1.20}$$

$$= \sum_{j=1}^{n} P_{ij}^{(1)} \left(U(x_j, t) - u(x_j, \tau) \right)$$

Utilizing interpolations conditions leads to:

$$\overline{U}_i' - U_i' = 0 \tag{1.21}$$

After utilizing (1.19), equation (1.20) transforms to

$$|u_i' - U_i'| \leq O\left({h_y}^4\right) \tag{1.22}$$

Similarly, the result of (1.18) can be proved.

1.9 STABILITY OF METHOD

By utilizing the differential quadrature method on partial differential equations, a set of ODEs is obtained that can be solved using any numerical methods such as the Runge-Kutta (R-K) method. To ensure stability while implementing the R-K method, conditions for eigenvalues are employed as given in the text [183]. On reducing the partial differential equation to a system of ordinary equations, the equation takes the form:

$u_t(x, t) = M + f\left(u(x, t)\right),$

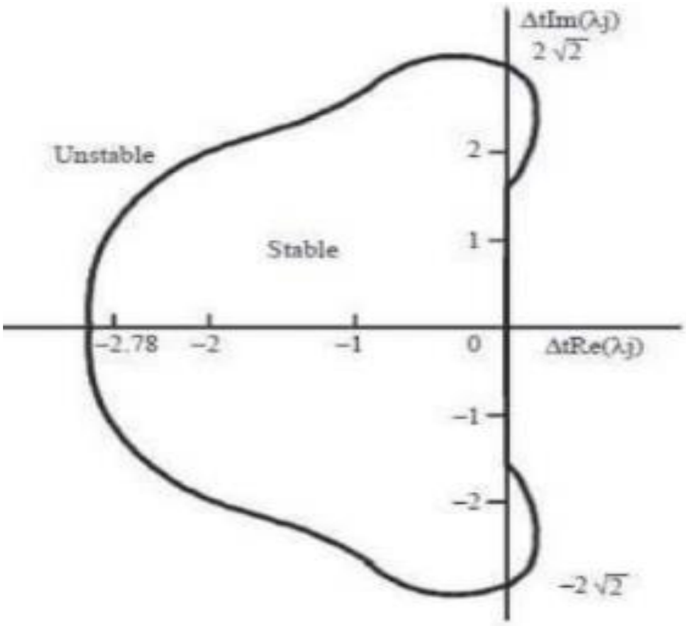


Figure 1.2 Region of stability

where M is a matrix derived from the partial differential equation and $f(u(x,t))$ represents the nonlinear terms. The eigenvalues of matrix M determine whether the system is robust. There are certain specific conditions that need to be satisfied depending on the value of time step Δt as discussed below:

(a) Real λ : $-2.78 < \Delta t\lambda < 0$

(b) Pure Imaginary: $-2\sqrt{2} < \Delta t\lambda < -2\sqrt{2}\ \ \lambda$

(c) Complexλ : λΔt, lies inside the region R (shown in Figure 1.1)

The system is stable only if the eigenvalues fall within certain ranges, as shown in Figure 1.2. It is essential to consider the time step value when determining the stability of the system.

1.10 ERROR NORMS

To ensure the accuracy of developed numerical technique, the findings are verified by comparing the approximations to both exact solutions and previously published numerical solutions. To accomplish this developed technique, various measures of error norms are utilized in the proposed research, and some of the most important formulas used to compute numerical errors are:

$$L_\infty = \max\left(\left|u_{exact}(x_{i,}t) - u_{numerical}(x_{i,}t)\right|\right);$$

$$L_2 = \sqrt{h\sum_{i=1}^{N}\left|u_{exact}(x_{i,}t) - u_{numerical}(x_{i,}t)\right|^2};$$

1.11 OBJECTIVES OF THE PROPOSED RESEARCH WORK

1. To study the application of B-spline basis function in use with different numerical techniques.
2. To solve one dimensional nonlinear evolution equations (NLEEs) with B-spline basis function in differential quadrature method.
3. To solve the two-dimensional nonlinear evolution equations with adequate B-spline basis function.
4. To explore the solution of nonlinear evolution equations using different forms of available B-spline basis function such as exponential, hyperbolic and cardinal basis functions.

1.12 LAYOUT OF THE THESIS

This thesis is structured into five chapters, providing a comprehensive and impressive exploration of solitons and their application in science and engineering. The brief layout of each chapter is discussed as follows:

In Chapter 1, A comprehensive introduction to NLEEs. In this chapter, solutions of non-linear NLEEs have been discussed, whose solutions are in the form of solitons, by implementing a quintic b-spline basis function using DQM. This chapter also includes the literature regarding the origin of the DQM. Furthermore, the developed formulae of "exponential cubic B-spline DQM" and its related methodology are also mentioned in this chapter. The error formulae to check the robustness of the schemes are also mentioned in this chapter. This chapter provides an overview of the concepts that are employed throughout the entire thesis.

In Chapter 2, Solution of non-linear evolution equation by implementing of quintic B-Spline basis function using differential quadrature method. The understanding of phenomena in science and technology depends on nonlinear evolution. The Korteweg-de Vries equation (KdV) is a prime example of a periodic solution equation that finds application across a number of scientific domains. To solve the KdV equation, a quintic B-spline basis function is presented using the differential quadrature method.

In Chapter 3, Fishers equation and Burgers equation using Crank Nickolson (CN) and Differential Quadrature Method (DQM). Nonlinear evolution is essential for understanding phenomena in science and technology. In this work, a hybrid scheme is developed called Crank Nickolson DQM for determining the solution of Fishers and Burgers equation by using a modified cubic B-spline as a basis function. To check the authenticity of the developed technique, the above method is implemented on the Fishers and Burgers equation. The authenticity and effectiveness of this methodology are shown by the findings, which are equivalent to those found in the literature and close to an exact solution. The results obtained are presented in the form of figures and tables.

In Chapter 4, Fisher 1D and 2D equation using Particle Swarm Optimization (PSO) and Differential Quadrature Method (DQM). The vast majority of real-world problems may be formulated as optimization problems. A tool that has been utilized to approach problems across several domains is particle swarm optimization. This chapter includes the differential quadrature method using an exponential basis function with a particle swarm optimization technique for finding the solution of Fisher one-dimensional and two-dimensional equations whose solutions exhibit the soliton form and have numerous applications.

In Chapter 5, Conclusion and Future scope. In the last chapter Conclusion and Future scope of the research work is discussed.

CHAPTER 2

SOLUTION OF NON-LINEAR EVOLUTION EQUATIONS BY IMPLEMENTING OF QUINTIC B-SPLINE BASIS FUNCTION USING DIFFERENTIAL QUADRATURE METHOD

2.1 Introduction

While performing studies to identify the most effective design for canal boats on the Edinburgh-Glasgow canal in 1844, John Scott Russell noticed a phenomenon. He noticed that after one or two miles, the height of water in the canal steadily decreases as it travels along the watercourse. He invented the term "Wave of Translation" to describe this unique and wonderful phenomenon. This gives rise to the soliton defined as a wave with a defined shape traveling at a constant speed through a given medium. The first wave to exhibit characteristics similar to a soliton was observed by Russell [1]. This was the beginning of an absolutely specific field of research to which scientists and mathematicians have contributed a lot over time. Nowadays it is known that many equations have soliton solutions. Some of the equations having soliton solution are KdV equation, Fisher's equation, NLS equation etc.

The Korteweg-de Vries (KdV) equation is a nonlinear partial differential equation developed by Korteweg and de Vries in 1895 with respect to plasma waves [70] and then again by Washimi and Taniuti [71] to study acoustic waves in a cold plasma. The KdV equation is used to examine the propagation of low-amplitude water waves in shallow water bodies. The solution to this equation produces solitary waves [72].

The KdV equation is given by.

$$\frac{\partial U}{\partial t}(x,t) + \varepsilon U(x,t)\frac{\partial U}{\partial x}(x,t) + \mu\frac{\partial^3 U}{\partial x^3}(x,t) = 0, \qquad a \le x \le b, t > 0 \tag{2.1}$$

When a, b stands for the range being considered, u is the velocity, x stands for the coordinates in space, t is for time, ε and μ are positive parameters.

The KdV equation is a third-order nonlinear evolution equation that characterizes long waves and is widely used in physical and engineering disciplines. For example, it is used in modeling ionic-acoustic solitons in plasma physics [73], in the study of a long wave in subsurface oceans, and shallow sols in geophysical fluid dynamics [74, 75]. It also describes the phenomenon in cluster physics and super deformed nuclei [76, 77], quantum field theory and classical general relativity [78]. The solution of the KdV equation has opened enormous possibilities for mathematical concepts.

The solutions of nonlinear equations are always of interest to researchers as they are studied using various approaches [79, 80]. In most cases, an analytical solution is not accessible, so numerical aspects are always necessary [81]. Gardner et al [82] proved both the existence and uniqueness of solutions to the KdV equation. Liu [83] provided an elliptic Jacobi function solution for the KdV equation. In the same research paper, Hufford and Xing [84] reported a numerical solution for the linearized version of the problem as well as super convergence for the approach used. Trogdon and Deconinck presented [85] a finite-genus solution to the equation. Grava and Klein [86] solved the KdV equation numerically and asymptotically for a small dispersion limit. Leach [87] gives the large-time evolution of the generalized Korteweg-de Vries equation. The wavelet Galerkin approach is used by Kumar and Mehra [88] to find a time accurate solution of this equation. To solve this equation, Bahadir [89] uses an exponential finite difference technique. Aksan and Ozde Å [90] use the Galerkin finite element approach with B-spline functions. This equation was solved numerically and analytically by Ozde Å and Kutluay [91]. Ascher and McLachlan [92] provided a multi-symplectic box technique for the KdV equation. Small time solutions of the equation were given by Kutluay et al [93]. Idrees et al [94] use the optimal homotopic asymptotic technique to solve this equation. To solve the KdV equation numerically, Gucuyenen and Tanoglu [95] used the iterative splitting approach. Sarma [96] provided a solitary wave solution for this equation. Barbera [97] used a variational methodology, which was further investigated by Yuliawati et al [70] using the steepest descent approach, to study the solution of the KdV equation in the Hamiltonian condition. In addition, there have been several other successful numerical approaches to the KdV equation, including the spectral method [98], the pseudo-spectral method [81], and the collocation method [99].

2.2 Numerical Scheme

The differential quadrature method involves estimating a derivative of a given function using linear summation of its components at different nodes of the problem domain. The domain [a, b] can be simply partitioned into uniformly distributed finite nodes x_i with distance h, such that

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b.$$

Let $B_i(x)$ be the quintic B-splines with knots at points x_n, $n = 0, 1, 2, \dots, N$. The arrangement of splines $\{B_{-1}, B_0, B_1, \dots, B_N, B_{N+1}\}$ forms the basis for any function on $[a, b]$. For $i = 1, 2, \dots, N+1$, the solution at each time point of the node x_i is $U(x_i, t)$. The estimated derivative parameters are calculated as follows:

$$U_x = \sum_{j=1}^{N} p_{ij} u(x_j, t), U_{xx} = \sum_{j=1}^{N} q_{ij} u(x_j, t), U_{xxx} = \sum_{j=1}^{N} r_{ij} u(x_j, t), \quad (2.2)$$

for $i = 1, 2, \dots, N+1$. The derivatives are approximated by $p_{ij}, q_{ij}, and\ r_{ij}$. Once the values of p_{ij} are fixed as described in the next section, the weighting coefficients $q_{ij}, and\ r_{ij}$ can be easily calculated.

The method for calculating the other coefficients is as follows:

$$\frac{\partial^2 u_i}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = \sum_{k=1}^{N} p_{ik} \left(\frac{\partial u}{\partial x}\right)_{x=x_k} = \sum_{k=1}^{N} p_{ik} \left(\sum_{j=1}^{N} p_{kj} u(x_j, t)\right)$$
$$= \sum_{k=1}^{N} \sum_{j=1}^{N} p_{ik} p_{kj} u(x_j, t) = \sum_{j=1}^{N} q_{ij} u(x_j, t), i = 1,2,3, \dots, N+1.$$

Since q_{ij} are calculated, using this the values for p_{ij}, r_{ij} can be calculated in a similar manner.

For $i = -2, -1, 0, \dots, N+2$, $B_i(x)$, the quintic B-spline basis function, describes a piecewise defined function with the properties of continuity and division of unity. The following equations can be used to calculate the basis functions.

$$B_i(x) = \frac{1}{h^5} \begin{cases} (x - x_{i-3})^5 & x \in [x_{i-3}, x_{i-2}] \\ (x - x_{i-3})^5 - 6(x - x_{i-2})^5 & x \in [x_{i-2}, x_{i-1}] \\ (x - x_{i-3})^5 - 6(x - x_{i-2})^5 + 15(x - x_{i-1})^5 & x \in [x_{i-1}, x_i] \\ (x_{i+3} - x)^5 - 6(x_{i+2} - x)^5 + 15(x_{i+1} - x)^5 & x \in [x_i, x_{i+1}] \\ (x_{i+3} - x)^5 - 6(x_{i+2} - x)^5 & x \in [x_{i+1}, x_{i+2}] \\ (x_{i+3} - x)^5 & x \in [x_{i+2}, x_{i+3}] \\ 0 & otherwise, \end{cases} \quad (2.3)$$

where $B_{-2}, B_{-1}, B_0, B_1, \ldots\ldots B_{N+1}, B_{N+2}$ are basis formed over the region $a \le x \le b$. Each quintic B-spline covers six elements, so that a total of six quintic B-splines cover one element. Table 2.1 summarizes the values of $B_i(x)$ and the first four derivatives.

Table 2.1 Values of $B_i(x)$ and its derivatives at the nodes.

x	x_{i-3}	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}	x_{i+3}
$B_i(x)$	0	1	26	66	26	1	0
$B_i'(x)$	0	$5/h$	$50/h$	0	$-50/h$	$5/h$	0
$B_i''(x)$	0	$20/h^2$	$40/h^2$	$-120/h^2$	$40/h^2$	$20/h^2$	0
$B_i'''(x)$	0	$60/h^3$	$-120/h^3$	0	$120/h^3$	$-60/h^3$	0
$B_i^{IV}(x)$	0	$120/h^4$	$-480/h^4$	$720/h^4$	$-480/h^4$	$120/h^4$	0

The first-order approximation of the derivative can be estimated using the relation:

$$B_i'(x_i) = \sum_{j=1}^{N} p_{ij} B_i(x_j) \text{ for } i = 1,2,\ldots,N. \tag{2.4}$$

As a result, a matrix system emerges as:

$$A\vec{p}[i] = \vec{s}[i]$$

here A is the coefficient matrix given by:

$$\begin{bmatrix} 66 & 26 & 1 & 0 & 0 & 0 & \cdot & 0 \\ 26 & 66 & 26 & 1 & 0 & 0 & \cdot & 0 \\ 1 & 26 & 66 & 26 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & 1 & 26 & 66 & 26 & 1 \\ 0 & \cdot & 0 & 0 & 1 & 26 & 66 & 26 \\ 0 & \cdot & 0 & 0 & 0 & 1 & 26 & 66 \end{bmatrix}$$

that represents the vector, corresponding to node point x_i. The unknown coefficients are $\vec{p}[i] = [p_{i1}, p_{i2}, \ldots, p_{iN}]^T$, $i = 1, 2, \ldots, N$, with right hand side given as:

$$\vec{s}[1] = [0, f, g, 0, \ldots, 0]^T$$
$$\vec{s}[2] = [-f, 0, f, g, 0, \ldots, 0]^T$$
$$\vec{s}[3] = [-g, -f, 0, f, g, \ldots, 0]^T$$
$$\cdot$$
$$\cdot$$
$$\vec{s}[N-2] = [0, \ldots, -g, -f, 0, f, g]^T,$$
$$\vec{s}[N-1] = [0, \ldots, 0, -g, -f, 0, f]^T,$$
$$\vec{s}[N] = [0, \ldots, 0, -g, -f, 0]^T$$

Here, $f = \frac{50}{h}$ and $g = \frac{5}{h}$.

The coefficients $p_{i1}, p_{i2}, \dots, p_{iN}$ for $i = 1, 2, \dots, N$. were calculated using MATLAB 2014 to solve the given five band matrix system. Substituting approximate values for the derived first and third order spatial derivatives in equation (2.1) yields the following system:

$$u_t = -\epsilon u \sum_{j=1}^{N} p_{ij} u_j - \mu \sum_{j=1}^{N} r_{ij} u_j \,. \tag{2.5}$$

The SSP-RK45 scheme [104] is then used to solve this system of ordinary differential equations, which offers numerical solutions at various time levels.

2.3 Numerical experiments

In this section the accuracy the proposed method is shown by calculating the L_2 *and* L_∞ errors.

The lowest three invariants related to mass, momentum, and energy conservation are also calculated by the following equations:

$$I_1 = \int_a^b U dx, I_2 = \int_c^b U^2 dx, I_3 = \int_a^b \left[U^3 - \frac{3\mu}{\varepsilon}(U')^2\right] dx.$$

2.3.1 Experimental evaluation of single soliton

Consider the KdV equation with exact solution given as [42] follows:

$$U(x,t) = 3C \operatorname{sech}^2(Ax - Bt + D), \tag{2.6}$$

here

$$A = \frac{1}{2}(\varepsilon C/\mu)^{1/2} \text{ and } B = \frac{1}{2}\varepsilon C(\varepsilon C/\mu)^{1/2},$$

so that (2.6) offers a single soliton with amplitude $3C$ and velocity εC moving toward the right. The equation is solved with the initial state taken from analytic solution (2.6) as

$$U(x,0) = 3C\operatorname{sech}^2(Ax + D),$$

and the boundary conditions $U(0,t) = U(2,t) = 0$ for $t \geq 0$. $\varepsilon = 1, \mu = 4.84 \times 10^{-4}, C = 0.3, D = -6$ is employed in order to create a comparison with other investigations. To demonstrate the evolution of the current technique using a modified quintic B-spline DQM, Table 2 and Table 3 show the error norm and invariants values, respectively.

Table 2.2 Experimental evaluation of single soliton at $\Delta t = 0.0005$.

Method	N	T	L_2	L_∞	I_1	I_2	I_3
Present Scheme	151	0.25	2.3524×10^{-6}	6.4229×10^{-6}	0.1446	0.0868	0.0469
		0.50	4.0430×10^{-6}	1.2235×10^{-5}	0.1446	0.0868	0.0469
		0.75	6.0076×10^{-6}	1.9232×10^{-5}	0.1446	0.0868	0.0469
		1.00	8.1958×10^{-6}	2.5869×10^{-5}	0.1446	0.0868	0.0469
		2.00	4.3005×10^{-5}	9.0073×10^{-5}	0.1446	0.0868	0.0469
		3.00	8.3086×10^{-4}	0.0022×10^{-6}	0.1444	0.0868	0.0469
MQ_DQM [39]	201	0.25	1.01×10^{-5}	2.66×10^{-5}	0.1445	0.0867	0.0468
		0.50	1.11×10^{-5}	2.59×10^{-5}	0.1445	0.0867	0.0468
		0.75	1.33×10^{-5}	3.94×10^{-5}	0.1445	0.0867	0.0468
		1.00	1.43×10^{-5}	4.08×10^{-5}	0.1445	0.0867	0.0468
		2.00	2.14×10^{-5}	6.74×10^{-5}	0.1445	0.0867	0.0468
		3.00	2.86×10^{-5}	8.15×10^{-5}	0.1446	0.0867	0.0468

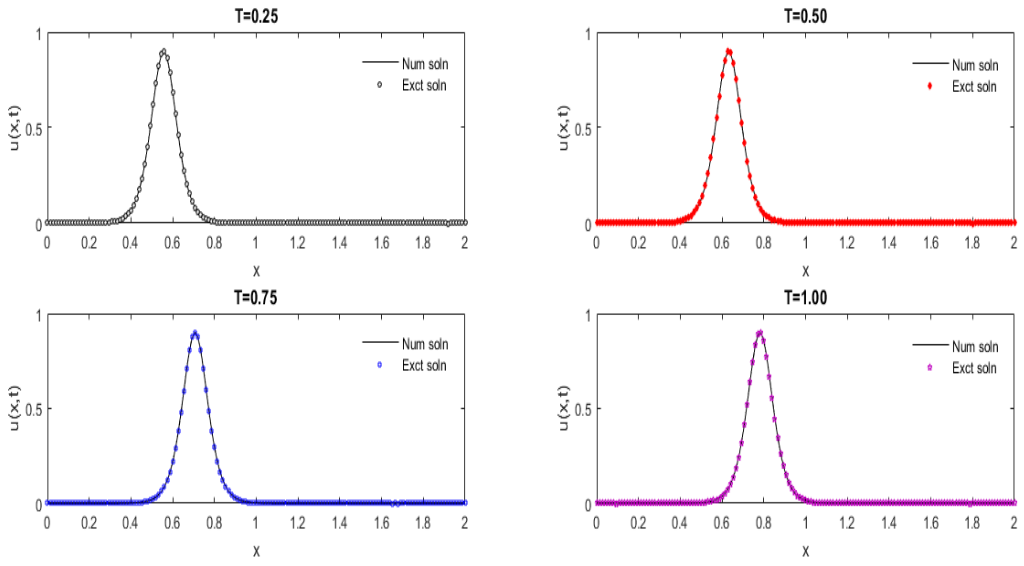


Figure 2.1 Simulations of single solitons $\Delta t = 0.0005$.

Table 2.3 Experimental evaluation of single soliton at $\Delta t = 0.001$.

Method	N	T	L_2	L_∞	I_1	I_2	I_3
Present Scheme	91	0.25	6.2478×10^{-5}	2.1816×10^{-4}	0.1446	0.0868	0.0469
		0.50	9.6636×10^{-5}	2.0265×10^{-4}	0.1446	0.0868	0.0469
		0.75	1.9160×10^{-4}	3.0993×10^{-4}	0.1446	0.0868	0.0469
		1.00	4.4543×10^{-4}	7.0164×10^{-4}	0.1446	0.0868	0.0469
MQ_DQM [39]	201	0.25	0.000010	0.000027	0.1445	0.0867	0.0468
		0.50	0.000010	0.000021	0.1445	0.0867	0.0468
		0.75	0.000012	0.000034	0.1445	0.0867	0.0468
		1.00	0.000012	0.000032	0.1445	0.0867	0.0468
[105]	200	0.25	0.00522	-			
		0.50	0.01200	-	0.144590	0.086759	0.046871
		0.75	0.01220	-			
		1.00	0.02220	-	0.144590	0.086759	0.046873

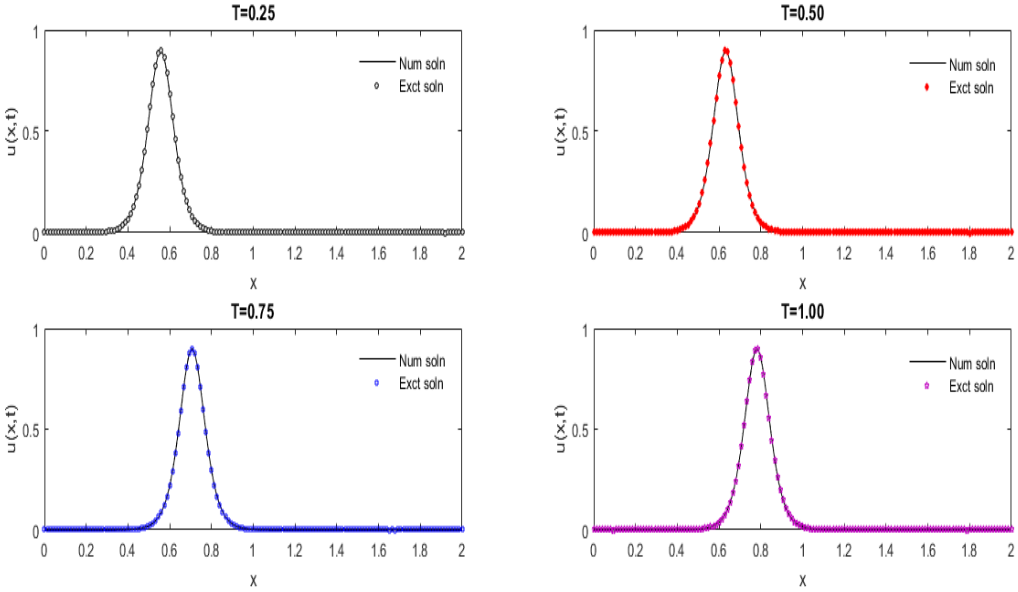


Figure 2.2 Simulations of single solitons $\Delta t = 0.001$.

2.3.2 Experimental evaluation of interaction of two solitons

Consider this second experiment [106] with initial condition stated as:

$$U(x,0) = \sum_{i=1}^{2} 3C_i \text{sech}^2\,(A_i x + x_i),\ A_i = \frac{1}{2}\left(\frac{\varepsilon C_i}{\mu}\right)^{\frac{1}{2}}, i = 1,2.$$

with boundary conditions

$$U(0,t) = U(2,t) = 0,$$

where $\varepsilon = 1, \mu = 4.84 \times 10^{-4}, C_1 = 0.3, C_2 = 0.1, x_1 = x_2 = -6$ is considered in all simulations. The same parameters as in the previous study [106] are used for numerical calculations using MATLAB R2015b (32-bit) in Windows 10 Version 21H2 for x64, with $N = 91$ and $\Delta t = 0.005$. Table 4 displays the error norm and invariants value.

Table 2.4 Experimental Evaluation of Interaction of Two Solitons at $\Delta t = 0.005$.

Method	N	T	I_1	I_2	I_3	CPU Time
Present Scheme	91	0.75	0.2281	0.1071	0.0533	0.208 sec
		1.50	0.2279	0.1071	0.0533	0.244 sec
		2.25	0.2278	0.1071	0.0533	0.283 sec

		3.00	0.2238	0.1074	0.0533	0.316 sec
MQ_DQM	91	0.75	0.2281	0.1070	0.0533	
[39]		1.50	0.2280	0.1070	0.0533	
		2.25	0.2279	0.1070	0.0533	
		3.00	0.2277	0.1070	0.0533	
MQ[106]	200	0.75	0.2280	0.1070	0.0535	
		1.50	0.2280	0.1070	0.0534	
		3.00	0.2279	0.1070	0.0532	

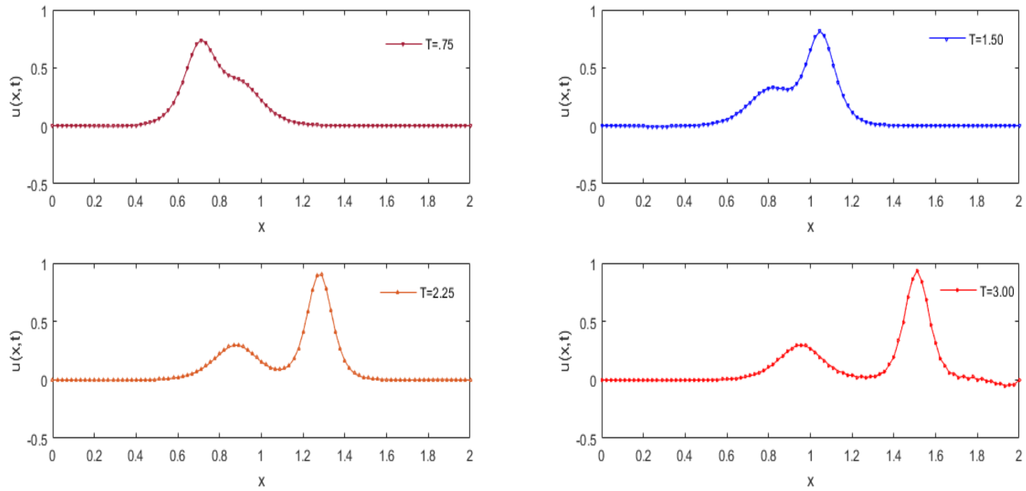


Figure 2.3 Simulations of two solitons $\Delta t = 0.005$.

2.3.3 Experimental evaluation of interaction of three solitons

The numerical solution is calculated for the interaction of three solitons having the initial condition [107] given as:

$$U(x,0) = \sum_{i=1}^{3} 12C_i^2 \text{sech}^2\left(C_i(x - x_i)\right),$$

with the zero boundary conditions for domain [-100,100] with $\varepsilon = 1.0, \mu = 1.0, C_1 = 0.3, C_2 = 0.25, C_3 = 0.2,\ x_1 = -60,\ x_2 = -44,\ x_3 = -26.$ The same parameters as in the previous study [107] at $\Delta t = 0.1$ and a much smaller number of grid points $N = 251$ than in the previous study [107] $N = 481$ were used in the numerical computations. Table 5 displays the error norm and invariants value.

Table 2.5 Experimental evaluation of interaction of three solitons at $\Delta t = 0.1$.

Method	N	T	I_1	I_2	I_3
Present Scheme	251	56	18.0004	9.8274	5.2615
		112	17.9991	9.8275	5.2633
		168	18.0029	9.8275	5.2627
		224	18.0098	9.8276	5.2624
		280	18.1445	9.8989	5.2627
MQ_DQM	481	56	18.0002	9.8273	5.2622
[39]		112	17.9994	9.8273	5.2621
		168	17.9989	9.8274	5.2623
		224	17.9988	9.8274	5.2623
		280	18.0006	9.8274	5.2623
MQ[107]	2000	56	18.0018	9.5936	5.0328
		112	17.9974	9.5138	4.9651
		168	17.9971	9.3228	4.7808
		224	17.9985	9.0697	4.5362
		280	17.9995	8.8327	4.3141

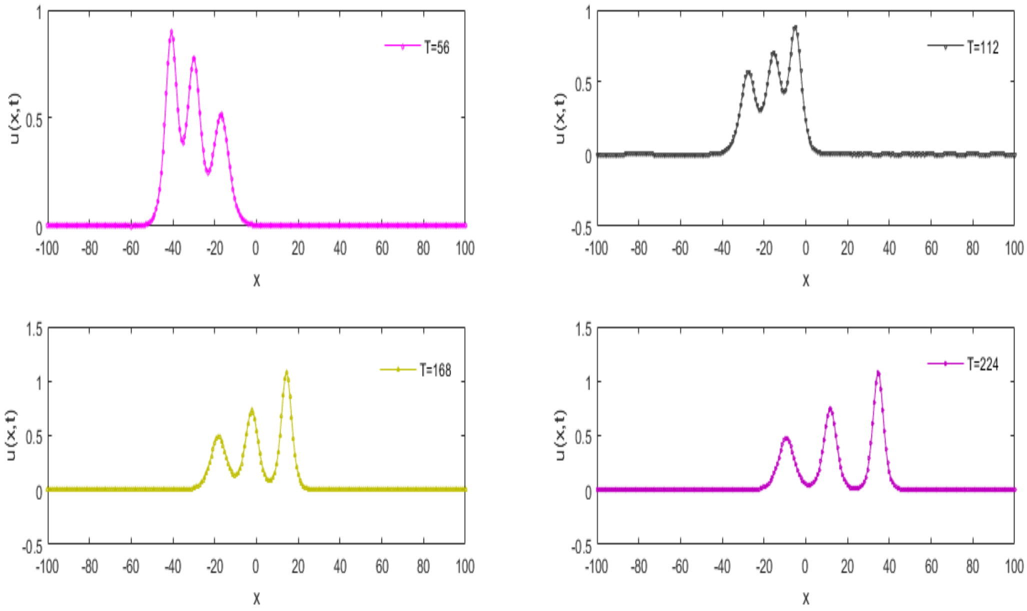


Figure 2.4 Simulations of three solitons $\Delta t = 0.1$.

2.3.4 Experimental evaluation of interaction of four solitons

In this example the interaction of four solitons is presented with initial condition [107] given as:

$$U(x,0) = \sum_{i=1}^{4} 12C_i^2 \text{sech}^2 \left(C_i(x - x_i)\right),$$

Along with zero boundary conditions for domain [-150,150] with $\varepsilon = 1.0, \mu = 1.0, C_1 = 0.3, C_2 = 0.25, C_3 = 0.2, C_4 = 0.15, x_1 = -85, x_2 = -60, x_3 = -35, x_4 = -10$.

The considered parameters are same as in the previous study [107] at $\Delta t = 0.1$ and a much smaller number of grid points $N = 401$ than in the previous study [107] $N = 451$ were used in the numerical computations. Table 6 displays the error norm and invariants value.

Table 2.6 Experimental evaluation of interaction of four solitons at $\Delta t = 0.1$.

Method	N	T	I_1	I_2	I_3	CPU Time
Present Scheme	401	80	21.6000	10.3887	5.2687	2.225 sec
		160	21.5999	10.3887	5.2683	3.819 sec
		240	21.6002	10.3887	5.2702	5.316 sec

		320	21.5992	10.3887	5.2691	7.263 sec
		400	21.5360	10.3973	5.2707	9.173 sec
MQ_DQM	451	80	21.6000	10.3887	5.2688	
[39]		160	21.6000	10.3886	5.2687	
		240	21.6000	10.3886	5.2688	
		320	21.5998	10.3887	5.2688	
		400	21.6000	10.3887	5.2688	
MQ[107]	1500	80	2.16028	9.9723	4.8594	
		160	21.6049	9.7448	4.6426	
		240	21.6011	9.7023	4.6035	
		320	21.6007	9.4774	4.3943	
		400	21.6074	9.1922	4.1368	

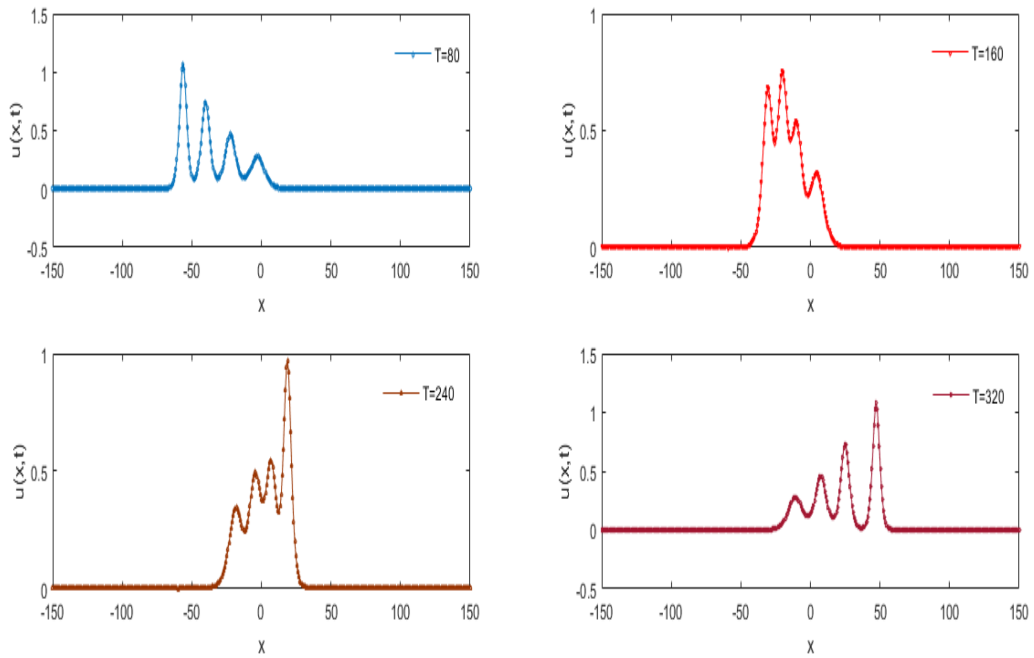


Figure 2.5 Simulations of four solitons at $\Delta t = 0.1$.

2.4 CONCLUSION

Due to the numerous applications of the KdV equation in physical phenomena, in recent years this equation has become a point of attraction for researchers who want to find a numerical solution for this equation using various methods. In this paper, the newly defined quintic B-spline basis function is presented to solve the equation using the differential quadrature method. The advantage of this approach is that it involves transforming the partial differential equation into an ordinary differential equation, which can be solved by any numerical technique for the solution of ordinary differential equations. In the present work, the SSP-RK45 is implemented to solve the obtained system of ordinary differential equations, which is a combination of the RK method of orders four and five that is a strong stability preserving scheme. Numerical results in terms of conservation variables and errors are calculated for the single soliton and extended till interaction of four solitons. The results are compared with numerical solutions from the literature. The results obtained agree well with those obtained earlier. The advantage of the proposed method is its ease of implementation compared to previous methods. Thus, the present approach can be utilized to solve a variety of nonlinear physical models with extension and application to two-dimensional problems.

CHAPTER 3

SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS USING CRANK NICKOLSON (CN) AND DIFFERENTIAL QUADRATURE METHOD (DQM)

3.1 Introduction

Differential equations are playing a significant part in applied mathematics that is involved in various computations in wave motion, wave distribution, electro-magnetic, network design, telecommunications, electronic dynamics, fluid dynamics and many related branches of engineering and sciences [108]. Researchers are looking for new methods for solving the modelled differential equation as a result of mathematical modelling utilising various analytical and numerical methodologies. But it is not always possible that the differential equation can be solved by the currents analytical techniques to yield the exact solution. There comes the role of numerical techniques that can help in solving the differential equations with the requisite accuracy.

With the growth of technology, there are numerous available choices of the finding the numerical solution of the differential equations utilising the software like MATLAB, Maple and Mathematica. Researchers are continuously working on the modification of the numerical methods to deliver the solution efficiently and precisely. One such attempt to adapt the computational numerical technique to solve the nonlinear differential equations is provided in this work.

In the present chapter, the Crank-Nicolson technique is employed to solve the Burgers equation in part A and Fishers equation in part B, followed by DQM with a modified form of cubic B-spline basis function to remove extra knot points. The equation is thus reduced to a system of equations by utilising differential quadrature technique to approximate the spatial derivatives. The next step consists of computing the solution of the produced system of equations using MATLAB programming in an iterative way. Since the present method is using the concept of differential quadrature method after discretizing the domain, hence it is acting as a tool to reduce the computational complexity of the collocation approach. As collocation method usually requires complex algebraic manual calculation.

Part A:

"NUMERICAL SIMULATION AND DYNAMICS OF BURGERS EQUATION USING MODIFIED CUBIC-B-SPLINE DIFFERENTIAL QUADRATURE METHOD."

Burgers equation is a basic partial differential equation that appears in many applications including classical dynamic, nonlinear harmonics, plasma physics, and traffic flow. Bateman [109] was the first researcher to present this equation, and Burgers [110] later examined it for its possible solution. There are various well known numerical methods reported in literature to solve Burgers equation. Some of the important numerical methods in last few years includes but are not limited to differential quadrature method [27], [43], [58], [111, 112], finite element method [113, 114], collocation approach [115, 116], and finite difference approach [93], [117–121].

Burgers equation in its general form is given as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2} \tag{3.1}$$

Where *v* depicts the kinetic viscosity or the diffusion coefficient. This form of the equation is called as viscous Burgers equation and is known as the inviscid Burgers equation in its absence.

3.2 Basis functions

Basis functions are a requirement to implement in the methodology of DQM to approximate the derivative. There are many well-known basis functions that are applied in DQM such as Lagrange's polynomial, simple polynomial and the radial basis function etc. The present study is to solve the FEs by implementing the modified form of the B-spline basis function of the third degree. This piecewise polynomial basis function of third-degree is defined on the knot intervals as follows:

$$\varphi_j(x) = \frac{1}{h^3}\begin{cases} (x_j - x_{j-2})^3 & x \in [x_{j-2}, x_{j-1}] \\ (x_j - x_{j-2})^3 - 4(x_j - x_{j-1})^3 & x \in [x_{j-1}, x_j] \\ (x_{j+2} - x)^3 - 4(x_{j-1} - x)^3 & x \in [x_j, x_{j+1}] \\ (x_{j+2} - x)^3 & x \in [x_{j+1}, x_{j+2}] \\ 0 & x \in otherwise \end{cases}$$

Where $\varphi_j(x)'s$ defines the basis for the considered domain. The basis function along with its two derivatives value at the nodal points are as follows:

$\varphi_j(x_j) = 1, \varphi_j(x_{j-1}) = 4, \varphi_j(x_{j+1}) = 4$

$\varphi_j'(x_j) = 0, \varphi_j'(x_{j-1}) = 3/h, \varphi_j'(x_{j+1}) = -3/h$

$\varphi_j''(x_j) = -12/h^2, \varphi_j''(x_{j-1}) = 6/h^2, \varphi_j''(x_{j+1}) = 6/h^2$

These basis functions are redefined at the boundary points to generate a system of equations that hold the property of being diagonally dominant. The modification in the basis functions is done at the first two and the last two knot points to deal with the contribution of the spline at the extra known points. Thus, the values of the function at the knots can be obtained as follows:

$\phi_1(x) = \varphi_1(x) + 2\varphi_0(x)$

$\phi_2(x) = \varphi_1(x) - \varphi_0(x)$

$\phi_j(x) = \varphi_j(x)\ for\ j = 3, \ldots., N-2$

$\phi_{N-1}(x) = \varphi_{N-1}(x) - \varphi_{N+1}(x)$

$\phi_N(x) = \varphi_N(x) + 2\varphi_{N+1}(x)$

It can be seen that the modifications are reported at the two boundary points on either side of the domain while the rest values remain unchanged.

3.2.1 Generated system of equations

Using the DQM approach with the modified cubic basis functions, the first derivative for the grid points x_i, $i = 1, \ldots \ldots, N$ can be written as:

$$\varphi_k' = \sum_{j=1}^{N} P_{ij}^1 \varphi_k(x_j), k = 1, \ldots \ldots.., N$$

Where, P_{ij}^1 are the unknown weighting coefficients to obtain the first derivative approximations using the reframed basis functions as defined in the above section.

Utilizing the values of $\varphi(x)$ and its derivative at the knot points leads to:

$\mathrm{X}\vec{p}[i] = Y\vec{p}[i]\ for\ i = 1, \ldots.., N$

The X represents the coefficients matrix defined as:

$X\ (i,i) = 4;\ i\ =\ 2, \ldots, N-1$

$X\ (i-1,i)\ =\ 1, i =\ 2, \ldots., N-\ 1$

$X\ (i,i+1)\ = 1, i =\ 1, \ldots, N-2$

$X\ (k.k)\ =\ 6, k\ =\ 1\ and\ k\ = N$

The weighting coefficients presented in a column form as $\vec{p}[i]$, and $\vec{Y}[i]$ is the coefficient vector that can be defined as a matrix Z with each column representing the right-hand side to find the solution for each x_i can be defined as:

$Z\,(1,1)\;=-\dfrac{6}{h};\; Z\,(2,1)\;\;=\dfrac{6}{h}$

$Z\,(i-l,i)\;=\;\dfrac{3}{h}, Z\,(i,i)\;=0, Z\,(i+1,i)\;\;=\;-\dfrac{3}{h}, i\;=\;1,2\,\ldots, N-1$

$Z\,(n,n)\;=-\dfrac{6}{h}, Z\,(n-1,n)\;=\;\dfrac{6}{h}$

$Z\,(i,j)\;=\;0,$ otherwise ∀ i and j.

The resulted tridiagonal system of equations provides the weighting coefficients p_{i1}, $p_{i2},\ldots, p_{in}$ *for* $i\;=\;1$ *to* $N.$

The obtained coefficients represent the values of P_{ij}^{1} which is used for the first derivative approximation and also for calculating the second-order derivative.

3.3 Crank Nicolson Scheme

The Crank-Nicolson scheme is a finite difference approach for numerical simulation of the differential equations. From last several years, it is implemented to solve numerous well-known equations using the collocation approach. The purpose of this scheme is the discretization of the derivative with the average of the values taken at the two successive time levels. In this work, Burgers equation is solved by applying the DQM for approximating the derivatives after implementation of the well-known crank Nicolson scheme on the time derivative as

$$u_t = \frac{u^{n+1}+u^n}{\Delta t}.$$

3.3.1 Scheme Implementation

Consider the Burgers equation (3.1) in simplified form as

$$u_t = vu_{xx} - uu_x \tag{3.2}$$

On discretizing the derivative of u with respect to t using finite forward difference approach and applying the Crank-Nicolson scheme on the right-hand side terms generate the system as follows:

$$\frac{u^{n+1}-u^n}{\Delta t} = v\left[\frac{(u_{xx})^n+u_{xx}^{n+1}}{2}\right] - \frac{(uu_x)^n+(uu_x)^{n+1}}{2} \tag{3.3}$$

$$u^{n+1} - u^n = \frac{v\Delta t}{2}[(u_{xx})^n + (u_{xx})^{n+1}] - \frac{\Delta t}{2}[(uu_x)^n + (uu_x)^{n+1}]$$

Applying the well-known quasi-linearization to linearize the nonlinear part of the equation produces $(uu_x)^{n+1}$ as:

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

Thus, the system can be simplified considering, $\alpha = \frac{v\Delta t}{2}$; $\beta = \frac{\Delta t}{2}$, and hence reduces to following form

$$u^{n+1} = u^n + \alpha[u_{xx}^n + u_{xx}^{n+1}] - \beta[u^n u_x^{n+1} + u^{n+1}u_x^n] \quad (3.4)$$

On separating the values of u at different time levels

$$u^{n+1}(1 + \beta u_x^n) - \alpha u_{xx}^{n+1} + \beta u_x^{n+1}u^n = u^n + \alpha u_{xx}^n \quad (3.5)$$

The values of the derivatives are obtained using DQM in the above system for $(n+1)^{th}$ time level followed leads to following system of equation for $i = 1$ to N:

$$(1 + \beta A_i^n)u_i^{n+1} - \alpha\sum_{j=1}^{N} q_{ij}u_j^{n+1} + \beta u_i^n \sum_{j=1}^{N} p_{ij}u_j^{n+1} = R.H.S. \quad (3.6)$$

Where $A_i^n = \sum_{j=1}^{N} p_{ij}u(x_j,t)^n$, is the approximation of the derivative at the n^{th} level. The obtained system of equation can be visualized as follows: $JU^{n+1} = R^n$

Here the matrix J can be discussed as follows:

$$J_{ij} = \begin{cases} 1 + \beta A_i + \beta u_i P_{ij} - \alpha q_{ij}\,, & for\ i = j \\ \beta u_i P_{ij} - \alpha q_{ij}\,, & for\ i \neq j \end{cases} \quad (3.7)$$

and $R^n = u^n + \alpha\sum_{j=1}^{N} q_{ij}u(x_j,t)$, can be then solved by any known method for solution of system of linear equations. Here, the Gauss-elimination method is implemented in an iterative way.

3.4 Numerical experiments

Now, three numerical examples are presented using the proposed blended method to solve Burgers equation numerically with varying sets of parameters. Standard errors are used to evaluate the performance of various approaches.

3.4.1 1st **Test problem:** Consider the equation (3.1) to be solved in the domain [0,8] with boundary condition taken from the exact solution [58], given as:

$$u(x,t) = \frac{x/t}{1+\sqrt{\left(\frac{t}{g}\right)}\exp\left(\frac{x^2}{4\nu t}\right)},\ g = \frac{0.125}{\nu} \text{ for } t \geq 1$$

And the initial solution as: $u(x,0) = \frac{x}{1+\exp\left(\frac{x^2-0.25}{4\nu}\right)}$ (3.8)

The numerical solution has been obtained for the high values of $\nu = 0.5$ at different time levels for $t \leq 5$. The domain is taken as per the literature review. There is no limitation of changing the domain but the physical behaviour can be depicted in the considered domain. Table 1, presents the comparison of the numerical and exact solutions at different points with respect to the time. In Table 2, the errors are estimated at different time levels for time-step, $k = 0.01$ and the number of domain partition as 41, thus $h = 0.2$ which is very less as compared to the domain. Figure 1 exhibits the solution behaviour to highlight the physical behaviour of the equation together with time. As can be depicted from the figure that with the advancement of time, the solution values are decreasing but satisfying the same characteristics at the boundary points.

Table 3.1 Solution of the equation at different time levels for first test problem.

X	**T=1.5**		**T=3.0**		**T=4.5**	
	Exact	Present	Exact	Present	Exact	Present
1.0	0.265771	0.265604	0.118804	0.118783	0.071869	0.071864
2.0	0.261421	0.261363	0.167623	0.167591	0.113387	0.113376
3.0	0.088070	0.088151	0.127382	0.127381	0.109491	0.109486
4.0	0.011859	0.011889	0.057975	0.057996	0.073605	0.073612
5.0	0.000741	0.000744	0.016735	0.016750	0.035717	0.035726
6.0	0.000023	0.000023	0.003238	0.003243	0.012919	0.012908

7.0	0.000000	0.000000	0.000433	0.000432	0.003582	0.003452

Table 3.2: Errors of the Burgers equation at different time levels for 1st test problem.

	T=1.5		**T=3.0**		**T=4.5**	
	Present	[112]	Present	[112]	Present	[112]
L_2	1.9281e-04	2.11e-03	4.9057e-05	1.64 e-03	3.4738e-04	1.38 e-03
L_∞	1.7771e-04	3.18e-03	3.4300e-05	1.43 e-03	5.5472e-04	0.89 e-03

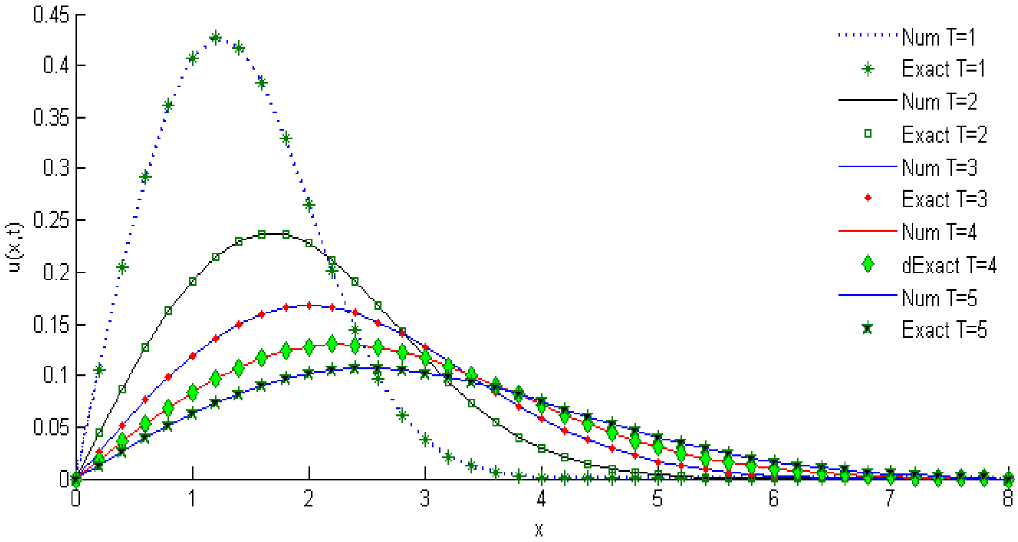


Figure 3.1 Numerical solution of the equation at different time levels.

3.4.2 2$^{\text{nd}}$ **Test problem:** Consider the Burgers equation (3.1) with domain [0, 2] with zero boundary condition and the initial condition taken from the exact solution. The solution of the equation is verified from the exact solution obtained analytically as:

$$u(x,t) \; = 2\pi\nu \frac{\sin(\gamma)e^{\beta}+4\sin(2\,\gamma)e^{-4\beta}}{4+\cos(\gamma)e^{\beta}+2\cos(2\,\gamma)e^{-4\beta}} \qquad (3.9)$$

Here $\gamma = \pi x, \beta = -\,\pi^2\nu^2 t$

The numerical solution of the equation is obtained for different time levels for $k = 0.1$ at different values of ν. In Table 3 and Table 4 the obtained numerical results are compared with the exact solutions and are presented in form of errors for $t = 0.1$ and $t = 1$ as was available in the literature. As shown by the obtained results in form of L_2 and L_∞ errors, the numerical solutions are in good agreement with the exact solution and is much better as compare to result available in literature at relatively small value of domain partition. The surface plot of the solutions with 121 nodes is presented in Figure 2 to discuss the physical behaviour of the equation along with time while the physical behaviour of the solution obtained at $t = 1$ for different values of ν is presented in Figure 3. It can be seen that the solution is showing the same behaviour with the changing of time but the amplitude of the wave is decreasing with decreasing the value of ν.

Table 3.3 Solution of the equation at time $t = 0.1$ for the 2^{nd} test problem.

T=0.1			$\boldsymbol{\nu = 10^{-2}}$		$\boldsymbol{\nu = 10^{-4}}$		$\boldsymbol{\nu = 10^{-6}}$	
	h	k	L_2	L_∞	L_2	L_∞	L_2	L_∞
Present	0.5	10^{-2}	2.23e-04	2.23e-04	2.38e-08	2.38e-08	2.39e-12	2.39e-12
[117]	0.025	10^{-3}	3.55e-03	4.41e-03	3.74e-07	4.62e-07	3.74e-11	4.62e-11
[111]	0.1	10^{-2}	3.41e-03	3.89e-03	3.56e-07	4.11e-07	3.56e-11	4.11e-11
[112]	0.025	10^{-3}	3.47e-03	4.26e-03	-	-	-	-

Table 3.4 Solution of the equation at time t=1 for the 2^{nd} test problem.

			$\nu = 10^{-2}$		$\nu = 10^{-4}$		$\nu = 10^{-6}$	
$\boldsymbol{t}$**=1**	$\boldsymbol{h}$	$\boldsymbol{k}$	$\boldsymbol{L_2}$	$\boldsymbol{L_\infty}$	$\boldsymbol{L_2}$	$\boldsymbol{L_\infty}$	$\boldsymbol{L_2}$	$\boldsymbol{L_\infty}$
Present	0.5	10^{-2}	2.16e-03	2.16e-03	2.39e-07	2.39e-07	2.39e-11	2.39e-11
[117]	0.025	10^{-3}	2.66e-02	3.13e-02	3.72e-06	4.61e-06	3.74e-10	4.62e-10
[111]	0.1	10^{-2}	2.63e-02	2.92e-02	3.55e-06	4.09e-06	3.56e-10	4.11e-10
[112]	0.025	10^{-3}	2.63e-02	3.08e-02	-	-	-	-

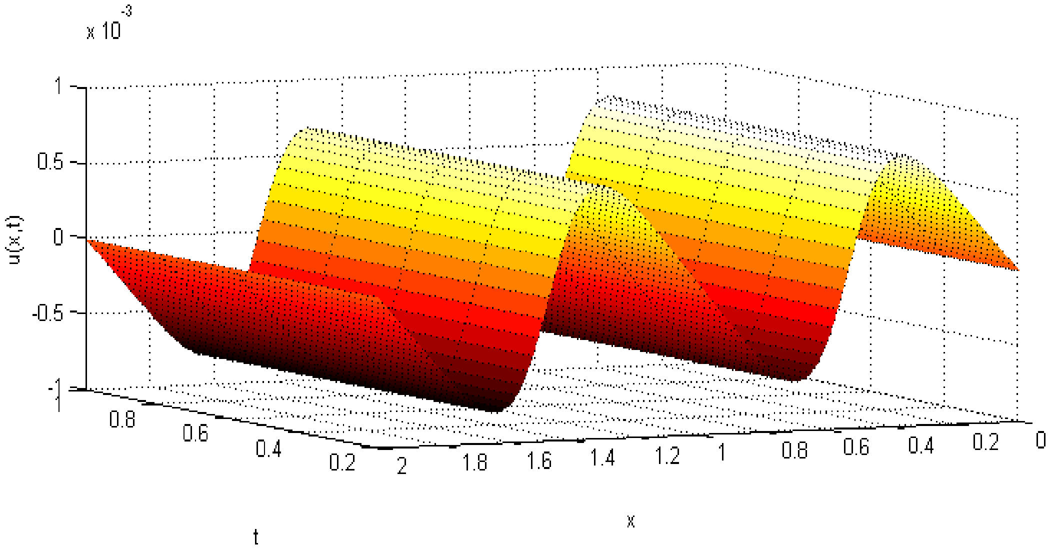


Figure 3.2 Physical behaviour of the numeric solutions of 2^{nd} test problem for $\nu = 10^{-4}$ at different time levels.

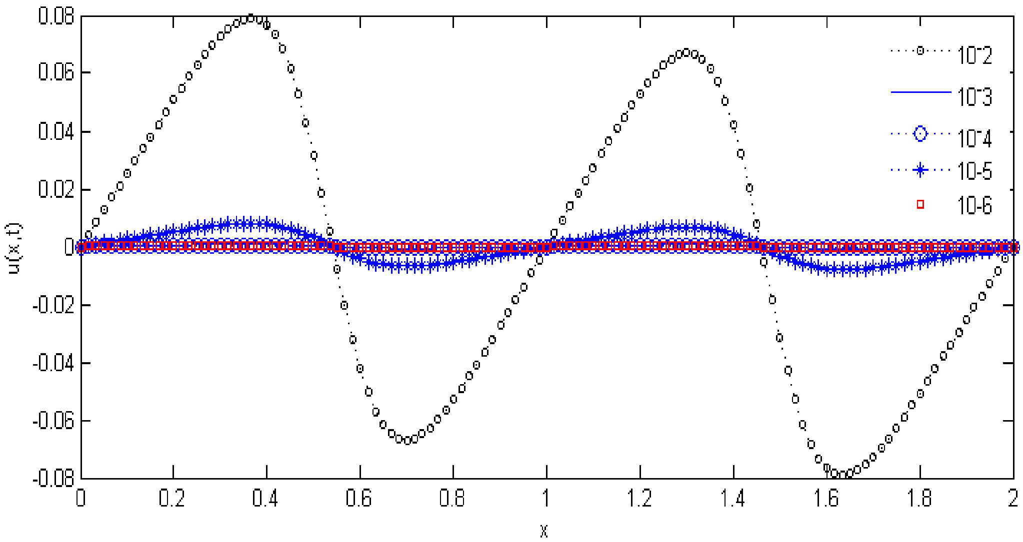


Figure 3.3 Physical behaviour of the solution of 2^{nd} test problem obtained at t=1 for different values of ν.

3.4.3 3rd **Test problem:** Consider the Burgers equation (3.1) with domain $[0, 1]$ and the boundary conditions as:

$$u(x, 0) = \sin(2\pi x) + 0.5 \sin(\pi x) \tag{3.10}$$

The equation's solution is calculated and validated with the results already presented in form of graphs [112]. The solution of the equation is calculated for the different value of time for the viscosity parameter ν with value as 0.05 for time step $k = 0.01$ with the number of partitions as 41. The results of the numerical simulation are presented for the different time-levels in Figure 4. From the figures presented under Figure 4 the physical behaviour of the solution can be seen changing the form along with the passage of time. The solution is of wave form can be seen changing from the two phases to single phase and then decreasing in amplitude with time.

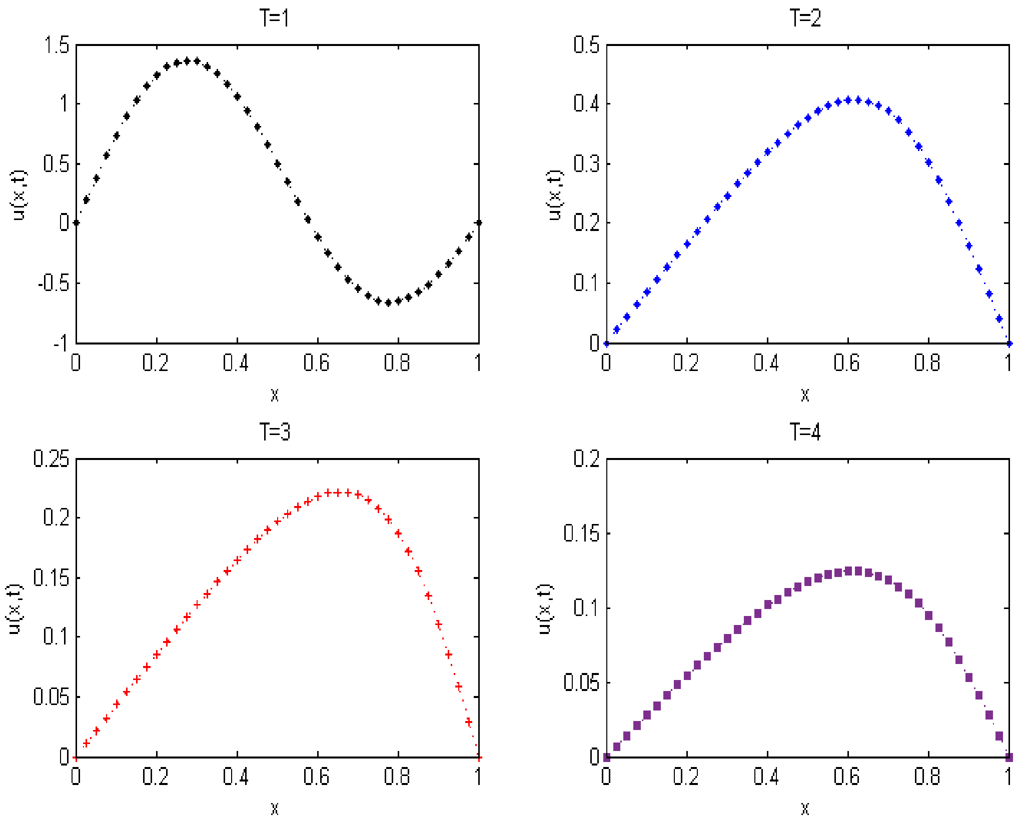


Figure 3.4 Physical behaviour of the 3^{rd} test problem for different values of time.

3.5 Conclusion

The present study demonstrates how effective the proposed hybrid approach of DQM with modified cubic B-splines and the finite difference scheme for discretization can be implemented to solve the partial differential equations. This work is an effort to develop numerical approach that can provide the approximate solution which is close to the exact solution to the Burger's equation. The present work is exemplified with the assistance of three examples that demonstrate the benefit of the method in terms of the accuracy obtained with less domain partition and for the adequate required time steps. The presented algorithm is capable of being modified that can be applied to higher order nonlinear differential equations in either linear or nonlinear form.

PART B:

"NONLINEAR DYNAMICS OF THE FISHERS EQUATION WITH NUMERICAL EXPERIMENTS."

3.6 Introduction

In this work. a hybrid scheme is developed to find more accurate solutions to the Fishers equation (FEs) which can be programmed and can be applied to solve similar equations. The Reaction-diffusion equations have been a keen topic of study for researchers in various fields of science and technology. One such reaction-diffusion equation was introduced by Fishers in 1937 [126-127] in a study of the development of favourable genes due to mutation which was known as Fishers equation. After framing, the application of this equation was investigated in many other related phenomena such as describing the pattern formation in wave propagation, studying the growth of cells in tissue engineering modelling oscillating chemical reactions, in population biology in the healing of a wound and in the growth of a tumour.

This nonlinear partial equation can be framed as:

$$u_t = vu_{xx} + \rho f(u), t \geq 0, x \in (-\infty, \infty) \quad (3.11)$$

where x represents the coordinates in space, t denotes the time, v is a constant that denotes the coefficient of diffusion, f depicts nonlinear reaction term and ρ is a term for reaction factor. These terms signify the parameter of the phenomenon that is modelled as Fishers equation. For example, in the study of brain tumours, it represents the carrying capacity, the diffusion coefficient shows the migration of cells and the reaction factor represents the growth rate. Similarly, in the study related to the gene propagation. it represents the frequency of the mutant gene and the reaction factor ρ signifies the strength of selection for the gene that shows the effect of the mutation. This equation is a well-known mathematical model that presents the asymptotic behaviour of the family of travelling waves. In many of the modelled scientific and biological phenomenon, the considered reaction term is taken as $\rho u(1-u)$ and the equation is solved with the following considered conditions at the boundary and at the initial phase of time:

I.C: $u(x, 0) = u_0(x) \in [0,1]$,

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

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

Based on the requirement of the modelled phenomenon and the parameters. The equation can be solved with the non-local conditions [I.C with B.C (a)] or the local conditions [I.C with B.C (b)] with a finite domain $[x_L, x_R]$

A number of efforts have been made by the researchers to find the solution to the equation using various approaches. For instance, FEs has been solved using the space derivative approach of the Fourier series. The explicit solution has been implemented by the authors to solve FEs in[128]. This equation has also been solved using the explicit finite difference approach [129], and the petrov-Galerkin finite element technique [130]. A modified finite-difference scheme has been constructed using dynamical consistency by mickens for solving the FEs [131]. The moving mesh method has been implemented to the FEs to discuss the numerical solutions in [132]. A comparative study has been done [133] for the finite difference scheme with a nodal integral method taking the FEs numerical solution. The exact solution of the FEs is obtained using the Adomian decomposition approach [134]. A spectral approach using Chebyshev-Lobatto points has been developed [135] to solve the equation. A differential quadrature approach based on polynomials has been implemented to Fisher's equation in one [136, 137] as well as in two dimension. Some other numeric approaches have also been implemented successfully for the study of Fisher's equation namely wavelet Galerkin method [138], tension spline-based method [139], collocation-based approaches [140–143] and boundary integral equation method [144].

The present work show cases a hybrid approach using the Crank-Nicolson scheme with discretization and application of DQM for approximating derivatives using B-spline basis functions.

The numerical scheme is discussed for implementation of the scheme on the FEs in the next section followed by the numerical example to present the obtained results by the proposed scheme. The work is concluded by the discussion of the present scheme with the obtained results.

3.7 Scheme implementation

Consider the FEs (3.11) defined in the introduction section. On discretizing the time derivative using finite difference approach with the implementation of the Crank-Nicolson technique on the space derivative and the other available functions, the equation can be written after taking terms of n^{th} and $(n+1)^{th}$ level separately as:

$$\frac{U_m^{n+1}-U_m^n}{\Delta t}=\nu\left(\frac{(U_{xx})_m^n+(U_{xx})_m^{n+1}}{2}\right)+\rho\frac{\{[U_m(1-U_m)]^n+[U_m(1-U_m)]^{n+1}\}}{2}$$

On linearizing the non-linear term using the approach given by Rubin and Graves [145] as follows:

$$(U^2)_m^{n+1}=2(U)_m^n(U)_m^{n+1-}(U_m^n)^2$$

results in the following equation:

$$U_m^{n+1}(a_1\rho\Delta t U_m^n)-a_2U_{xx}^{n+1}=a_3U^n+a_2U_{xx}^n$$

Where, $a_1=1-\frac{\rho\Delta t}{2}, a_2=\frac{v\Delta t}{2},, \; a_3=1+\frac{\rho\Delta t}{2}$

On replacing the derivatives in the equation using the DQM, the iterative scheme can now be written as follows:

$$U_m^{n+1}(a_1\rho\Delta t U_m^{n+1})-a_2\sum_{k=1}^{N}P_{mk}^{(2)}\,U_k^{n+1}=a_3U^n+a_2\sum_{k=1}^{N}P_{mk}^{(2)}\,U_k^n$$

Equation (3.4) can be written in a matrix form as:

$$\begin{bmatrix} A_{11}^n & A_{12}^n & \cdots & A_{1n}^n \\ A_{21}^n & A_{22}^n & & A_{2n}^n \\ \vdots & \vdots & \ddots & \vdots \\ A_{(N-1)1}^n & A_{(N-2)2}^n & \cdots & A_{(N-1)N}^n \\ A_{n1}^n & A_{n2}^n & \cdots & A_{nn}^n \end{bmatrix}\begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ \vdots \\ \vdots \\ U_N^{n+1} \end{bmatrix}=\begin{bmatrix} B_{11}^n & B_{12}^n & \cdots & B_{1n}^n \\ B_{21}^n & B_{22}^n & & B_{2n}^n \\ \vdots & \vdots & \ddots & \vdots \\ B_{(N-1)1}^n & B_{(N-2)2}^n & \vdots & B_{(N-1)N}^n \\ B_{n1}^n & B_{n2}^n & \cdots & B_{nn}^n \end{bmatrix}\begin{bmatrix} U_1^n \\ U_2^n \\ \vdots \\ \vdots \\ U_N^n \end{bmatrix}$$

Where

$$A_{ij}^n=\begin{cases} a_1+\rho\Delta t U_m^n-a_2P_{ij}^{(2)}, i=j \\ -a_2P_{ij}^{(2)}, i\neq j \end{cases} \text{ and } B_{ij}^n=\begin{cases} a_3+a_2P_{ij}^{(2)}, i=j \\ a_2P_{ij}^{(2)}, i\neq j \end{cases}$$

On implementation of the required boundary conditions to the system of equations, the system is solved for the solution iteratively applying the Gauss elimination method.

3.8 Results

The numerical solutions for the FEs are evaluated taking the different set of parameters in form of a numerical example with large reaction factor by the proposed hybrid method. By evaluating the errors, the solutions are given in tables and presented in figures.

The FEs was presented by Petzold and Ren [123] in study related to the stability of a moving mesh approach taking a large value of non-linear reactive term (ρ) as compared to the diffusion term with the reaction rate coefficient (v) greater than equal to one. The analytic solution for the above said equation is reported in literature [128] as:

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

Solution of the FEs is obtained using the proposed hybrid scheme considering a finite domain as $[x_L, x_R] = [-0.2, 0.8]$ with nonlocal boundary conditions. The solution is obtained for the time $t = 0.001$ to $t = 0.003$ and reaction factor $\rho = 2000, 5000$ and $10{,}000$. The obtained numerical solutions have been presented in Figure 1 and Figure 2 for different reaction factors to show the solutions at different time levels with the number of partitions as 40 and 80 respectively. The solution is also presented in the 3D form in Figure 3 and Figure 4 to showcase the change in behaviour with respect to time. The results are also discussed for the accuracy in terms of the L_∞ errors in Table 1 and Table 2 for reaction factor $\rho = 2000$ and $\rho = 5000$ for different values of knot partitions. Comparison of the solutions obtained by the present scheme is given in Table 3 and Table 4 for reaction factor $\rho = 10{,}000$ which are available in the literature with the exact solutions at different knot points. The work is compared with the results obtained by the standard collocation approach [140] and another form of collocation approach [141]. The calculated results are almost similar but the difference is regarding the number of domain partitions considered. This is a major factor responsible for the computational time which is less in the present scheme as compared to the collocation approaches.

Table 3.5 L_∞ errors obtain by the hybrid scheme at different time levels for $\rho = 2000$ for Δt=1$\times 10^{-5}$.

$\boldsymbol{N}$	$\boldsymbol{t = 0.0001}$	$\boldsymbol{t = 0.0015}$	$\boldsymbol{t = 0.0020}$	$\boldsymbol{t = 0.0025}$
11	1.2021E-04	8.8798E-04	2.4096E-03	5.7924E-03
21	6.1551E-06	3.0570E-06	6.5669E-06	6.28168-06
31	1.5327E-05	3.9296E-06	8.5750E-07	3.5690E-07

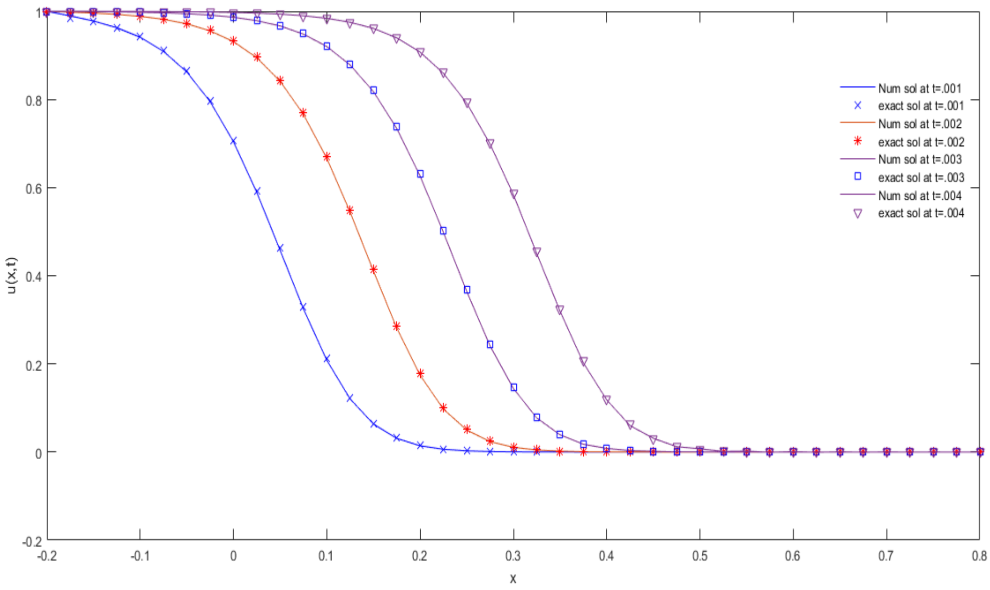


Figure 3.5 Solution profile of the travelling wave for FEs at $\rho = 2000$

Table 3.6 L_∞ errors obtain by the hybrid scheme at different time levels for $\rho = 5000$ for Δt=1$\times 10^{-5}$.

$\boldsymbol{N}$	$\boldsymbol{t = 0.0001}$	$\boldsymbol{t = 0.0015}$	$\boldsymbol{t = 0.0020}$	$\boldsymbol{t = 0.0025}$
31	7.1895E-06	1.9510E-04	6.9982E-03	1.9982E-02
41	7.1122E-07	1.8788E-06	3.86138-05	9.6423E-04
61	1.4733E-07	2.4601 E-07	1.2951 E-07	1.1264E-06
81	1.1419E-07	2.7305E-07	3.8991 E-07	5.2300E-07

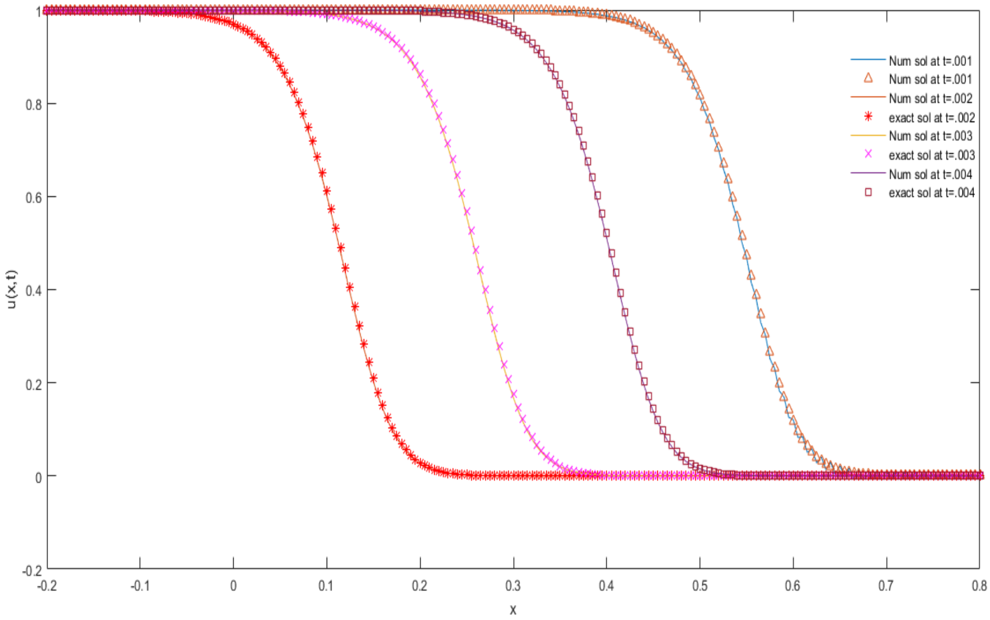


Figure 3.6 Solution profile of the travelling wave for FEs at $\rho = 5000$

Table 3.7 Solutions obtained by the hybrid scheme in comparison with the exact solutions and with others given in the literature at $t = 0.001$ at $\rho = 10{,}000$.

X	**[140]**	**[141]**	**Present**	**Exact**
-0.2	1.00000	1.00000	1.00000	1.00000
-0.1	0.99999	0.99999	1.00000	1.00000
0.1	0.97203	0.97199	0.9668	0.9691
0.2	0.28644	0.29002	0.2398	0.2578
0.3	0.00032	0.00035	2.37E-04	2.73E-04
0.4	0.00000	0.00000	8.82E-08	7.50E-08
0.5	0.00000	0.00000	2.22E-09	1.99E-11
0.6	0.00000	0.00000	1.24E-10	5.29E-15
0.7	0.00000	0.00000	4.63E-12	1.40E-18

Table 3.8 Solutions obtained by the hybrid scheme in comparison with the exact solutions and with others given in the literature at $t = 0.002$ at $\rho = 10{,}000$.

X	**[140]**	**[141]**	**Present**	**Exact**

-0.2	1.00000	1.00000	1.00000	1.00000
-0.1	1.00000	1.00000	1.00000	1.00000
0.1	0.99999	0.99999	0.99992	0.99992
0.2	0.99958	0.99959	0.99950	0.99953
0.3	0.99454	0.97551	0.97026	0.97200
0.4	0.30845	0.32607	0.26831	0.28375
0.5	0.00036	0.00045	0.00063	0.00033
0.6	1.03E-07	0.00000	1.02E-07	9.16E-08
0.7	2.93E-11	0.00000	2.67E-10	2.43E-11

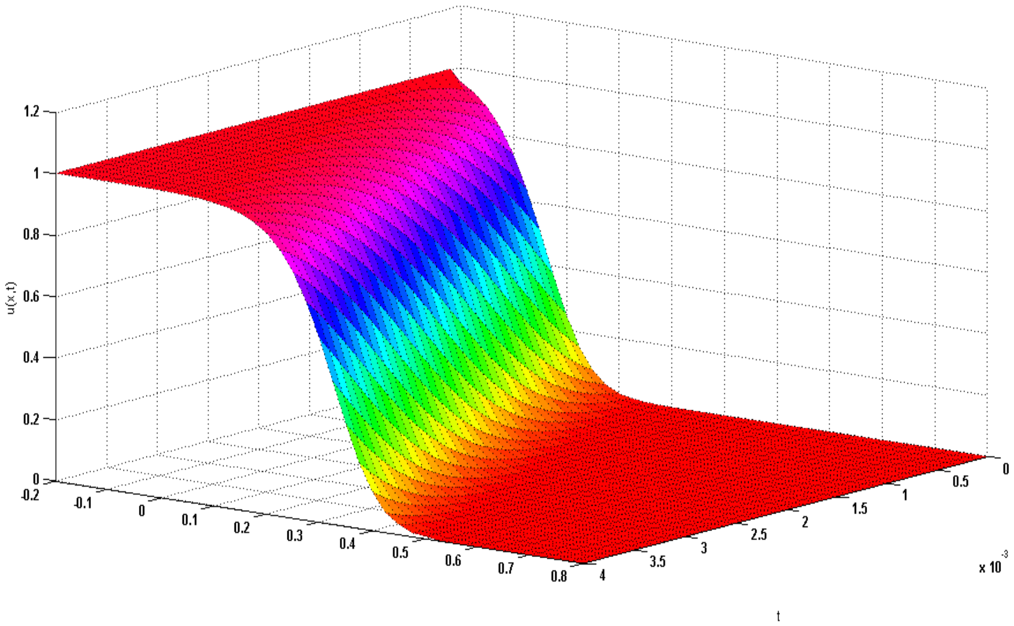


Figure 3.7 Solution profile of the travelling wave for FEs at $\rho = 2000$

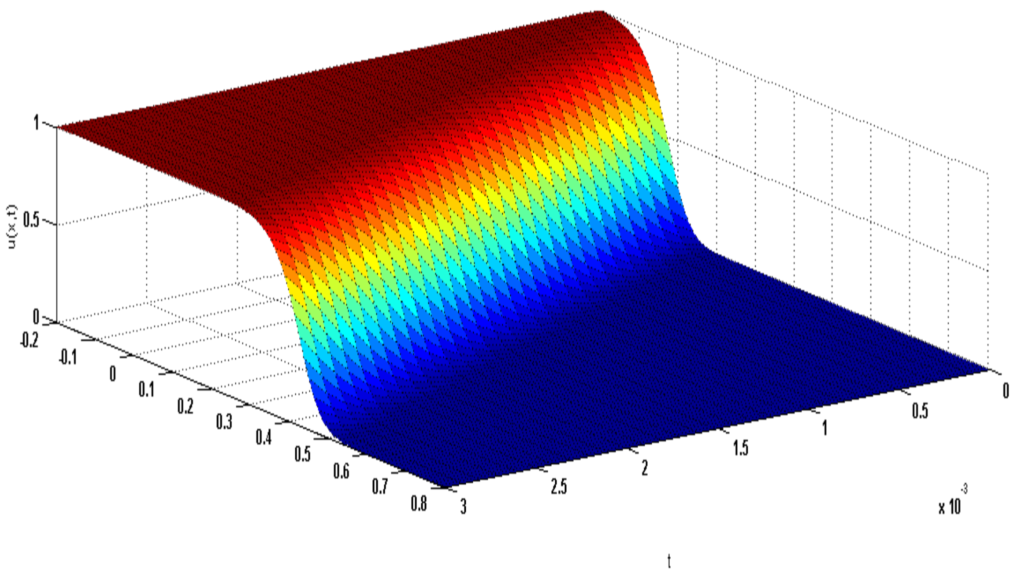


Figure 3.8 Solution profile of the travelling wave for FEs at $\rho = 5000$

3.9 Conclusion

The present work is showcasing the efficiency of the proposed hybrid approach of Crank-Nicolson based DQM with reframed cubic B-spline method for the study of Fishers equation. This is an effort in the development of the numerical approaches to find the improved solution of the FEs. The present work is illustrated with help of an example with large reaction factor. The obtained results show the benefit of the method in terms of the accuracy obtained with less domain partition. The presented algorithm can be extended for the application to the higher order nonlinear partial differential equations.

CHAPTER 4

FISHER 1D AND 2D EQUATION USING PARTICLE SWARM OPTIMIZATION (PSO) AND DIFFERENTIAL QUADRATURE METHOD (DQM)

4.1 Introduction:

Fisher [126] and Kolmogorov et al. [164] presented the Fisher-Kolmogorov-Petrovsky-Piscunov equation, also known as the Fisher KPP equation, in 1937, which is now generally known as Fishers equation. Fishers equation is a time-dependent PDE resembling a reaction diffusion equation with periodic solutions and pattern formation behaviours[154], [165, 166]. In many physical and biological applications, the above equation designs the traveling wavefront with bounded solutions and it also appears in chemical kinetics, in various logistic growth models and study of wave propagation. Fishers equation is notable in physical phenomena, and there is a special approach to the reaction-diffusion equation in 1-D as follows:

$$u_t = \lambda u_{xx} + u\,(1-u), -\infty \,<\, x \,<\, \infty, t \,>\, 0 \tag{4.1}$$

where β is known as a real parameter, reaction term is given by $u\,(1-u), u > 0$ and it can depend upon space variable, λu_{xx} is known as diffusion term and coefficient is known as a non-negative constant. Equation (4.1) is well-known 1-D Fishers equation.

As in the phenomenon of cell migration, cell proliferation is one of the necessary mechanisms for wound healing and tumour progression. During this process, cells detect and answerable to variety of chemical and physical intimates. Numerous studies have examined the biochemical intimates of this process in depth, but physical intimates are still being considered to evaluate the phenomenon of tumour progression and wound healing. In both phenomena, the role of geometry in the proliferation rate of cell populations is extensively studied [130]. A model used to characterise the proliferation of cells is expressed as the well-known 2-D Fishers Kolmogorov equation:

$$u_t = \nabla(\lambda \nabla u) + \mu f(u) \tag{4.2}$$

Here $u\ (x, y, t)$ is dimensional cell, λ is cell diffusivity, μ is cell proliferation rate and ∇ two- dimensional gradient operator and the function $f(u) = u\ (1 - u)$ is a term in the equation containing non-linearity which the effect of reaction is represented by it.

Assuming λ_1, λ_2 as the principal values of λ along the principal axes having coordinates (X, Y) [7], after that equation can be written in the form:

$$u_t = \lambda_1 u_{xx} + \lambda_2 u_{yy} + \mu f(u) \quad (4.3)$$

the value of x, y and t on rescaling can obtained as:

$$t = \mu T\ , x = \sqrt{\frac{\mu}{\lambda_1}} X \text{ and } y = \sqrt{\frac{\mu}{\lambda_2}} Y$$

Above equation becomes,

$$u_t = \lambda_1 u_{xx} + \lambda_2 u_{yy} + \mu f(u) \quad (4.4)$$

It is the general form of Fishers equation in two-dimension.

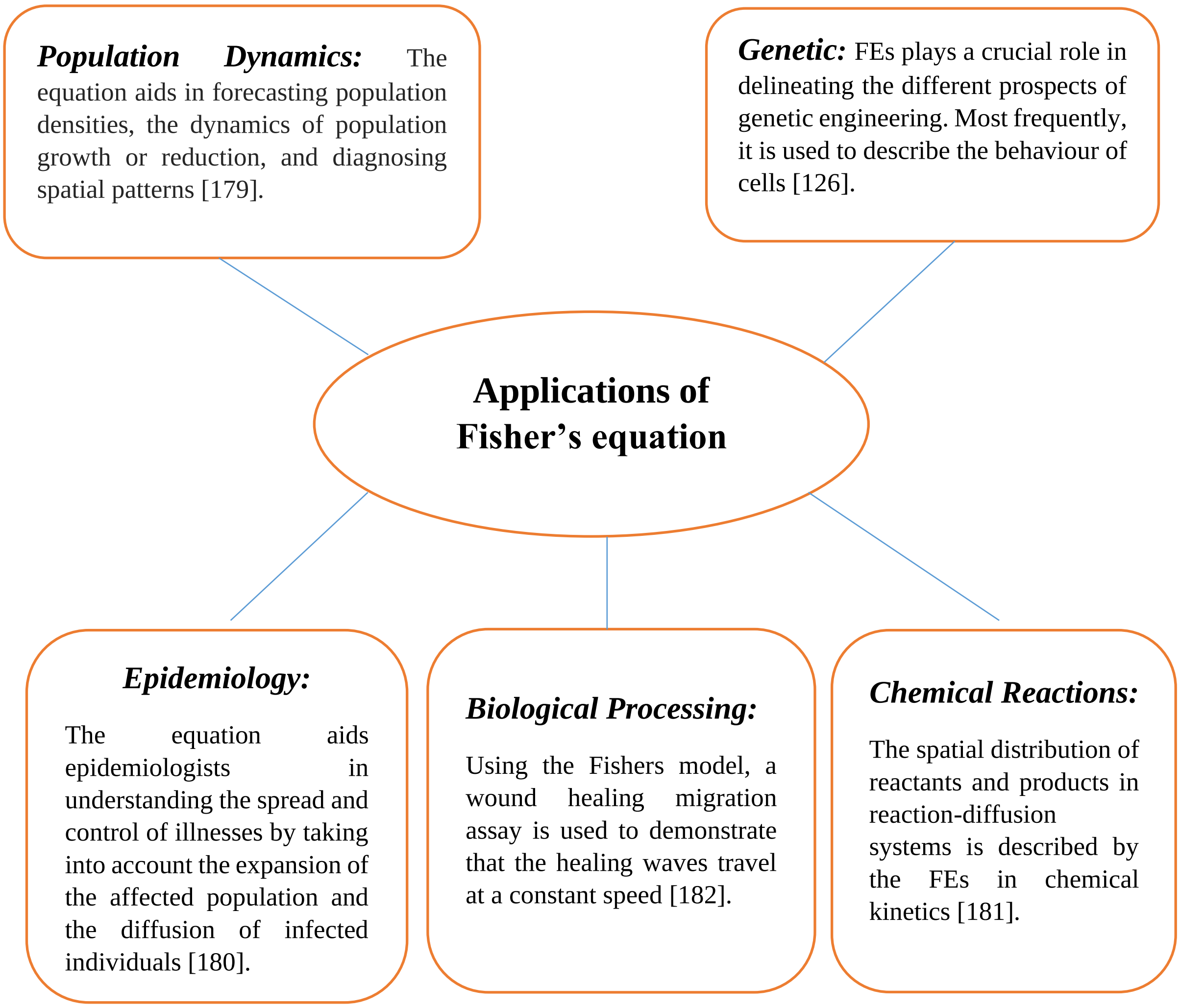
In the previous decades efforts at a large scale has been made to solve 1-D fishers equation like Galerkin method by B-spline and Crank Nicolson method , cubic spline method [115], combination of collocation and Crank- Nicolson technique [139], [143], finite difference method [131], cubic B-spline quasi interpolation method [167], mesh method [152], sinc collocation method [155] , finite difference method [168], pseudo-spectral method [135], time-asymptotic convergence [169], one-dimensional and two-dimensional by quadrature method [170], finite difference [131], B-spline method [140], FRDE [141], DQM method [171], B-spline Galerkin method[143], exponential cubic B-spline method [172], cubic B-spline quasi-interpolation [173], wavelet Galerkin approach [174], Adomian decomposition method [134].

In last few years, the researchers focussed on the existence, uniqueness and stability of the two-dimensional Fishers equation. First numerical solution was analysed in 1974 by pseudo spectral approach [135] and then it has been solved by other methods such as petrov Galerkin finite element method [130], alternating group explicit iterative method [175], HAM [176], tanh method [177], travelling wave transform [178], implicit and explicit finite difference algorithm [154], DQM method [136], and cubic B-spline method [140].

In the chapter DQM, PSO technique and exponential basis functions are used to discuss one-dimensional and two-dimensional FEs. To determine the parameter value which involved in the exponential B-spline method which is significant for the solution that has been resolved by optimisation technique.

4.2 Application of Fisher's Equation:

Fisher's reaction-diffusion equation, is a partial differential nonlinear equation that models how a population or quantity spreads across time and space. It can be used in a number of different fields. Here are a few illustrations.



4.2.1 Population Dynamics: To describe the dispersal of species or populations through space, the Fishers equation is frequently employed in population biology and ecology. It takes into account how population growth and diffusion interact, taking into account things like birth, death, and dispersal rates. The equation aids in forecasting population densities as well as the dynamics of population growth or reduction and in diagnosing spatial patterns [179]

4.2.2 Epidemiology*:* The Fishers equation is used to simulate the spatial spread of infectious illnesses in epidemiology. The equation aids epidemiologists in understanding the spread and control of illnesses by taking into account the expansion of the affected population and the diffusion of infected individuals [180]. It facilitates in comprehending how various aspects of disease spread, such as transmission rates, movement patterns, and control methods, are impacted.

4.2.3 **Chemical Reactions:** The spatial distribution of reactants and products in reaction-diffusion systems is described by the FEs in chemical kinetics [181]. Scientists can research reaction fronts, the creation of chemical patterns, and the general behavior of reaction-diffusion systems thanks to its assistance in modeling chemical reactions that display both diffusion and reaction processes.

4.2.4 **Genetic Engineering:** Genetic Engineering is a new field of study in which various techniques have been used to alter the genetic makeup of cells in order to enhance an organism in some manner. By altering the genome, scientists are able to confer desirable characteristics on different organisms. Animal genetic engineering has increased dramatically in recent years. Animal reproduction and the treatment of genetic disorders are the most promising potential applications for genetic modelling. FEs plays a crucial role in delineating the different prospects of genetic engineering. Most frequently, it is used to describe the behavior of cells [126] . It is one of the simplest reaction-diffusion equations and was initially used to examine the dissemination of a favorable gene through a population. FEs is also a useful model for describing the expansion of a monolayer cell province in vitro during tissue engineering. It depicts the behavior of a cell population as a combination of irregular cell movement and logistic multiplication, i.e. growth to the maximum cell thickness. It also analyses that, under certain conditions, the cell movement front will assume the form of a travelling wave with a constant shape and speed.

4.2.5 **Biological Processing:** The fundamental FEs in which cell division is influenced by chemical signals and cell movement is influenced by mechanochemical factors. In a study, using the Fisher model, a very straightforward wound healing migration assay was used to demonstrate that the healing waves travel at a constant speed, as predicted by FEs. In addition, it has been demonstrated that the manner in which the wave speed approaches the asymptotic wave speed closely matches that predicted by the model. According to our knowledge, this is the first medical evidence of FEs [182].

4.3 Differential Quadrature Method:

DQM is an approach used to find the efficient solution of the partial differential equations by converting them to the ordinary differential equations by representing the derivatives as the linear sum of the function with the weighting coefficients. There are a lot of basis functions that can be implemented to approximate the derivative such as Lagrange's polynomial, simple polynomial and the radial basis function etc.

Let the finite domain of the differential equation is given as [a, b]. It can be discretized in to number of known points as $a = x_0 < x_1 < \cdots\cdots < x_{n-1} < x_n = b$

The r^{th}derivative of the function can be written as:

$$\left\lfloor \frac{d^r U}{dx^r} \right\rfloor_{x_i} = \sum_{j=1}^{N} p_{ij}^{(r)} U(x_j), i = 1 \; to \; N, r = 1 \; to \; N-1 \tag{4.5}$$

where, $p_{ij}^{(r)}$ represents the coefficients to be calculated using the exponential B-Spline basis function.

As per the literature, the polynomial B-splines with a free parameter are the generalization of the exponential B-spline functions. The third-degree exponential B-spline can be defined as follows:

$$B_i(x) = \begin{cases} b_2\left[(x_{i-2} - x) - \frac{1}{p}\left(\sinh\left(p(x_{i-2} - x)\right)\right)\right] & \text{if } x \in [x_{i-2}, x_{i-1}]; \\ a_1 + b_1(x_i - x) + c_1 e^{p(x_i - x)} + d_1 e^{-p(x_i - x)} & \text{if } x \in [x_{i-1}, x_i]; \\ a_1 + b_1(x - x_i) + c_1 e^{p(x - x_i)} + d_1 e^{-p(x - x_i)} & \text{if } x \in [x_i, x_{i+1}]; \\ b_2\left[(x - x_{i+2}) - \frac{1}{p}\left(\sinh\left(p(x - x_{i+2})\right)\right)\right] & \text{if } x \in [x_{i+1}, x_{i+2}]; \\ 0 & \text{otherwise.} \end{cases}$$

here

$$p = \max_{0 \le i \le N} p_i, s = \sinh(ph), c = \cosh(ph)$$

$$b_2 = \frac{p}{2(phc - s)}, a_1 = \frac{phc}{phc - s}, b_1 = \frac{p}{2}\left[\frac{c(c-1) + s^2}{(phc - s)(1 - c)}\right],$$

$$c_1 = \frac{1}{4}\left[\frac{e^{-ph}(1 - c) + s(e^{-ph} - 1)}{(phc - s)(1 - c)}\right], d_1 = \frac{1}{4}\left[\frac{e^{ph}(c - 1) + s(e^{ph} - 1)}{(phc - s)(1 - c)}\right]$$

Each basis function $B_i(x)$ is twice continuously differentiable. The values of $B_i(x), B_i'(x)$ and $B_i''(x)$ at the knots x_i 's are obtained from the Table 1.

Table 4.1. Exponential B-spline values:

	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$B_i(x)$	0	$\frac{s-ph}{2(ph-s)}$	1	$\frac{s-ph}{2(phc-s)}$	0
$B_i'(x)$	0	$\frac{p(1-c)}{2(ph-s)}$	0	$\frac{p(c-1)}{2(phc-s)}$	0
$B_i''(x)$	0	$\frac{p^2 s}{2(phc-s)}$	$\frac{-p^2 s}{phc-s}$	$\frac{p^2 s}{2(phc-s)}$	0

For functions which defined on the interval $[a,b]$, the $B_i(x), i=-1,\ldots,N+1$ form a basis.

In order to calculate the value of parameter 'p' for which the error will be minimise and in order to minimise the error an optimisation technique is required. The process of minimizing or maximizing an objective function by choosing the best values for each of its variables from within the permitted range of values is referred to as "optimization."

4.4 Applications of optimization techniques

Optimization techniques have a variety of applications. Some of the applications of optimization techniques include:

4.4.1 Operations research*:* Optimization techniques are widely used in operations research to solve problems such as resource allocation, inventory management, and scheduling. For example, linear programming can be used to optimize production and transportation schedules to minimize costs in a manufacturing company.

4.4.2 Engineering: Optimization techniques are used in engineering to optimize the designing of complex problems, such as airplanes, bridges, and electronic circuits. For example, genetic algorithms can be used to optimize the shape of a wing for a given set of designs and nonlinear programming (NLP) can be used to optimize the shape of an aircraft wing to minimize drag while maximizing lift.

4.4.3 Finance: Optimization techniques are used in finance to optimize investment portfolios, risk management, and trading strategies. For example, quadratic programming can be used to minimize the risk of an investment portfolio while maximizing its return.

4.4.4 Machine learning: Gradient descent is an optimization technique that is widely used in machine learning to optimize the parameters of a model. It is used to minimize the loss function of a model, which measures the difference between the predicted and actual output. Gradient descent is used in many such as linear regression, machine learning algorithms, neural networks and logistic regression.

4.4.5 Energy systems: Optimization techniques are used in energy systems to optimize the operation and control of power grids, renewable energy systems, and energy storage systems. An example is optimal power flow, which is an optimization problem that aims to minimize the operating cost of a power system while satisfying various constraints, such as the demand for electricity and the capacity of transmission lines.

Overall, optimization techniques have numerous applications across different fields, and they are an essential tool for solving complex problems and improving efficiency and performance. In this chapter, Particle swarm optimisation tool which is an optimization technique is used to optimize the value of errors.

4.6 Particle Swarm Optimization (PSO)

In the class of algorithms inspired by nature, PSO is one of the potential global optimisation methods. PSO is a computer technique used in computational science to optimize problems by repeatedly attempting to raise the quality of possible answers.

A computational worldwide optimization technique called PSO was initially suggested by Dr. Kennedy and Eberhart in 1995 [69]. A particle without quality or volume assumes the role of each individual and a simple behavioural pattern is regulated for each particle to demonstrate the complexity of the complete particle swarm. One may use this strategy to resolve the difficult optimist problems. Swarms are considered as population particles that transmit information to improve the effectiveness of the search solution and find the global optimum in this nature-based swarm optimization technique. Each particle has their best experience which is called their current position. There is a best experience of all the particles that is called their global best position. A particle changes the current best position for particle when it detects a target superior to all previously detected locations. During the iterations, there is a new best solution for each of the n particles. The main purpose of this algorithm is to find the overall best solution from all available solutions. This process goes up to the defined iterations or until the objective is not reached. In this optimization algorithm, the velocity U_i^{k+1} and the position X_i^{k+1} of the i^{th}particle are updated as:

$$V_i^{k+1} = V_i^k + a_1 * random * \left(P_{best_i} - X_i^k\right) + a_2 * random * \left(G_{best} - X_i^k\right),$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}$$

Where V_i^k - is velocity of particle i at k^{th} iteration,

a_1, a_2 – are real parameters,

$random-$ is a random number, whose value lies between 0 & 1,

X_i^k = is i^{th}- particle position at k^{th}-iteration,

P_{best_i} – Personal best position of particle i,

G_{best} – Global best position of whole search space.

PSO is a computational technique that iteratively optimizes, a problem to reduce error. It is a statistical method used to determine parameter values. To find the optimal answer to an optimization problem, the particles communicate, share their knowledge and follow a simple rule. The PSO method is an innovative method for evaluating the best shape parameter value of RBF using the non-linear partial differential equation. It is a global search optimization strategy and offers numerous characteristics in the parameter space.

4.6.1 Algorithm of PSO technique

The collective activities of birds while searching for food served as the inspiration for the development of the PSO algorithm [161]. In this technique, particles are considered entities, and their location affects how they behave. There is a component of the solution that has to be optimized at each location. The search process is driven by the updating of particle positions and velocities at each time step. There is a location for each particle in the swarm that can be resolved in one-dimensional space. Each particle moves according to its best-known locations both locally and across the search space, which are updated when new locations are discovered by other particles. With the use of a simple mathematical formula, the updating guidelines for each particle's location and speed are provided by

$$v_{id}^{t+1} = \chi[v_{id}^t + c_1 r_1(p_{id}^t - x_{id}^t) + c_2 r_2(p_{gd}^t - x_{id}^t)]$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$$

Where, x_{id}^{t}, represents c particle's position and v_{id}^{t} represents i^{th} particle's velocity in d dimension at time step t, p_{gd} represents the particle having the best fitness value, p_{id} is the particle's best position visited so far, c_1, c_2 are acceleration coefficients which quantifies particle personal and global experience respectively, χ is called constriction coefficient which evaluates a value in the range [0,1] and is given by

$$\chi = \frac{2\kappa}{\left|2 - \phi - \sqrt{\phi(\phi - 4)}\right|}$$

With $\phi = \phi_1 + \phi_2, \phi_1 = c_1 r_1, \phi_2 = c_2 r_2$ and $\kappa \approx 1$.

4.6.2 PSO Algorithm

Input fitness function, lb, ub, Np, T, a_1, a_2

Initialize random population (P) and velocity U_i of particle i

Evaluate objective function value (f)

Assign P_{best_i} as P and f_{best} as f

Identify the solution with best fitness & assign the solution as G_{best} and fitness as f_{best}

For $t = 1$ to T

For $i = 1$ to N_p

Determine the velocity U_i

Determine the new position (X_i)

Bound X_i

Find objective function value

Update the population by including X_i & f_i

Update P_{best} and f_{pbest}

Update G_{best} and f_{gbest}

End

End

The result of using the strategy is a solution that is accurate since it finding out for the number of repetitions with the no. of iterations across a set population size. Using radial basis functions, this strategy has been effectively used to calculate equation solutions [62]. The parameters taken here are as: size of swarm:20; no. of maximum iterations: 50; inertia weight is decreased linearly with size $c_1 = c_2 = 2.05$.

4.6.3 Applications of PSO technique

A number of papers have been published on applications of PSO. One of the prospective global optimisation method is PSO that has been extensively applied to solve issues in numerous fields such as health care, environmental, industrial, commercial, smart city [63], [64], geotechnical engineering [46], civil engineering [65], electromagnetics[66], and wireless networks [67] etc.

4.6.4 Advantages of PSO technique

Advantages of the PSO are as follows [68]:

- The PSO is built on intelligence. Both engineering and scientific research can utilized it.
- PSO do not calculate mutations or overlap. The search can be carried out using the particle's speed. Over the course of several generations, only the most optimistic particle may transfer information to the other particles, and knowledge evolves swiftly.
- The PSO computation is extremely simple. In comparison to other development calculations, it takes up greater optimization space and is easy to execute.
- PSO makes a decision immediately based on the response and a real-number code. The constant of the solution and the dimension have the same number.

4.6.5 Disadvantage of PSO technique

This method cannot be used to solve non-coordinate system problems like the rules for how particles move in an energy field since it lacks dimensionality [68].

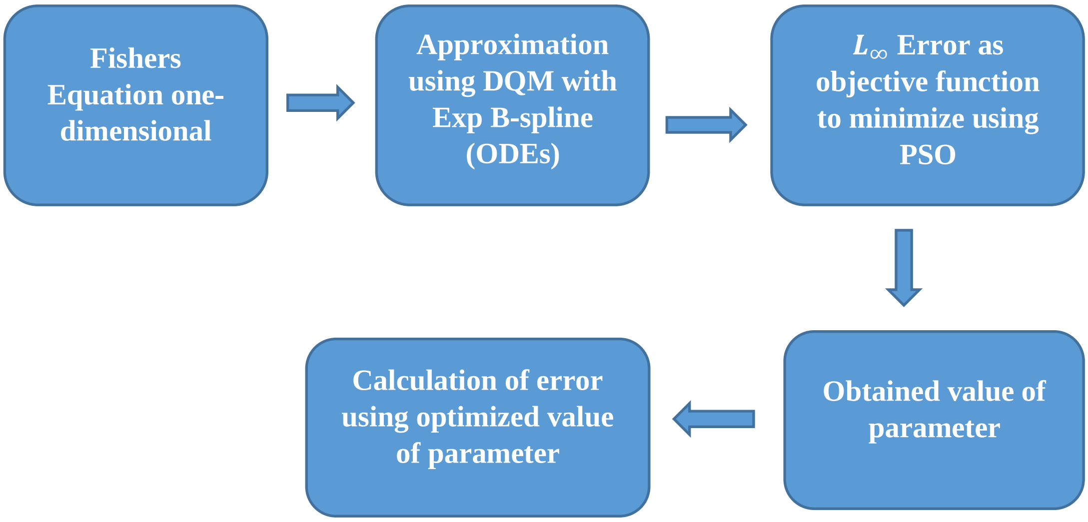
4.6.6 Implementation:

To solve the fisher equation

$$u_t = \nu u_{xx} + \rho f(u)$$

Substituting the approximations of the space derivatives using the DQM [100] with exponential B-spline basis functions results in an ODEs that can be solved by any appropriate numerical method. Once the solutions have been identified for the problem with the known initial condition, the PSO technique is used to minimise errors and reduce the obtained error by comparing exact and numerical solutions. Once the value of the parameter is obtained for the minimum error, numerical results can be calculated on predefined domain and time intervals.

The numerical scheme can be summarized and visualised as follows:



4.7 Numerical Result:

In this section, the numerical solution of the FEs is considered taking the different set of parameters in form of two numerical examples by the developed hybrid technique. The accuracy and the efficiency of the methods are compared by evaluating the error norms.

4.7.1 1st **test problem:** The FEs (4.1) was implemented and presented by Petzold and Ren [123] in the study of stability of a moving mesh system taking a large value of non-linear reactive term as compared to the diffusion term with the reaction rate coefficient greater than equal to one. The analytical solution for the above said parametric equation was obtained by Ablowitz and Zepetella [128] as follows:

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

Solution of the FEs is obtained considering a finite domain as $[x_L, x_R] = [-0.2, 0.8]$ with nonlocal boundary conditions. The obtained numerical solutions have been presented in form of figures and tables with the comparison of solutions obtained by differential quadrature and the collocation approach.

For different values of N and for different values of p which varies between a range $[1, 200]$.

Table 4.2: L_∞ errors obtained by the hybrid scheme at different time levels for $\rho = 2000$ for $t = 1 \times 10^{-5}$.

`		$t = 0.0010$			$t = 0.0015$	
	L_∞	**P**	L_∞ [160]	L_∞	**P**	L_∞ [160]
11	3.7880E-06	1	1.2021E-04	1.9631E-06	1	8.8798E-04
21	6.2634E-06	184.3448	6.1551E-06	1.8821E-06	64.9465	3.0570E-06
31	1.6386E-05	200	1.5327E-05	3.9521E-06	200	3.9296E-06
		t=0.0020			$t = 0.0025$	
11	3.2457E-06	12.1442	2.4096E-03	3.0905E-06	16.2559	5.7924E-03
21	5.9916E-07	27.4870	6.5669E-06	3.6021E-07	47.3644	6.2816E-06
31	8.9303E-07	190.5990	8.5750E-07	4.4764E-07	127.5467	3.5690E-07

The exact and numerical solution at different intervals shown in Figure 4.1. To compare the findings with the exact answer, the results are displayed graphically. The specific approach to the problem is found to be in good accord with the obtained results to assess whether the numerical solution is accurate for $\rho = 2000 \; and \; \Delta t = 1 \times 10^{-6}$, results are depicted by L_∞ errors and are presented in Table 4.2 and comparison is done with the results exists in the literature.

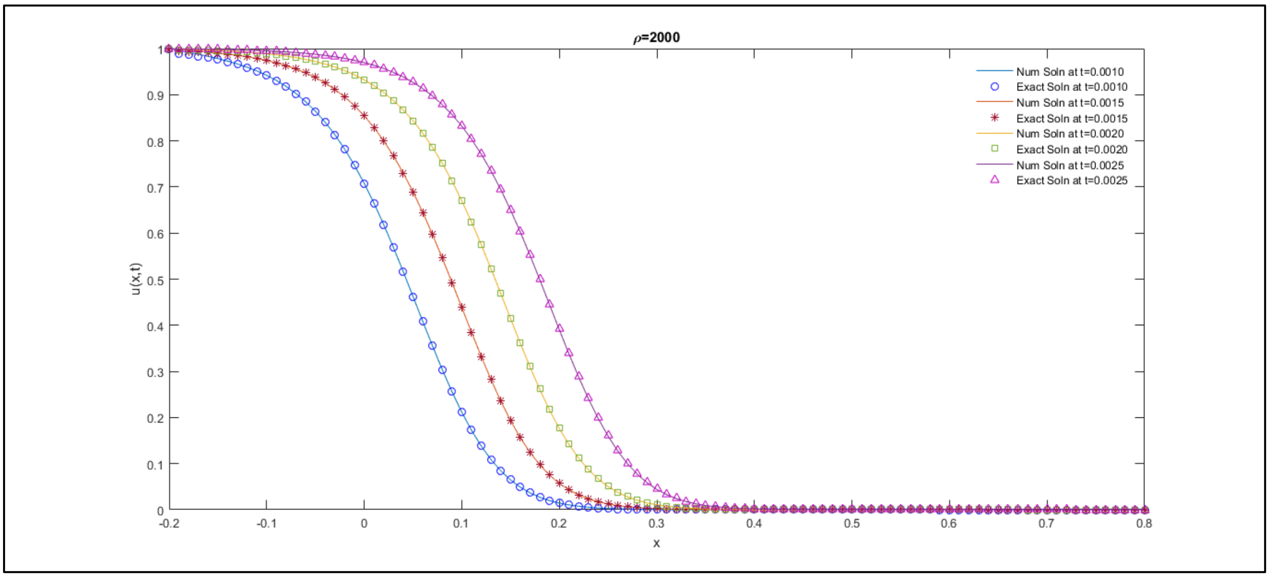


Figure 4.1 A graphical depiction of time dependent profiles versus *x* of 1^{st} test problem for $\rho = 2000$ for $t = 1 \times 10^{-5}$.

Table 4.3: L_∞ errors obtained by the hybrid scheme at different time levels for $\rho = 5{,}000$ for $t = 1 \times 10^{-5}$.

N	$\boldsymbol{t = 0.0010}$			$\boldsymbol{t = 0.0015}$		
	$\boldsymbol{L_\infty}$	**p**	$\boldsymbol{L_\infty}$ [160]	$\boldsymbol{L_\infty}$	**p**	$\boldsymbol{L_\infty}$ [160]
31	1.8794E-06	127.4740	7.1895E-06	8.2982E-07	101.02222	1.9510E-04
41	5.5736E-06	200	7.1122E-07	1.8174E-06	200	1.8788E-06
61	1.7687E-06	200	1.4733E-07	6.2910E-05	200	2.4601E-07
81	2.2558E-05	200	1.1419E-07	9.4953E-05	200	2.7305E-07
		t=0.0020			$\boldsymbol{t = 0.0025}$	
31	5.2087E-07	94.817065	6.9982E-03	3.1489E-07	92.4473	1.9982E-02
41	6.3480E-07	182.46816	3.8613E-05	4.9421E-07	171.7256	9.6423E-04
61	1.5366E-04	200	1.2951E-07	3.0157E-04	200	1.1264E-06
81	2.5400E-04	200	3.8991E-07	5.1449E-04	200	5.2300E-07

The exact and numerical solution at different intervals shown in Figure 4.2. To compare the findings with the exact answer, the results are displayed graphically. The specific approach to the problem is found to be in good accord with the obtained results to assess whether the numerical solution is accurate for $\rho = 5000\ and\ \Delta t = 1 \times 10^{-6}$, results are depicted by L_∞ errors and are presented in Table 4.3 and comparison is done with the results exists in the literature.

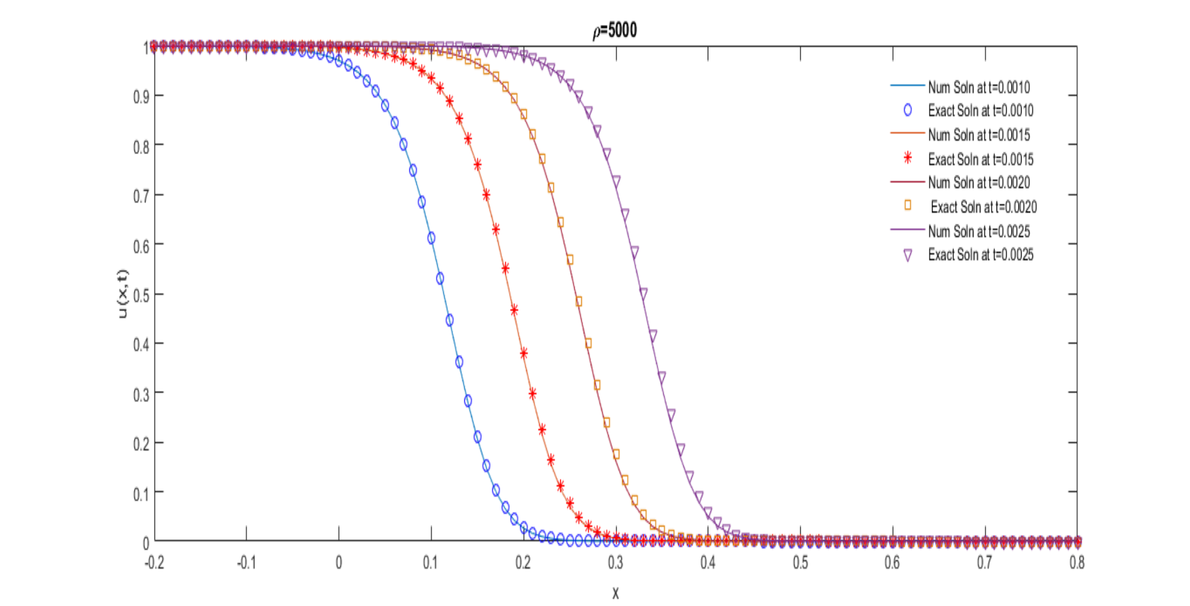


Figure 4.2 A graphical depiction of time dependent profiles versus *x* of 1^{st} test problem for $\rho = 5000$ for $t = 1 \times 10^{-5}$.

Table 4.4: L_∞ errors obtained by the hybrid scheme at different time levels for $\rho = 10{,}000$ for $t = 1 \times 10^{-5}$.

N	$t = 0.0010$		$t = 0.0015$	
	L_∞	p	L_∞	P
11	0	26.7534	0	28.1728
21	3.2683E-05	59.8813	4.5659E-05	67.7689
31	2.3438E-06	104.4888	3.9461E-06	105.1255
41	1.6684E-06	191.9409	9.7326E-07	179.5408
61	6.1756E-04	200	2.0E-03	200
81	9.7124E-04	200	3.5E-03	200

	t=0.0020		$t = 0.0025$	
11	0	32.908	0	90.4507
21	4.0265E-05	72.1092	3.0399E-05	74.4567
31	4.6042E-06	106.3847	4.7671E-06	107.3924
41	8.3694E-07	175	9.0643E-07	172.7307
61	4.3116E-03	200	7.7256E-03	200
81	8.1045E-03	200	1.4758E-02	200

Similarly, Figure 4.3 shows the exact and numerical solution at different time levels. To compare the numerical solution with the exact solution for $\rho = 10{,}000$ *and* $\Delta t = 1 \times 10^{-6}$, results are depicted by L_∞ errors and are presented in Table 4.4.

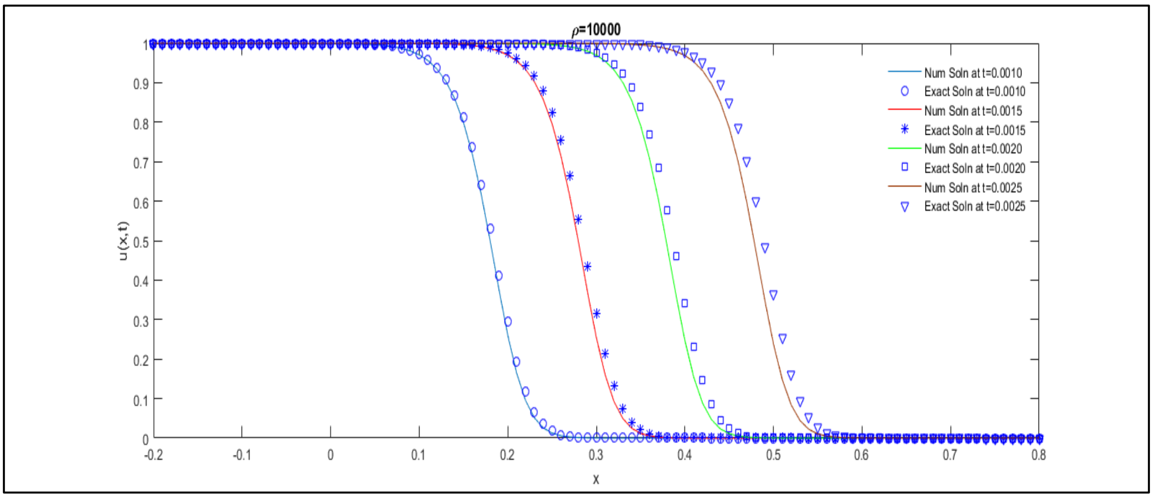


Figure 4.3 A graphical depiction of time dependent profiles versus *x* of 1^{st} test problem for $\rho = 10{,}000$ for $t = 1 \times 10^{-5}$.

4.7.2 2^{nd} **test problem*:*** Now, Solution is obtained by considering the equation (4.1) on a finite domain as $[x_L, x_R]$ = [0,1] with nonlocal boundary conditions. With initial condition has been considered as $u(x,0) = \frac{1}{(1+exp(\rho x))^2}$

With boundary condition $\lim_{x\to-\infty} u(x,t) = 1$, $\lim_{x\to\infty} u(x,t) = 0$ and exact solution is given by:

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

Table 4.5: L_2 and L_∞ errors obtained by the hybrid scheme for the different time levels solutions of 2nd test problem for $\rho = 6$ with $h = 1 \times 10^{-3}$ and $\Delta t = 1 \times 10^{-6}$ at different levels of time.

Time(t)	P	L_2	L_∞	TBC L_2 [162]	TBC L_∞ [162]
0.0001	10	1.5378E-07	1.4354E-13	7.0594E-05	1.2502E-04
0.0002	10	7.7521E-07	4.5552E-12	1.5179E-04	2.5006E-04
0.0003	10	1.8599E-06	2.7389E-11	2.4280E-04	3.7514E-04
0.0004	10	3.3955E-06	9.3024E-11	3.4091E-04	5.0025E-04

Table 4.6: Value of absolute error in the PSO-DQM solution of 2nd test problem with $\rho = 6, n = 11\ and\ \Delta t = 1 \times 10^{-6}$ at different levels of time.

x	$t = 0.00010$			$t = 0.0002$		
	Num. value	**Exact value**	**Abs. Error**	**Num. value**	**Exact value**	**Abs. Error**
0.1	0.2257628692996	0.2257634281619	3.7886e-7	0.225879623742	0.225881758036	2.1342e-6
0.2	0.2027609522921	0.2027608716127	8.0679e-8	0.202872642577	0.202872349425	2.9315e-7
0.3	0.1812032056058	0.1812032213726	1.5766e-8	0.181307257299	0.181307308866	5.1566e-8
0.4	0.1611480352987	0.1611480329367	2.3620e-9	0.161244518459	0.161244510103	8.3563e-9
0.5	0.1426256974841	0.1426256992945	1.8104e-9	0.142714475916	0.142714480472	4.5555e-9
0.6	0.1256405426468	0.1256405405606	2.0861e-9	0.125721671388	0.125721665786	5.6021e-9
0.7	0.1101729266816	0.1101729400578	1.3376e-8	0.110246528176	0.110246562534	3.4358e-8
0.8	0.0961822267045	0.0961821575567	6.9147e-8	0.096248719771	0.096248528247	1.9152e-7
0.9	0.0836093218458	0.0836096068232	2.8497e-7	0.083667905038	0.083669057269	1.1522e-6
	$t = 0.0003$			$t = 0.0004$		
0.1	0.2259950683966	0.2260003019396	5.2335e-6	0.2261092349428	0.2261188798587	9.6449e-6
0.2	0.2029844770615	0.2039838634058	6.1365e-7	0.2030964331616	0.2030954135439	1.0196e-6
0.3	0.1814113315203	0.1814114339737	1.0245e-7	0.1815154326416	0.1815155966887	1.6404e-7

0.4	0.1613410425742	0.1613410256438	1.6930e-8	0.1614376067503	0.1614375795559	2.7194e-8
0.5	0.1428032922044	0.1428033001299	7.9225e-9	0.1428921466008	0.1428921582693	1.1668e-8
0.6	0.1258028390791	0.1258028289999	1.0079e-8	0.1258840453382	0.1258840302039	1.5134e-8
0.7	0.1103201611064	0.1103202219756	6.0869e-8	0.1103938272620	0.1103939183868	9.1124e-8
0.8	0.0963152913145	0.0963149344314	3.5688e-7	0.0963819319448	0.0963813761180	5.5582e-7
0.9	0.0837259563702	0.0837285413805	2.5850e-6	0.0837834888559	0.0837880591671	4.5703e-6

The exact and numerical solution at different intervals shown in Figure 4.4. To compare the findings with the exact answer, the results are displayed graphically. The specific approach to the problem is found to be in good accord with the obtained results to assess whether the numerical solution is accurate for $\rho = 6, n = 11 \; and \; \Delta t = 1 \times 10^{-6}$, results are depicted in Table 4.6 and the L_2 and L_∞ errors are presented in Table 4.5.

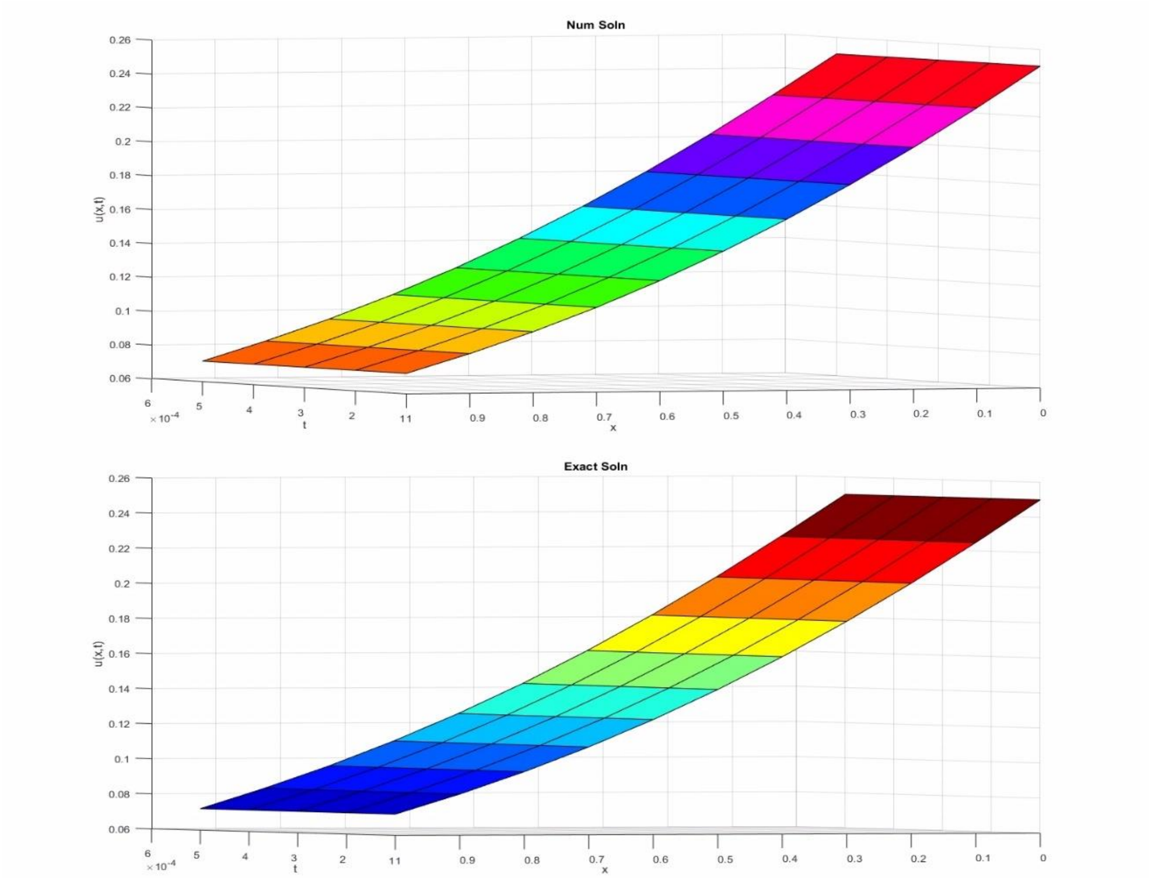


Figure 4.4 Traveling wave solutions of 2^{nd} test problem for $\rho = 6$ and $\Delta t = 1 \times 10^{-6}$.

4.7.3 3rd **test problem:** Consider the equation (4.1) in the domain [0,1]. With initial condition has been considered as $u(x,0) = \frac{1}{(1+exp(\rho x))^2}$

With boundary condition $\lim_{x\to-\infty} u(x,t) = 1$, $\lim_{x\to\infty} u(x,t) = 0$ and exact solution is given by:

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

Table 4.7: L_2 and L_∞ errors obtained by the hybrid scheme for the different time levels results of 3rd test problem for $\rho = 6$ with $h = 1\times 10^{-1}$ and $\Delta t = 1\times 10^{-6}$ at different levels of time.

Time(t)	P	L_2	L_∞	TBC L_2 [162]	TBC L_∞ [162]
0.0001	20	1.8906e-07	2.1298e-13	7.0594E-05	1.2502E-04
0.0002	20	8.3693e-07	5.1964e-12	1.5179E-04	2.5006E-04
0.0003	20	1.9407e-06	2.9406e-11	2.4280E-04	3.7514E-04
0.0004	20	3.4892e-06	9.7229e-11	3.4091E-04	5.0025E-04

Table 4.8: Value of absolute error in the PSO-DQM solution of 3rd test problem with $\rho = 6, n = 11\ and\ \Delta t = 1\times 10^{-6}$ at different levels of time.

x	$t = 0.00010$			$t = 0.0002$		
	Num. value	**Exact value**	**Value of Abs. Error**	**Num. value**	**Exact value**	**Value of Abs. Error**
0.1	0.2257627866610	0.2257632481619	4.6150E-7	0.2258794784654	0.2258817580365	2.2795E-6
0.2	0.2027609402831	0.2027608716127	6.8670E-8	0.2028725991685	0.2028723494258	2.4974E-7
0.3	0.1812032076058	0.1812032213726	1.3767E-8	0.1813072665336	0.1813073088665	4.2330E-8
0.4	0.1611480323046	0.1611480329367	6.3200E-10	0.1612445111409	0.1612445101033	1.0400E-9
0.5	0.1426256963159	0.1426256992945	2.9790E-9	0.1427144740057	0.1427144804720	6.4700E-9
0.6	0.1256405401524	0.1256405405606	4.0800E-10	0.1257216658328	0.1257216657868	4.6000E-11
0.7	0.1101729287962	0.1101729400578	1.1262E-8	0.1102465345315	0.1102465625347	2.8000E-8

0.8	0.0961822174396	0.0961821575567	5.9883E-8	0.0962486925578	0.0962485282471	1.6431E-7
0.9	0.0836092382694	0.0836096068232	3.6855E-7	0.0836677472989	0.0836690572691	1.3099E-6
		$t = 0.0003$			$t = 0.0004$	
0.1	0.2259948792125	0.2260003019396	5.4422E-6	0.2261090193860	0.2261188798587	9.8604E-6
0.2	0.2029843846945	0.2029838634058	5.2129E-7	0.2030962759823	0.2030954135439	8.6244E-7
0.3	0.1814113522192	0.1814114339737	8.1750E-8	0.1815154681274	0.1815155966887	1.2856E-7
0.4	0.1613410299318	0.1613410256438	4.2900E-9	0.1614375880720	0.1614375795559	8.52E-9
0.5	0.1428032898499	0.1428033001299	1.028E-8	0.1428921439961	0.1428921582693	1.427E-8
0.6	0.1258028300526	0.1258028289999	1.05E-9	0.1258840325651	0.1258840302039	2.36E-9
0.7	0.1103201733875	0.1103202219756	4.859E-8	0.1103938467665	0.1103939183868	7.162E-8
0.8	0.0963152383047	0.0963149344314	3.0387E-8	0.0963818460254	0.0963813761180	4.699E-7
0.9	0.0837257332798	0.0837285413805	2.8081E-6	0.0837832087020	0.0837880591671	4.850E-6

The numerical and exact value at different intervals shown in Figure 4.5. For comparison, the findings with the exact answer, the results are displayed graphically. The specific approach to the problem is found to be in good accord with the obtained results to assess whether the numerical solution is accurate for $\rho = 6, n = 11 \; and \; \Delta t = 1 \times 10^{-6}$, results are described in Table 4.8 and the L_2 and L_∞ errors are shown in Table 4.7.

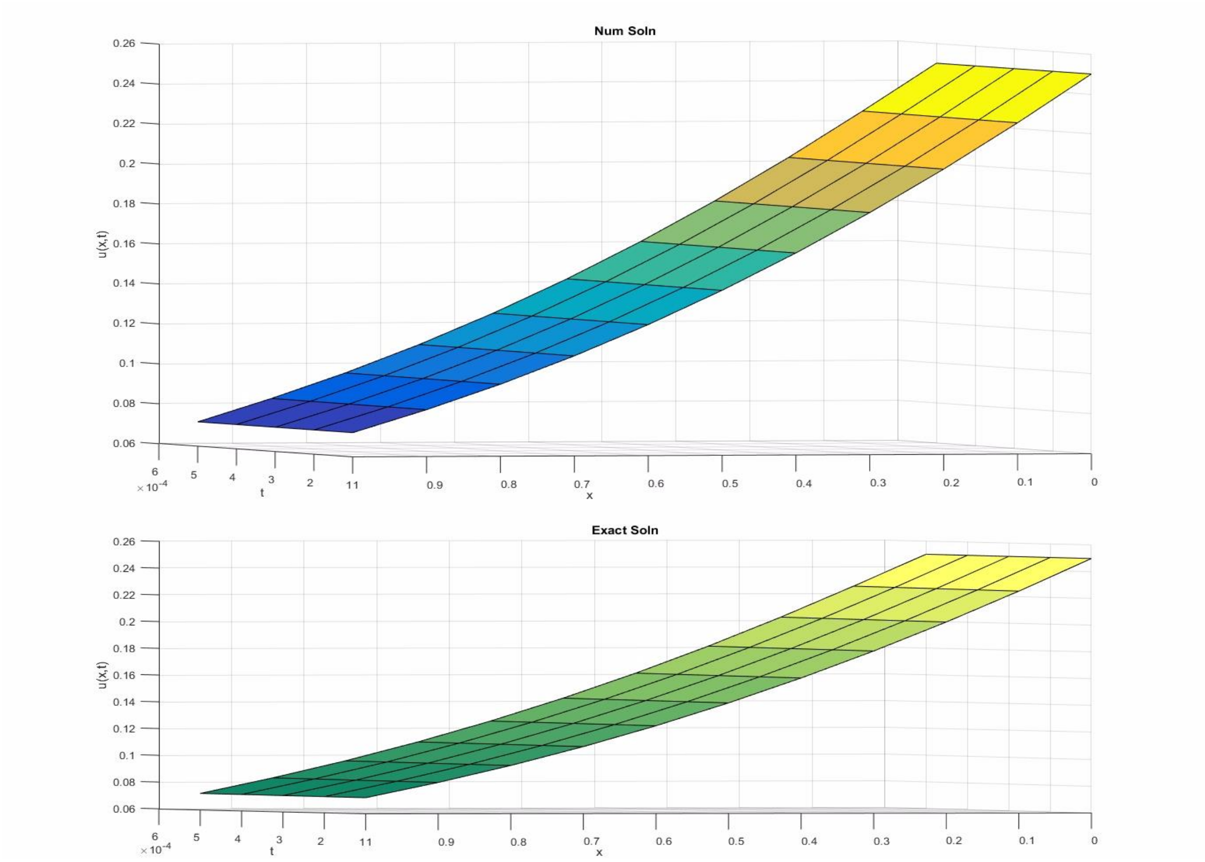


Figure 4.5 Graphical solution of traveling wave presented in 3^{rd} test problem for $\rho = 6$ and $\Delta t = 1 \times 10^{-6}$.

4.7.4 4^{th} **test problem:** Tang and Weber (1991) [130] have taken initial disturbance with a flat top in a constrained location into consideration with $\lambda_1 = 0.1, \lambda_2 = 0.01, \mu = 0.1, x \in [-2,2]$ $and\ y \in [-1,1]\ taking\ h_x = h_y = 0.05$ and $\Delta t = 0.001$

$$u(x, y, 0) = \begin{cases} 1, & if\ x^2 + 4y^2 \le 0.25 \\ exp[-10(x^2 + 4y^2 - 0.25)], & otherwise \end{cases}$$

Figure 4.6 displays the numerical solutions contour and surface plots at times $t = 0,1,2$ and 4. This illustration demonstrates how the top falls first with spreading disruption over time, as opposed to initial disruption, which has a flat top where diffusion is zero in the middle and at the edge diffusion is significant.

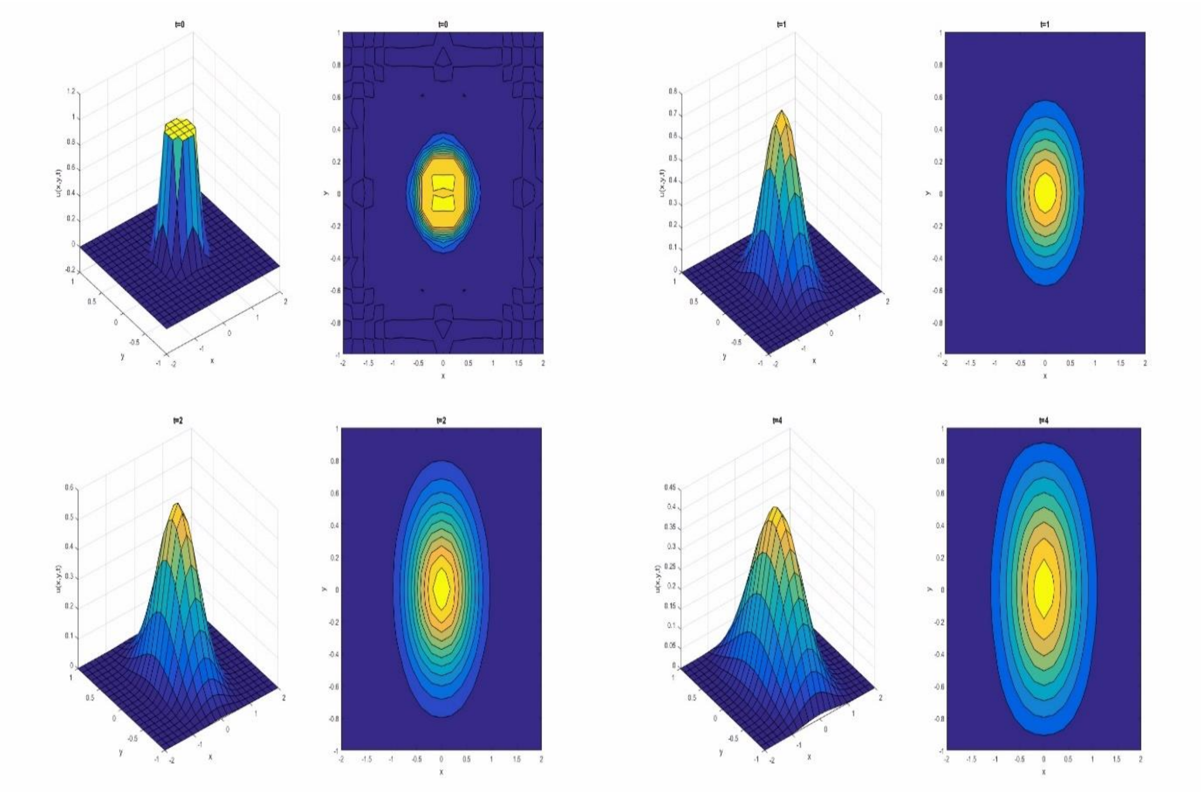


Figure 4.6 The surface and contour plot of 4^{th} test problem at time $t = 0,1,2 \ and \ 4$ with $\Delta t = 1 \times 10^{-3}$ and $h_x = h_y = 0.05$.

4.8 Conclusion:

Travelling wave solutions of the FEs, which have been evaluated by different numerical methods. This nonlinear Fishers problem has been solved using the differential quadrature method in this article employing the exponential B-Spline basis function with PSO. By calculating the L_2 and L_∞, the approximated results are shown. Obtaining the numerical solutions and the errors make it clear that the results are in excellent agreement, with the exact solutions. In the case of two-dimensional work, contour and surface plots are also shown, which are comparable to the results displayed graphically in the previous work. This approach demonstrates to be a cost-effective and viable methodology for obtaining numerical solutions for a distinct-nonlinear real-life models, and may therefore be expanded to effectively handle problems of higher dimensions.

CHAPTER 5

CONCLUSIONS AND FUTURE SCOPE

Nonlinear evolutionary equations dynamically describe nonlinear sciences, the space and time two-dimensional system through the nonlinear systems. NLEEs are a class of nonlinear PDEs that have solitons as the solution and a number of other significant characteristics. NLEEs are called "nonlinear" because they involve nonlinear functions of the variables, which can lead to complex and often unpredictable behaviour of the system. They are also called "evolutionary" because they describe how the system evolves, taking into account the effects of various factors that influence the system's behaviour.

Nonlinear Evolution Equations (NLEEs) have various applications across different fields of science and engineering. These equations describe the time evolution of physical systems with nonlinearities, meaning that the relationship between variables is not directly proportional. Applications of NLEEs in different fields such as in physics the famous nonlinear Korteweg-de Vries equation and burgers equation, describing the behaviour of dispersion effects with nonlinear interactions and used to model various phenomena in fluid dynamics respectively and NLEEs can be used to model biological phenomena like the spread of infectious diseases, population dynamics, and the behaviour of complex biological systems. The Fisher-Kolmogorov equation is an example of an NLEEs used to study the spread of advantageous genes in populations.

In the field of optics, NLEEs are employed to study the behaviour of intense light waves in nonlinear materials, leading to phenomena like self-phase modulation, self-focusing, and optical solitons. NLEEs are used in financial mathematics to model the behaviour of financial instruments and markets. These models take into account various nonlinear factors that affect asset prices and market dynamics. It also plays a role in climate modelling to study the complex interactions between the atmosphere, oceans, and land surface, taking into account nonlinear feedback mechanisms. Also applied in various engineering disciplines, such as electrical engineering, mechanical engineering, and materials science. They are used to analyze and predict the behaviour of nonlinear systems and materials. These are involved in the study of pattern formation in natural phenomena, such as Turing patterns in reaction-diffusion systems. NLEEs are used in image processing and computer vision for tasks like image denoising, edge detection, and image segmentation.

Solving Nonlinear Evolution Equations (NLEEs) analytically can be challenging in most cases, and closed-form solutions may not be possible for many nonlinear systems. However, various techniques and methods can be employed to analyze and approximate solutions to NLEEs. Some NLEEs have special soliton solutions. Solitons are self-reinforcing waves that maintain their shape during propagation. These soliton solutions can be found through various mathematical techniques.

Numerical methods are commonly used to find solutions to Nonlinear Evolution Equations (NLEEs) due to several reasons as analytical techniques may not exist or may be extremely difficult to apply. Numerical methods provide a practical and effective means of approximating solutions in such cases. These methods offer flexibility in handling various types of NLEEs, including systems with high dimensionality, complex boundary conditions, or irregular geometries. They can be adapted to a wide range of problem setups, making them versatile for different applications. They can handle large-scale NLEEs by utilizing parallel computing and high-performance computing resources, allowing for faster and more accurate simulations. NLEEs can exhibit numerical instabilities due to nonlinearities, which can lead to difficulties in finding stable analytical solutions. Numerical methods, when properly implemented, can provide stable and accurate solutions even in the presence of nonlinearities. In the present study, the Crank-Nicolson technique is employed to solve these equations, followed by DQM with a modified form of cubic B-spline basis function to remove extra knot points. The equation is thus reduced to a system of equations by utilising differential quadrature technique to approximate the spatial derivatives. The next step consists of computing the solution of the produced system of equations using MATLAB programming in an iterative way. Since the present method is using the concept of differential quadrature method after discretizing the domain, hence it is acting as a tool to reduce the computational complexity of the collocation approach. As collocation method usually requires complex algebraic manual calculation and a combination of DQM, PSO technique and exponential basis functions are used to discuss different equations. To determine the parameter value which involved in the exponential B-spline method which is significant for the solution that has been resolved by optimisation technique and present study demonstrates how effective the proposed hybrid approach of DQM with modified cubic B-splines and the finite difference scheme for discretization can be implemented to solve the partial differential equations. This work is an effort to develop numerical approach that can provide the approximate solution which is close to the exact solution to the Burgers equation. The present work is showcasing the efficiency of the proposed hybrid approach of Crank-Nicolson based DQM with reframed cubic B-spline method for the study of Fishers equation. This is an effort in the development of the numerical approaches to find the improved solution of the FEs. Travelling wave solutions of the FEs, which have been evaluated by different numerical methods. This nonlinear Fishers problem has been solved using the differential quadrature method in this article employing the exponential B-Spline basis function with PSO. By calculating the L_2 and L_∞, the approximated results are shown. Obtaining the numerical solutions and the errors make it clear that the results are in excellent agreement, with the exact solutions. In the case of two-dimensional work, contour and surface plots are also shown, which are comparable to the results displayed graphically in the previous work.

FUTURE SCOPE:

1. The numerical solutions in this thesis only take initial and boundary concerns into consideration, fractional-order partial differential equations may be used to evaluate these techniques.
2. In this research work discussion about different types of solutions of NLEEs, particularly one of them i.e., soliton type solution has been explored. This may further be extended to explore more type of solutions of PDEs. The presented algorithm is capable of being modified that can be applied to higher order nonlinear differential equations in either linear or nonlinear form.
3. In this research work the exponential cubic B-spline differential quadrature method has been used with PSO technique to find the best value of the parameter p in the basis function other optimization techniques may be explored to optimize the parameter value p in the basis function, for example Artificial Bee Colony (ABC) algorithm, Genetic algorithm techniques, Ant Colony Optimization, Grey wolf optimization, Bat algorithm and many more.

References:

[1] J. S. Russell, “Report on Waves: Made to the Meetings of the British Association in 1842-43,” in *Report of the fourteenth meeting of the British Association for the Advancement of Science*, 1844.

[2] A. Yokus and H. Bulut, “Numerical simulation of KdV equation by finite difference method,” *Indian J. Phys.*, vol. 92, no. 12, 2018.

[3] M. M. Mousa and S. F. Ragab, “Application of the homotopy perturbation method to linear and nonlinear schrödinger equations,” *Zeitschrift fur Naturforsch. - Sect. A J. Phys. Sci.*, vol. 63, no. 3–4, 2008.

[4] T. V. Nguyễn, “Existence of multi-solitary waves with logarithmic relative distances for the NLS equation,” *Comptes Rendus Math.*, vol. 357, no. 1, 2019.

[5] A. L. Fabian, R. Kohl, and A. Biswas, “Perturbation of topological solitons due to sine-Gordon equation and its type,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, no. 4, 2009.

[6] S. Fermi, E., Pasta, J., & Ulam, *Studies of Nonlinear Problems*. Collected Works of Enrico Fermi., 1965.

[7] S. Abbasbandy, H. Roohani Ghehsareh, and I. Hashim, “A meshfree method for the solution of two-dimensional cubic nonlinear Schrödinger equation,” *Eng. Anal. Bound. Elem.*, vol. 37, no. 6, 2013.

[8] T. R. Raoa, “A Study On Linear and Nonlinear Schrodinger Equations by Reduced Differential Transform Method.,” *Malaya J. Mat.*, vol. 4, no. 1, pp. 59–64, 2016.

[9] M. M. El-Horbaty and F. M. Ahmed, “Numerical Optics Soliton Solution of the Nonlinear Schrödinger Equation Using the Laplace and the Modified Laplace Decomposition Method,” *J. Adv. Math. Comput. Sci.*, 2019.

[10] X. Bin Wang and B. Han, “Inverse scattering transform of an extended nonlinear Schrödinger equation with nonzero boundary conditions and its multisoliton solutions,” *J. Math. Anal. Appl.*, vol. 487, no. 1, 2020.

[11] N. Pervaiz and I. Aziz, “Haar wavelet approximation for the solution of cubic nonlinear Schrodinger equations,” *Phys. A Stat. Mech. its Appl.*, vol. 545, 2020.

[12] W. X. Ma, “Inverse scattering for nonlocal reverse-time nonlinear Schrödinger equations,” *Appl. Math. Lett.*, vol. 102, 2020.

[13] D. Kaya, “A numerical solution of the sine-Gordon equation using the modified decomposition method,” *Appl. Math. Comput.*, vol. 143, no. 2–3, 2003.

[14] S. S. Ray, “A numerical solution of the coupled sine-Gordon equation using the modified decomposition method,” *Appl. Math. Comput.*, vol. 175, no. 2, 2006.

[15] B. Batiha, M. S. M. Noorani, and I. Hashim, “Numerical solution of sine-Gordon equation by variational iteration method,” *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 370, no. 5–6, 2007.

[16] M. Dehghan and A. Shokri, “A numerical method for solution of the two-dimensional sine-Gordon equation using the radial basis functions,” *Math. Comput. Simul.*, vol. 79, no. 3, 2008.

[17] U. Yücel, “Homotopy analysis method for the sine-Gordon equation with initial conditions,” *Appl. Math. Comput.*, vol. 203, no. 1, 2008.

[18] A. Sadighi, D. D. Ganji, and B. Ganjavi, “Traveling wave solutions of the sine-gordon and the coupled sine-gordon equations using the homotopy-perturbation method,” *Sci. Iran.*, vol. 16, no. 2 B, 2009.

[19] R. Jiwari, S. Pandit, and R. C. Mittal, “Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method,” *Comput. Phys. Commun.*, vol. 183, no. 3, 2012.

[20] R. E. Bellman and J. Casti, “Differential quadrature and long -term integration,” *J Math. Anal. Appl.*, vol. 34, pp. 235–238, 1971.

[21] R. E. Bellman, B. G. Kashef, and J. Casti, “Differential quadrature :a technique for the rapid solution of nonlinear partial differential equation,” *J. Comput. Phys.*, vol. 10, pp. 40–52, 1972.

[22] C. W. Bert, S. K. Jang, and A. G. Striz, “Two new approximate methods for analyzing free vibration of structural components,” *AIAA J.*, vol. 26, pp. 612–618, 1988.

[23] C. W. Bert and M. Malik, “Differential quadrature method in computational mechanics: A review,” *Appl. Mech. Rev.*, vol. 50, no. 9, 1997.

[24] C. Shu, *Differential quadrature and its application in engineering*. Springer-Verlag London Ltd., 2000.

[25] J. R. Quan and C. T. Chang, “New insights in solving distributed system equations by the quadrature method-I,” *Comput. Chem. Eng.*, vol. 13, pp. 779–788, 1989.

[26] R. C. Mittal and S. Dahiya, “Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method,” *Appl. Math. Lett.*, vol. 313, pp. 442–452, 2017.

[27] A. Korkmaz and I. Dag, “Polynomial based differential quadrature method for numerical solution of nonlinear Burger’s equation,” *J. Franklin Inst.*, vol. 348, no. 10, pp. 2863–2875, 2011.

[28] A. Krowiak, “Hermite type radial basis function-based differential quadrature method for higher order equations,” vol. 40, pp. 2421–2430, 2016.

[29] A. H. Msmali, M. Tamsir, N. Dhiman, and M. A. Aiyashi, “New trigonometric B-spline approximation for numerical investigation of the regularized long-wave equation,” *Open Phys.*, vol. 19, no. 1, pp. 758–769, 2021.

[30] A. H. Msmali, M. Tamsir, and A. A. H. Ahmadini, “Crank-Nicolson-DQM based on cubic exponential B-splines for the approximation of nonlinear Sine-Gordon equation,” *Ain Shams Eng. J.*, vol. 12, no. 4, pp. 4091–4097, 2021.

[31] F. Tornabene, N. Fantuzzi, F. Ubertini, and E. Viola, “Strong formulation finite element method based on differential quadrature: A survey,” *Appl. Mech. Rev.*, vol. 67, no. 2, 2015.

[32] A. Korkmaz and I. Dağ, “Polynomial based differential quadrature method for numerical solution of nonlinear Burgers equation,” *J. Franklin Inst.*, vol. 348, no. 10, 2011.

[33] A. Korkmaz and I. Dag, “Numerical Simulations of Boundary-Forced RLW Equation with Cubic B-Spline-based Differential Quadrature Methods,” *Arab. J. Sci. Eng.*, vol. 38, no. 5, pp. 1151–1160, 2013.

[34] A. Bashan, “An effective application of differential quadrature method based on modified cubic B-splines to numerical solutions of the KdV equation,” *Turkish J. Math.*, vol. 42, no. 1, pp. 373–394, 2018.

[35] R. Kalatehjari, N. Ali, M. Hajihassani, and M. Kholghi Fard, “The application of particle swarm optimization in slope stability analysis of homogeneous soil slopes,” *Int. Rev. Model. Simulations*, vol. 5, no. 1, pp. 458–465, 2012.

[36] G. Arora, V. Joshi, and R. C. Mittal, “Numerical simulation of nonlinear Schrodinger Equation in One and Two Dimensions,” *Math. Model. Comput. Simulations*, vol. 11, no. 4, pp. 634–648, 2019.

[37] A. Bahan, S. Battal, G. Karakoç, and T. Geyikli, “B-spline Differential Quadrature Method for the Modified Burgers’ Equation,” *Ç ankaya Univ. J. Sci. Eng.*, vol. 12, no. 1, 2015.

[38] A. Bashan, N. M. Yagmurlu, Y. Ucar, and A. Esen, “An effective approach to numerical soliton solutions for the Schrodinger equation via modified cubic B-spline differential quadrature method,” *Chaos, Solitons and Fractals*, vol. 100, no. December 2020, pp. 45–56,

2017.

[39] A. Bashan, “Modification of quintic B-spline differential quadrature method to nonlinear Korteweg-de Vries equation and numerical experiments,” *Appl. Numer. Math.*, vol. 167, pp. 356–374, 2021.

[40] A. Bashan, Y. Ucar, N. Murat Yagmurlu, and A. Esen, “A new perspective for quintic B-spline based Crank-Nicolson-differential quadrature method algorithm for numerical solutions of the nonlinear Schrödinger equation,” *Eur. Phys. J. Plus*, vol. 133, no. 1, p. 12, 2018.

[41] A. Korkmaz, A. M. Aksoy, and I. Dag, “Quartic B-spline differential quadrature method,” *Int. Nonlinear Sci.*, vol. 11, no. 4, pp. 403–411, 2011.

[42] A. Korkmaz and I. Dag, “Shock wave simulations using sinc differential quadrature method,” *Eng. Comput. Int. J. Comput. Aided Eng. Softw.*, vol. 28, no. 6, pp. 654–674, 2011.

[43] M. Tamsir, V. K. Srivastava, and R. Jiwari, “An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers ’ equation,” *Appl. Math. Comput.*, vol. 290, pp. 111–124, 2016.

[44] M. Tamsir, V. K. Srivastava, N. Dhiman, and A. Chauhan, “Numerical Computation of Nonlinear Fisher’s Reaction–Diffusion Equation with Exponential Modified Cubic B-Spline Differential Quadrature Method,” *Int. J. Appl. Comput. Math.*, vol. 4, no. 1, pp. 1–13, 2018.

[45] G. Arora and V. Joshi, “Numerical Simulation of Cable Equation Using Differential Quadrature Method,” *Indian J. Ind. Appl. Math.*, vol. 10, no. 1, p. 108, 2019.

[46] G. Arora and V. Joshi, “Comparison of Numerical Solution of 1D Hyperbolic Telegraph Equation using B-Spline and Trigonometric B-Spline by Differential Quadrature Method,” *Indian J. Sci. Technol.*, vol. 9, no. 45, pp. 1–8, 2016.

[47] H. S. Shukla and M. Tamsir, “Numerical solution of nonlinear Sine-Gordon equation by using the modified cubic B-spline differential quadrature method,” *Beni-Suef Univ. J. Basic Appl. Sci.*, vol. 7, no. 4, pp. 359–366, 2018.

[48] H. S. Shukla and M. Tamsir, “An exponential cubic B-spline algorithm for multi-dimensional convection-diffusion equations,” *Alexandria Eng. J.*, vol. 57, no. 3, pp. 1999–2006, 2018.

[49] M. Kapoor and V. Joshi, “Numerical regime " Uniform Algebraic Hyperbolic tension B-spline DQM " for the solution of Fisher ’ s Reaction-Diffusion equation,” *3rd Int. Conf. Appl. Res. Eng. Sci. Technol. Paris, Fr. Diam. Sci. Publ.*, pp. 20–22, 2020.

[50] M. Kapoor, “Numerical simulation of Burgers ’ equations via quartic HB - spline DQM,”

Nonlinear Eng., vol. 12, no. 1, p. 20220264, 2023.

[51] M. Kapoor and V. Joshi, "Numerical solution of coupled 1D Burgers' equation by employing Barycentric Lagrange interpolation basis function based differential quadrature method," *Int. J. Comput. Methods Eng. Sci. Mech.*, vol. 23, no. 3, pp. 263–283, 2022.

[52] S. Kumar, R. Jiwari, R. C. Mittal, and J. Awrejcewicz., "Dark and bright soliton solutions and computational modeling of nonlinear regularized long wave model," *Nonlinear Dyn.*, vol. 104, no. 1, pp. 661–682, 2021.

[53] S. I.J., "Contribution to the problem of approximation of equidistant data by analytical functions," *Quart. Appl. Math*, vol. 4, pp. 45–99, 1946.

[54] S. L.L., *Spline functions, basic theory,* . Wiley.

[55] M. G. Cox, "The numerical evaluation of b-splines," *IMA J. Appl. Math. (Institute Math. Its Appl.*, vol. 10, no. 2, 1972.

[56] J. R. and C. de Boor, "A Practical Guide to Splines.," *Math. Comput.*, vol. 34, no. 149, 1980.

[57] I. Dag and O. E. Hepson, "Hyperbolic-Trigonometric Tension B-Spline Galerkin Approach for the Solution of RLW Equation," in *AIP Conference Proceedings*, 2021.

[58] G. Arora and V. Joshi, "A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions," *Alexandria Eng. J.*, vol. 57, no. 2, 2018.

[59] M. Kapoor and V. Joshi, "A new technique for numerical solution of 1D and 2D non-linear coupled Burgers' equations by using cubic Uniform Algebraic Trigonometric (UAT) tension B-spline based differential quadrature method," *Ain Shams Eng. J.*, vol. 12, no. 4, 2021.

[60] D. Görgülü, M., Dağ, İ., & Irk, *Galerkin Method for the numerical solution of the RLW equation by using exponential B-splines.* 2015.

[61] A. T. Abed and A. S. Y. Aladool, "Appling Particle Swarm Optimization Based on Pade Approximant to Solve Ordinary Differential Equation.," *.Numer. Algebr. Control Optim*, vol. 12, no. 2, pp. 321–337, 2022.

[62] J. Alikhani, M. Firouznia, S. Mohammad, and M. Hosseini, "Finding a good shape parameter of RBF to solve PDEs based on the particle swarm optimization algorithm," *Alexandria Eng. J.*, vol. 57, no. 4, pp. 3641–3652, 2018.

[63] A. G. Gad, *Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review*, vol. 29, no. 5. Springer Netherlands, 2022.

[64] Y. Zhang, S. Wang, and G. Ji, “A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications,” *Math. Probl. Eng.*, vol. 2015, p. 38, 2015.

[65] B. T. Pham *et al.*, “A novel hybrid soft computing model using random forest and particle swarm optimization for estimation of undrained shear strength of soil,” *Sustainability*, vol. 12, no. 6, pp. 1–16, 2020.

[66] J. Robinson and Y. Rahmat-Samii, “Particle swarm optimization in electromagnetics,” *IEEE Trans. Antennas Propag.*, vol. 52, no. 2, pp. 397–407, 2004.

[67] R. V Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: A brief survey,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 2, pp. 262–267, 2011.

[68] M. Juneja and S. K. Nagar, “Particle swarm optimization algorithm and its parameters : A review,” *6 Int. Conf. Control. Comput. Commun. Mater.*, vol. IEEE, no. Iccccm, pp. 1–5, 2016.

[69] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *MHS’95. Pro_ceedings Sixth Int. Symp. Micro Mach. Hum. Sci.*, no. IEEE, pp. 39–43, 1995.

[70] L. Yuliawati, W. S. Budhi, and D. Adytia, “Numerical Studying of Soliton in the Korteweg-de Vries (KdV) Equation,” in *Journal of Physics: Conference Series*, 2019.

[71] H. Washimi and T. Taniuti, “Propagation of ion-acoustic solitary waves of small amplitude,” *Phys. Rev. Lett.*, vol. 17, no. 19, 1966.

[72] O. T. Kolebaje and O. E. Oyewande, “Numerical Solution of the Korteweg De Vries Equation by Finite Difference and Adomian Decomposition Method,” *Int. J. Basic Appl. Sci.*, vol. 1, no. 3, 2012.

[73] G. C. Das and J. Sarma, “ Response to ‘Comment on “A new mathematical approach for finding the solitary waves in dusty plasma” ’ [Phys. Plasmas 6 , 4392 (1999)] ,” *Phys. Plasmas*, vol. 6, no. 11, 1999.

[74] A. R. Osborne, “The inverse scattering transform: Tools for the nonlinear fourier analysis and filtering of ocean surface waves,” *Chaos, Solitons and Fractals*, vol. 5, no. 12, 1995.

[75] L. A. Ostrovsky and Y. A. Stepanyants, “Do internal solitions exist in the ocean?,” *Reviews of Geophysics*, vol. 27, no. 3. 1989.

[76] D. Reatto, L. and Galli, “‘What is ROTON?’.,” *Mod. Phys. B*, vol. 13, no. 5, pp. 607–616, 1999.

[77] R. L. Horne, “A (Very) Brief Introduction to Soliton Theory in a class of Nonlinear PDEs,” *Russell J. Bertrand Russell Arch.*, no. July, 2002.

[78] D. G. Crighton, “Applications of KdV,” *Acta Appl. Math.*, vol. 39, no. 1–3, pp. 39–67, 1995.

[79] Z. Yan and H. Zhang, “New explicit solitary wave solutions and periodic wave solutions for Whitham-Broer-Kaup equation in shallow water,” *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 285, no. 5–6, 2001.

[80] Y. Lei, Z. Fajiang, and W. Yinghai, “The homogeneous balance method, Lax pair, Hirota transformation and a general fifth-order KdV equation,” *Chaos, Solitons and Fractals*, vol. 13, no. 2, 2002.

[81] A. Rashid, “Numerical solution of Korteweg-de Vries equation by the fourier pseudospectral method,” *Bull. Belgian Math. Soc. - Simon Stevin*, vol. 14, no. 4, 2007.

[82] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura, “Method for solving the Korteweg-deVries equation,” *Phys. Rev. Lett.*, vol. 19, no. 19, 1967.

[83] H. Z. Liu, “A note on ‘Jacobi elliptic function solutions for the modified Korteweg-de Vries equation,’” *J. King Saud Univ. - Sci.*, vol. 26, no. 2, 2014.

[84] Y. Hufford, C., & Xing, “Superconvergence of the local discontinuous Galerkin method for the linearized Korteweg–de Vries equation.,” *J. Comput. Appl. Math.*, vol. 255, pp. 441–455, 2014.

[85] T. Trogdon and B. Deconinck, “Numerical computation of the finite-genus solutions of the Korteweg-de Vries equation via Riemann-Hilbert problems,” *Appl. Math. Lett.*, vol. 26, no. 1, 2013.

[86] C. Grava, T., & Klein, “A numerical study of the small dispersion limit of the Korteweg–de Vries equation and asymptotic solutions.,” *Phys. D Nonlinear Phenom.*, vol. 241, no. 23–24, p. Grava, T., Klein, C. (2012). A numerical study o, 2012.

[87] J. A. Leach, “The large-time development of the solution to an initial-value problem for the generalized Korteweg-de Vries equation,” *Appl. Math. Lett.*, vol. 24, pp. 214–218, 2011.

[88] M. Kumar, B. R., & Mehra, “Time-accurate solutions of Korteweg–de Vries equation using wavelet Galerkin method.,” *Appl. Math. Comput.*, vol. 162, no. 1, pp. 447–460, 2005.

[89] A. R. Bahadir, “Exponential finite-difference method applied to Korteweg-de Vries equation for small times,” *Appl. Math. Comput.*, vol. 160, no. 3, 2005.

[90] A. (2006). Aksan, E. N., & Özdeş, “Numerical solution of Korteweg–de Vries equation by

Galerkin B-spline finite element method.," *Appl. Math. Comput.*, vol. 175, no. 2, pp. 1256–1265, 2006.

[91] S. Özer, S., & Kutluay, "An analytical–numerical method for solving the Korteweg–de Vries equation.," *Appl. Math. Comput.*, vol. 164, no. 3, pp. 789–797, 2005.

[92] U. M. Ascher and R. I. McLachlan, "Multisymplectic box schemes and the Korteweg-de Vries equation," in *Applied Numerical Mathematics*, 2004. doi: 10.1016/j.apnum.2003.09.002.

[93] A. Kutluay, S., Bahadir, A. R., & Özdeş, "A small time solutions for the Korteweg–de Vries equation.," *Appl. Math. Comput.*, vol. 107, no. 2–3, pp. 203–210, 2000.

[94] S. Idrees, M., Islam, S., Tirmizi, S. I. A., & Haq, "Application of the optimal homotopy asymptotic method for the solution of the Korteweg–de Vries equation.," *Math. Comput. Model.*, vol. 55, no. 3–4, pp. 1324–1333, 2012.

[95] N. Gücüyenen and G. Tanoğlu, "On the numerical solution of Korteweg-de Vries equation by the iterative splitting method," in *Applied Mathematics and Computation*, 2011.

[96] J. Sarma, "Solitary wave solution of higher-order Korteweg-de Vries equation," *Chaos, Solitons and Fractals*, vol. 39, no. 1, 2009.

[97] E. Van de Fliert, B. W., & Van Groesen, *On variational principles for coherent vortex structures.* Springer Netherlands., 1993.

[98] M. He-Ping and G. Ben-Yu, "The Fourier pseudospectral method with a restrain operator for the Korteweg-de Vries equation," *J. Comput. Phys.*, vol. 65, no. 1, 1986.

[99] M. Dehghan and A. Shokri, "A numerical method for KdV equation using collocation and radial basis functions," *Nonlinear Dyn.*, vol. 50, no. 1–2, 2007.

[100] Bellman, R., Kashef, B., Lee, E. S., & Vasudevan, R. "Differential quadrature and splines," Computer and Mathematics with Applications., vol. 1, no. 3-4, 1975.

[101] A. Başhan, S. B. G. Karakoç, and T. Geyikli, "Approximation of the KdVB equation by the quintic B-spline differential quadrature method," *Kuwait J. Sci.*, vol. 42, no. 2, 2015.

[102] J. R. Quan and C. T. Chang, "New insights in solving distributed system equations by the quadrature method-II. Numerical experiments," *Comput. Chem. Eng.*, vol. 13, no. 9, 1989.

[103] C. Shu and B. E. Richards, "Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations," *Int. J. Numer. Methods Fluids*, vol. 15, no. 7, 1992.

[104] R. J. Spiteri and S. J. Ruuth, "A new class of optimal high-order strong-stability-preserving

time discretization methods," *SIAM J. Numer. Anal.*, vol. 40, no. 2, 2002.

[105] A. A. Soliman, A. H. A. Ali, and K. R. Raslan, "Numerical solution for the KdV equation based on similarity reductions," *Appl. Math. Model.*, vol. 33, no. 2, 2009.

[106] I. Dağ and Y. Dereli, "Numerical solutions of KdV equation using radial basis functions," *Appl. Math. Model.*, vol. 32, no. 4, 2008.

[107] D. Kong, Y. Xu, and Z. Zheng, "A hybrid numerical method for the KdV equation by finite difference and sinc collocation method," *Appl. Math. Comput.*, vol. 355, 2019.

[108] B. Saka and I. Dağ, "Quartic B-spline collocation method to the numerical solutions of the Burgers' equation," *Chaos, Solitons and Fractals*, vol. 32, no. 3, 2007.

[109] H. Bateman, "Some recent researches on the motion of fluids.," *Mon. Weather Rev.*, vol. 43, no. 4, pp. 163–170, 1915.

[110] J. M. Burgers, "A mathematical model illustrating the theory of turbulence.," *Adv. Appl. Mech.*, vol. 1, pp. 171–199, 1948.

[111] B. K. Arora, G., & Singh, "Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method.," *Appl. Math. Comput.*, vol. 224, pp. 166–177, 2013.

[112] A. Başhan, "Correction to: Nonlinear dynamics of the Burgers' equation and numerical experiments," *Mathematical Sciences*, vol. 16, no. 3. 2022.

[113] T. Öziş, E. N. Aksan, and A. Özdeş, "A finite element approach for solution of Burgers' equation," *Appl. Math. Comput.*, vol. 139, no. 2–3, 2003.

[114] A. Dogan, "A Galerkin finite element approach to Burgers' equation," *Appl. Math. Comput.*, vol. 157, no. 2, 2004.

[115] I. Dağ, D. Irk, and A. Şahin, "B-spline collocation methods for numerical solutions of the Burgers' equation," *Math. Probl. Eng.*, vol. 2005, no. 5, 2005.

[116] R. C. Mittal and G. Arora, "Numerical solution of the coupled viscous Burgers' equation," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, no. 3, 2011.

[117] R. C. Mittal and R. K. Jain, "Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method," *Appl. Math. Comput.*, vol. 218, no. 15, 2012.

[118] M. K. Kadalbajoo, K. K. Sharma, and A. Awasthi, "A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equations," *Appl. Math. Comput.*, vol. 170, no. 2, 2005.

[119] H. A. Hassanien, I. A., Salama, A. A., & Hosham, "Fourth-order finite difference method for solving Burgers' equation.," *Appl. Math. Comput.*, vol. 170, no. 2, pp. 781–800, 2005.

[120] W. Liao, "An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation.," *Appl. Math. Comput.*, vol. 206, no. 2, pp. 755–764, 2008.

[121] B. Inan and A. R. Bahadir, "Numerical solution of the one-dimensional Burgers' equation: Implicit and fully implicit exponential finite difference methods," *Pramana - J. Phys.*, vol. 81, no. 4, 2013.

[122] M. Ghasemi, "High order approximations using spline-based differential quadrature method : Implementation to the multi-dimensional PDEs," *Appl. Math. Model.*, vol. 46, pp. 63–80, 2017.

[123] Y. Li, S., Petzold, L., & Ren, "Stability of moving mesh systems of partial differential equations.," *SIAM J. Sci. Comput.*, vol. 20, no. 2, pp. 719–738, 1998.

[124] B. K. Singh and M. Gupta, "A New Efficient Fourth Order Collocation Scheme for Solving sine – Gordon Equation," *Int. J. Appl. Comput. Math.*, vol. 123, no. 7, p. 138, 2021.

[125] B. K. Singh, G. Arora, and P. Kumar, "A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme," *Ain Shams Eng. J.*, vol. 9, no. 4, 2018.

[126] R. A. FISHER, "The wave of advance of advantageous genes," *Ann. Eugen.*, vol. 7, no. 4, 1937.

[127] J. Gazdag, J., & Canosa, "Numerical solution of Fisher's equation.," *J. Appl. Probab.*, vol. 11, no. 3, pp. 445–457, 1974.

[128] A. Ablowitz, M. J., & Zeppetella, "Explicit solutions of Fisher's equation for a special wave speed.," *Bull. Math. Biol.*, vol. 41, no. 6, pp. 835–840, 1979.

[129] W. G. Twizell, E. H., Wang, Y., & Price, "Chaos-free numerical solutions of reaction-diffusion equations.," *Proc. R. Soc. London. Ser. A Math. Phys. Sci.*, vol. 430, no. 1880, pp. 541–576, 1990.

[130] R. Tang, S., & Weber, "Numerical study of Fisher's equation by a Petrov-Galerkin finite element method.," *ANZIAM J.*, vol. 33, no. 1, pp. 27–38, 1991.

[131] R. E. Mickens, "A best finite-difference scheme for the fisher equation.," *Numer. Methods Partial Differ. Equ.*, vol. 10, no. 5, pp. 581–585, 1994.

[132] Y. Qiu and D. M. Sloan, "Numerical solution of Fisher's equation using a moving mesh

method,” *J. Comput. Phys.*, vol. 146, no. 2, 1998.

[133] U. Rizwan, “Comparison of the nodal integral method and non-standard finite-difference scheme for the Fisher’s equation.,” *SIAM J. Sci. Comput*, vol. 22, no. 6, pp. 1926–1942, 2001.

[134] A. Wazwaz, A. M., & Gorguis, “An analytic study of Fisher’s equation by using Adomian decomposition method.,” *Appl. Math. Comput.*, vol. 154, no. 3, pp. 609–620, 2004.

[135] B. D. Olmos, D., & Shizgal, “A pseudospectral method of solution of Fisher’s equation.,” *J. Comput. Appl. Math.*, vol. 193, no. 1, pp. 219–242, 2006.

[136] G. Singh, B. K., & Arora, “A numerical scheme to solve Fisher-type reaction-diffusion equations.,” *Nonlinear Stud. Eng. Sci. Aerosp.*, vol. 5, no. 2, pp. 153-164., 2014.

[137] V. S. Aswin and A. Awasthi, “Polynomial Based Differential Quadrature Methods for the Numerical Solution of Fisher and Extended Fisher–Kolmogorov Equations,” *Int. J. Appl. Comput. Math.*, vol. 3, 2017.

[138] S. Mittal, R. C., & Kumar, “Numerical study of Fisher’s equation by wavelet Galerkin method.,” *Int. J. Comput. Math.*, vol. 83, no. 3, pp. 287–298, 2006.

[139] M. Aghamohamadi, J. Rashidinia, and R. Ezzati, “Tension spline method for solution of non-linear Fisher equation,” *Appl. Math. Comput.*, vol. 249, 2014.

[140] G. Mittal, R. C., & Arora, “Efficient numerical solution of Fisher’s equation by using B-spline method.,” *Int. J. Comput. Math.*, vol. 87, no. 13, pp. 3039–3051, 2010.

[141] R. K. Mittal, R. C., & Jain, “Numerical solutions of nonlinear Fisher’s reaction–diffusion equation with modified cubic B-spline collocation method.,” *Math. Sci.*, vol. 7, no. 1, p. 12, 2013.

[142] M. J. Tamsir, M., & Huntul, “A numerical approach for solving Fisher’s reaction–diffusion equation via a new kind of spline functions.,” *Ain Shams Eng. J.*, vol. 12, no. 3, pp. 3157–3165, 2021.

[143] B. Şahin, A., Dağ, İ., & Saka, “AB-spline algorithm for the numerical solution of Fisher’s equation.,” *Kybernetes*, vol. 37, no. 2, pp. 326–342, 2008.

[144] M. Ilati and M. Dehghan, “Direct local boundary integral equation method for numerical solution of extended Fisher–Kolmogorov equation,” *Eng. Comput.*, vol. 34, no. 1, 2018.

[145] R. A. Rubin, S. G., & Graves Jr, *A cubic spline approximation for problems in fluid mechanics*. 1975.

[146] B. K. Singh, J. P. Shukla, and M. Gupta, “Study of One Dimensional Hyperbolic Telegraph

Equation Via a Hybrid Cubic B-Spline Differential Quadrature Method," *Int. J. Appl. Comput. Math.*, vol. 7, no. 1, 2021.

[147] B. K. Singh and P. Kumar, "An algorithm based on a new DQM with modified exponential cubic B-splines for solving hyperbolic telegraph equation in (2 + 1) dimension," *Nonlinear Eng.*, vol. 7, no. 2, pp. 113–125, 2018.

[148] A. Bashan, "A mixed method approach to Schrodinger equation: Finite difference method and quartic B-spline based differential quadrature method," *Int. J. Optim. Control Theor. Appl.*, vol. 9, no. 2, pp. 223–235, 2021.

[149] C. Shu and Y. L. Wu, "Integrated radial basis functions-based differential quadrature method and its performance," *Int. J. Numer. Methods Fluids*, vol. 53, no. 6, 2007.

[150] G. Devshali, P., & Arora, "Solution of two-dimensional fractional diffusion equation by a novel hybrid D (TQ) method.," *Nonlinear Eng.*, vol. 11, no. 1, pp. 135–142, 2022.

[151] A. C. M. Tamsir, V. K. Srivastava, N. Dhiman, "Numerical Computation of Nonlinear Fisher's Reaction–Diffusion Equation with Exponential Modified Cubic B-Spline Differential Quadrature Method.," *Int. J. Appl. Comput. Math*, vol. 4, no. 1, pp. 1–13, 2018.

[152] D. Qiu, Y., & Sloan, "(1998). Numerical solution of Fisher's equation using a moving mesh method. , 146(2), .," *J. Comput. Phys.*, vol. 146, no. 2, pp. 726–746, 1998.

[153] D. D. Bhatta and M. I. Bhatti, "Numerical solution of KdV equation using modified Bernstein polynomials," *Appl. Math. Comput.*, vol. 174, no. 2, 2006, doi: 10.1016/j.amc.2005.05.049.

[154] N. Parekh and S. Puri, "A new numerical scheme for the Fisher equation," *J. Phys. A. Math. Gen.*, vol. 23, no. 21, 1990.

[155] K. Al-Khaled, "Numerical study of Fisher's reaction-diffusion equation by the Sinc collocation method," *J. Comput. Appl. Math.*, vol. 137, no. 2, 2001.

[156] T. Mavoungou and Y. Cherruault, "Numerical study of fisher's equation by Adomian's method," *Math. Comput. Model.*, vol. 19, no. 1, 1994.

[157] G. W. Zhao, S., & Wei, "Comparison of the discrete singular convolution and three other numerical schemes for solving Fisher's equation.," *SIAM J. Sci. Comput.*, vol. 25, no. 1, pp. 127–147, 2003.

[158] V. Kapoor, M., & Joshi, "Solution of non-linear Fisher's reaction-diffusion equation by using Hyperbolic B-spline based differential quadrature method.," *J. Phys. Conf. Ser.*, vol. 1531, no. 1, 2020.

[159] V. Arora, G., & Joshi, “A computational approach for one and two dimensional Fisher’s equation using quadrature technique.,” *Am. J. Math. Manag. Sci.*, vol. 40, no. 2, pp. 145–162, 2021.

[160] G. Arora, S. Mishra, and B. K. Singh, “Nonlinear dynamics of the Fisher’s equation with numerical experiments,” *Nonlinear Stud.*, vol. 29, no. 3, 2022.

[161] A. T. Abed and A. S. Y. Aladool, “Appling Particle Swarm Optimization Based on Pade Approximant to Solve Ordinary Differential Equation,” *Numer. Algebr. Control Optim.*, vol. 12, no. 2, pp. 321–337, 2022.

[162] G. Arora and V. Joshi, “A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes,” *Ain Shams Eng. J.*, vol. 9, no. 4, 2018.

[163] V. Arora, G., & Joshi, “Simulation of generalized nonlinear fourth order partial differential equation with quintic trigonometric differential quadrature method.,” *Math. Model. Comput. Simulations*, vol. 11, no. 6, pp. 1059–1083, 2019.

[164] N. S. Kolmogorov, A. N., Petrovsky, I. G., & Piskunov, “Investigation of the equation of diffusion combined with increasing of the substance and its application to a biology problem.,” *Bull. Moscow State Univ. Ser. A Math. Mech*, vol. 1, no. 6, pp. 1–25, 1937.

[165] W. E. Kastenberg and P. L. Chambré, “On the Stability of Nonlinear Space-Dependent Reactor Kinetics,” *Nucl. Sci. Eng.*, vol. 31, no. 1, 1968.

[166] J. Gazdag, J., & Canosa, “Numerical solution of Fisher’s equation.,” *J. Appl. Probab.*, vol. 11, no. 3, pp. 445-457., 1974.

[167] M. Jeyar, A. Lamnii, M. Yassir Nour, F. Oumellal, and A. Zidna, “An Algebraic Hyperbolic Spline Quasi-Interpolation Scheme for Solving Burgers-Fisher Equations,” in *Simulation Modeling*, 2022.

[168] Y. Carey, G. F., & Shen, “Least-squares finite element approximation of Fisher’s reaction–diffusion equation.,” *Numer. Methods Partial Differ. Equ.*, vol. 11, no. 2, pp. 175–186, 1995.

[169] D. A. Larson, “Transient Bounds and Time-Asymptotic Behavior of Solutions to Nonlinear Equations of Fisher Type,” *SIAM J. Appl. Math.*, vol. 34, no. 1, 1978, doi: 10.1137/0134008.

[170] V. Arora, G., & Joshi, “A computational approach for one and two dimensional Fisher’s equation using quadrature technique.,” *Am. J. Math. Manag. Sci.*, vol. 40, no. 2, pp. 145–162, 2021.

[171] R. Mittal, R.C. and Jiwari, “Numerical study of Fisher’s equation by using differential

quadrature method.," *Int. J. Inf. Syst. Sci*, vol. 5, no. 1, pp. 143–160, 2009.

[172] O. Dag, I. and Ersoy, "The exponential cubic B-spline algorithm for Fisher equation.," *Chaos Solit. Fract*, vol. 86, pp. 101–106, 2016.

[173] C. G. Zhu and W. S. Kang, "Numerical solution of Burgers-Fisher equation by cubic B-spline quasi-interpolation," *Appl. Math. Comput.*, vol. 216, no. 9, 2010.

[174] A. Cattani, C., & Kudreyko, "Mutiscale analysis of the Fisher equation.," in *In Computational Science and Its Applications–ICCSA 2008: International Conference.*, Perugia, Italy: Springer Berlin Heidelberg., 2008, pp. 1171–1180.

[175] M. S. Sahimi and D. J. Evans, "THE ALTERNATING GROUP EXPLICIT (AGE) ITERATIVE METHOD TO SOLVE PARABOLIC AND HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS," *Annu. Rev. Heat Transf.*, vol. 2, no. 2, 2013.

[176] Y. Tan, H. Xu, and S. J. Liao, "Explicit series solution of travelling waves with a front of Fisher equation," *Chaos, Solitons and Fractals*, vol. 31, no. 2, 2007.

[177] A. M. Wazwaz, "The tanh method for traveling wave solutions of nonlinear equations," *Appl. Math. Comput.*, vol. 154, no. 3, 2004.

[178] H. (2007). A. method of travelling wave solutions for a class of nonlinear reaction diffusion equation. Jiaqi, M., Weijiang, Z., & Ming, "Asymptotic method of travelling wave solutions for a class of nonlinear reaction diffusion equation.," *Acta Math. Sci.*, vol. 27, no. 4, pp. 777–780, 2007.

[179] L. A. Martino, A. Osella, C. Dorso, and J. L. Lanata, "Fisher equation for anisotropic diffusion: Simulating South American human dispersals," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 76, no. 3, 2007.

[180] S. Ai and W. Huang, "Travelling waves for a reaction-diffusion system in population dynamics and epidemiology," *R. Soc. Edinburgh - Proc. A*, vol. 135, no. 4, 2005.

[181] J. Ross, A. F. Villaverde, J. R. Banga, S. Vázquez, and F. Morán, "A generalized Fisher equation and its utility in chemical kinetics," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 107, no. 29, 2010.

[182] P. K. Maini, D. L. S. McElwain, and D. Leavesley, "Travelling waves in a wound healing assay," *Appl. Math. Lett.*, vol. 17, no. 5, 2004.

[183] Jain, M. K., Iyengar, S. R., & Jain, R. K. (2007). Numerical methods: problems and solutions. New Age International.

List of Published and Communicated Papers/ Book Chapter/ List of Attended Conferences

List of Published and Communicated Papers

1. Arora, G., Mishra, S., & Singh, B.K. (2022). Numerical dynamic of Fisher's equation with numerical experiments. *Nonlinear Studies*, *29(3)*, 665-675. (Scopus, Impact Factor-0.46, SJR 2022-0.166, Q4, CiteScore-1.0, ISSN-1359-8678,

Web link of journal indexing-

http://www.nonlinearstudies.com/index.php/nonlinear.

)

2. Arora, G., Mishra, S., & Emadifar, H., Khademi M. (2022). Numerical Simulations and Dynamics of Burger's Equation Using the Modified Cubic B-Spline Differential Quadrature Method. *Discrete Dynamics in Nature and Society*, 2023. (Sci, Impact Factor-1.457, SJR 2022-0.264, Q3, CiteScore-2.0, ISSN- 1026-0226,

Web link of journal indexing-

https://www.hindawi.com/journals/ddns.

)

3. Mishra, S., Arora, G., & Emadifar, H., Sahoo S.K., Ghanizadeh A. (2022). Differential Quadrature Method to Examine the Dynamic Behavior of Soliton Solutions to the Korteweg-de-Vries Equation. *Advances in Mathematical Physics, 2022*. (Sci, Impact Factor-1.15, SJR 2022-0.263, Q3, CiteScore-1.9, ISSN-1687-9120,

Web link of journal indexing-

https://www.hindawi.com/journals/amp.

)

4. Mishra, S., Arora, G. (2023). Particle Swarm Optimization for solving Nonlinear Fisher's equation.

5. Arora, G., Mishra, S. (2023) Exploring the application of Fisher equation (1D and 2D) and Numerical solution using Particle Swarm Optimization and Differential Quadrature Method.

Book Chapter

Mishra, S., & Arora, G., (2022). "B-spline Basis Function and its various forms explained Concisely" has been accepted for publication as a book chapter in the book "Advance Numerical Techniques to solve linear and Nonlinear Differential Equations" under River Publication (Scopus Indexed).

List of Attended Conferences

1. Presented a paper as Poster Presentation entitled "Numerical study of travelling-wave solutions for the Korteweg-de Vries equation" in 3rd International Conference on Recent Advances in Fundamental and Applied Sciences" (RAFAS 2021) held on 25th - 26th June 2021 at Lovely Professional University, Phagwara, Punjab, India.

2. Presented a paper as Oral Presentation entitled "Nonlinear Dynamics of the Fishers equation with numerical experiments" in 5th International Conference on "Mathematical Techniques in Engineering Applications" (ICMTEA2021) held on 3rd - 4th December 2021 at Graphic Era Deemed to be university, Dehradun, Uttarakhand, India.

3. Presented a paper as Oral Presentation entitled "Study of Fisher's Equation with different B-spline function" in "1st International Conference on Mathematical Methods and Techniques in Engineering and Sciences" (ICMMTES2022) at Graphic Era Deemed to be University & Graphic Era Hill University, Dehradun, India held on 9-10 Dec 2022.